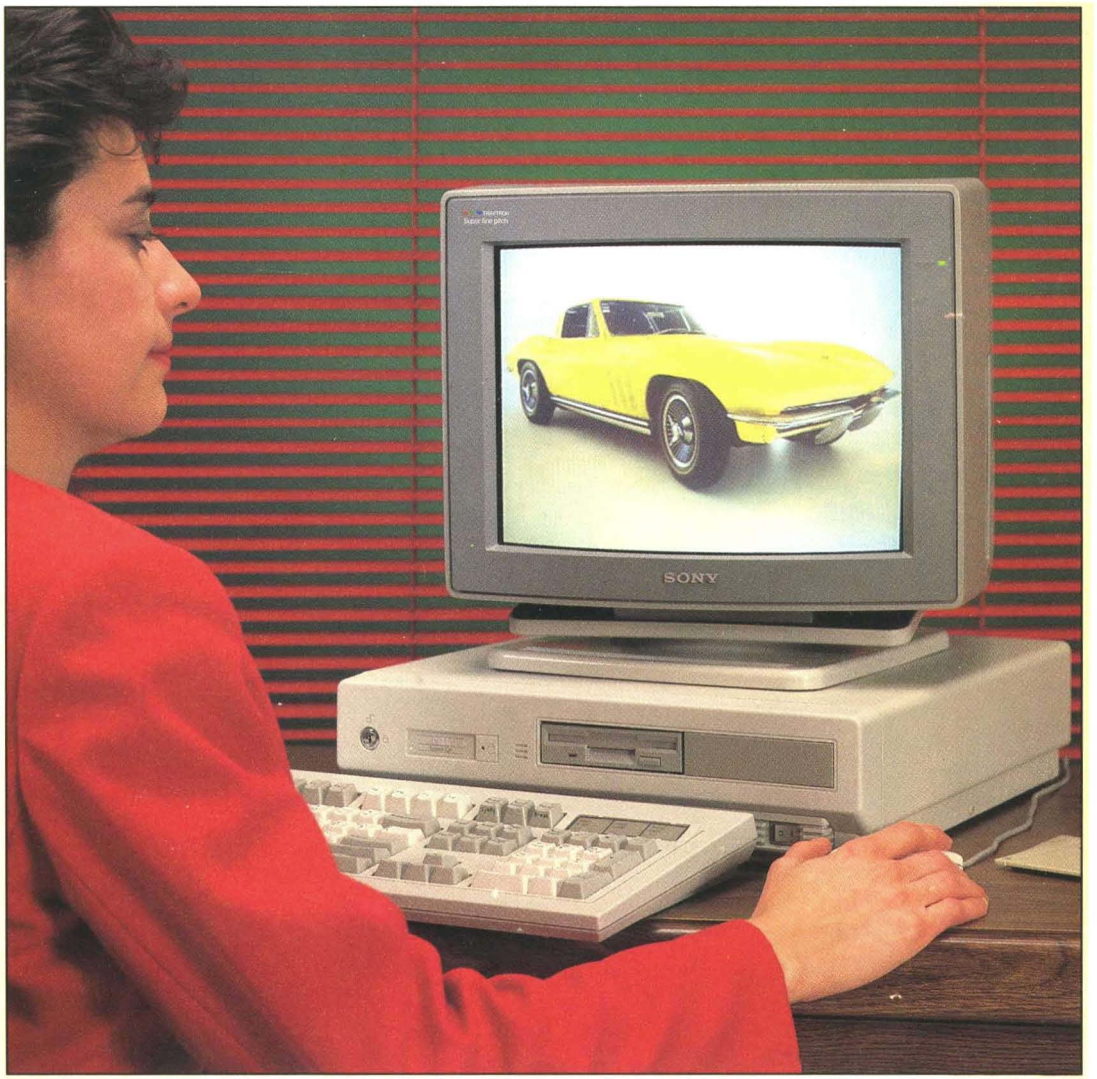
Personal Computer Products

Processors, Coprocessors, Video, and Mass Storage

Advanced
Micro
Devices



Advanced
Micro
Devices

Personal Computer Products Data Book

© 1989 Advanced Micro Devices

Advanced Micro Devices reserves the right to make changes in its products without notice in order to improve design or performance characteristics. The performance characteristics listed in this document are guaranteed by specific tests, correlated testing, guard banding, design and other practices common to the industry.
For specific testing details, contact your local AMD sales representative.
The company assumes no responsibility for the use of any circuits described herein.

901 Thompson Place, P.O. Box 3453, Sunnyvale, California 94088-3000
(408)732-2400 TWX: 910-339-9280 TELEX: 34-6306

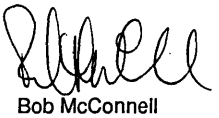
PAL is a registered trademark of Advanced Micro Devices, Inc.
Ada is a registered trademark of the Department of Defense.
ASM186 is a trademark of Microtech Research, Inc.
dBASE is a trademark of Ashton-Tate.
DEC F, DEC D, DEC G, and VAX are trademarks of the Digital Equipment Corporation.
Ethernet is a registered trademark of Xerox Corporation.
1-2-3 is a trademark of Lotus Corporation.
IBM PC/AT is a registered trademark of IBM Corporation.
MULTIBUS is a trademark of Intel Corporation.
Norton SI is a trademark of Peter Norton Computing, Inc.
NT**Multibus is a trademark of Intel Corporation.
OS/2 is a trademark of Microsoft and International Business Machines Corporation.
Paradox is a registered trademark of AT&T Technologies, Inc.
PC-XT is a trademark of International Business Machines Corporation.
TopView is a trademark of International Business Machines Corporation.
Z8000 is a trademark of Zilog, Inc.
Z8530H is a trademark of Zilog, Inc.

INTRODUCTION

Advanced Micro Devices is dedicated to providing silicon systems solutions to the personal computer industry. These solutions include specific AMD integrated circuits as well as support tools from third party vendors designed to aid in the debug and evaluation of your PC designs.

The Personal Computer Products Data Book includes 80X86 processors and coprocessors. In addition, support peripherals such as color palettes, mass storage devices, and document processing products are included.

At AMD we are dedicated to keeping your designs competitive with leading edge solutions. Nearly 60 percent of AMD's personal computer products are CMOS high-performance ICs. CMOS is where AMD has the opportunity to apply our process technology expertise and our systems expertise most readily.



Bob McConnell
Vice President
Personal Computer Products Division

PREFACE

Advanced Micro Devices provides a broad product portfolio for personal computer applications. Included in this data book are microprocessors, video products, mass storage devices, and related peripherals available to optimize the cost performance of your PC design. AMD also manufactures other PC-related products such as EPROM and PAL devices listed in data books available from your local AMD sales office.

This data book is divided by product type into six chapters, with blue pages highlighting AMD CMOS products. Chapter 1 lists the 80XX and 80286 processors. In addition to the high-performance NMOS 16-MHz 80286, AMD now offers a CMOS version, the 80C286, which runs at clock speeds from 12.5 to 25 MHz. The 80C286 microprocessor is a cost-effective, high-performance solution for desktop and laptop PC markets. Also in this chapter is AMD's CMOS 80C287, which is a plug-in compatible with Intel's NMOS 80287 and is offered at speeds up to 16 MHz. It also is capable of automatic sleep mode for laptop PC applications.

Chapter 2 contains support peripherals for computational systems—the high-performance CMOS Am29C325 Double-Precision Floating Point Processor, the popular Z8530H Serial Communications Controller used in AppleTalk network connections, and the CMOS Z85C30, which is capable of a fast Mb/s data rate.

Chapter 3 focuses on graphics products for personal computers and high-performance workstations. AMD offers a full line of industry standard NMOS and CMOS VGA color palettes for IBM- and Apple-compatible PCs, including products with low-power sleep mode, permitting longer battery life performance for laptop PC applications. For workstations, AMD offers a total systems solution Am95C60 QPDM for the display of bit-mapped and alphanumeric graphics.

Chapter 4 presents both CMOS and NMOS products for mass storage applications. The Am95C94 Advanced Burst Error Processor, Am95C95 Magnetic Disk Controller, and Am95C96 Optical Disk Controller combine to create high density and high-performance disk-embedded control designs with error detection and correction on the fly. In addition, Am53C80N and Am33C93A SCSI chips provide system interface for your mass storage needs.

Chapter 5 lists AMD's document processing products. The Am95C71 VCEP can compress an 8½ × 11-inch page format to 4 percent of its original size for FAXing and then expand it back to its original size at 60 Mb/s. The high speeds of the VCEP make this device ideal for those systems requiring real time processing of stored data while reducing the amount of data stored.

The final chapter of general information includes packaging, thermal characteristics, and support literature available on all AMD PC products.

PERSONAL COMPUTER PRODUCTS DATA BOOK

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

SYSTEM INTEGRATION GUIDE

Personal Computer Segment	Processor Products	Video Products	Mass Storage Products	Other Related Products*
Cost Sensitive PCs	8088/8086 80286–8, –10, –12		Am9580A Am9590	Am9517A
Price/Performance PCs	80286–16 80C286–16, –20, –25	Am81C471/478 Am81C176 Am81C453	Am95C94 Am95C95 Am95C96	AMD 80C287 Am53C80N Am33C93 See note 1
Laptop PCs	80C286–12, –16, –20	Am81EC176 Am81EC478 Am81EC471	Am95C94	AMD 80EC287
Workstations	Am29000, other RISC 680XX, etc.	Am81C453 Am81C458 Am95C60	Am95C94 Am95C95 Am95C96	Am9513 Am9516
Peripheral Cards or Devices				
FAX cards	Am7971A			
Graphics	Am95C60 Am29000	Am81C176		
Networking	80186/88			See note 2
Disk Control	80186/88 80C521		Am95C94 Am95C95 Am95C96	Am33C93 Am53C80N
Memory management				Am95C85
Laser printers	Am95C75 Am95C76 Am29000			

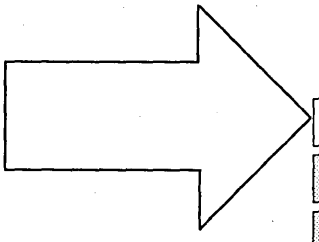
* AMD makes a complete line of PAL® and EPROM devices necessary for PC design. Further information can be found in other data books available from your AMD sales office.

Notes: 1. AMD's PC products are completely compatible with Integrated Peripherals such as those available from Chips & Technologies, Headland (G-2), VLSI Technologies, Faraday, VIA, etc. An 80286 cache controller for 20- and 25-MHz–based cache designs is available from Austek Microsystems.

2. AMD manufactures a complete line of Networking and Telecommunications Products. Further information can be found in other data books available from your AMD sales office.

NUMERIC LISTING

Advanced
Micro
Devices



CHAPTER 1
MICROPROCESSORS

8086

16-Bit Microprocessor
iAPX86 Family
FINAL

DISTINCTIVE CHARACTERISTICS

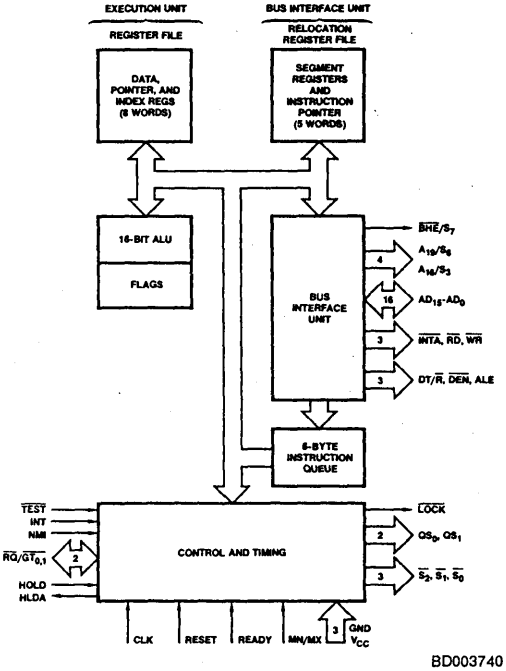
- Directly addresses up to 1 Mbyte of memory
- 24 operand addressing modes
- Efficient implementation of high level languages
- Instruction set compatible with 8080 software
- Bit, byte, word, and block operations
- 8 and 16-bit signed and unsigned arithmetic in binary or decimal
- MULTIBUS® system interface
- Three speed options
  - 5MHz for 8086
  - 8MHz for 8086-2
  - 10MHz for 8086-1

GENERAL DESCRIPTION

The 8086 is a general purpose 16-bit microprocessor CPU. Its architecture is built around thirteen 16-bit registers and nine 1-bit flags. The CPU operates on 16-bit address spaces and can directly address up to 1 megabyte using offset addresses within four distinct memory segments, designated as code, data, stack and extra code. The 8086 implements a powerful instruction set with 24 operand addressing modes. This instruction set is compatible with that of the 8080 and 8085. In addition, the 8086 is particularly effective in executing high level languages.

The 8086 can operate in minimum and maximum modes. Maximum mode offloads certain bus control functions to a peripheral device and allows the CPU to operate efficiently in a multi-processor system. The CPU and its high performance peripherals are MULTIBUS compatible. The 8086 is implemented in N-channel, depletion load, silicon gate technology and is contained in a 40-pin CERDIP package, Molded DIP package, or Plastic Leaded Chip Carrier.

BLOCK DIAGRAM

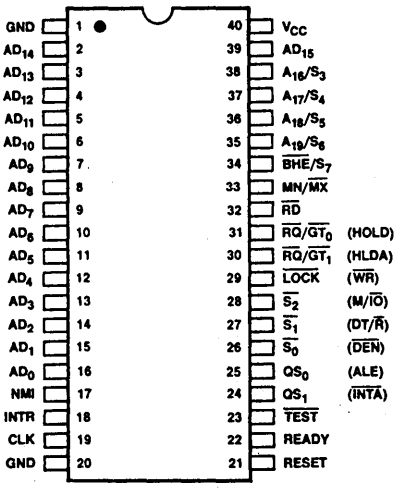


BD003740

Publication # 01966 | Rev. D | Amendment /0
Issue Date: August 1989

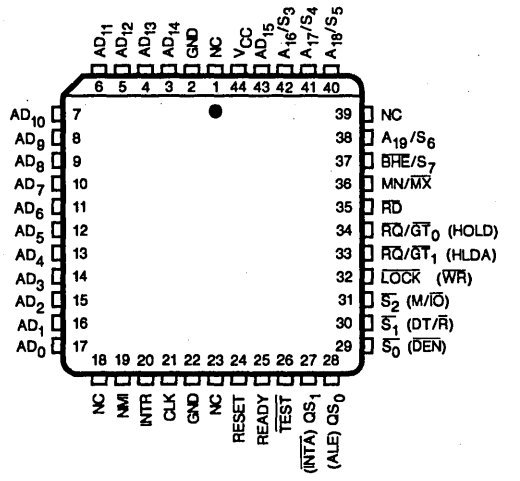
CONNECTION DIAGRAMS
Top View

DIP



CD005511

PLCC



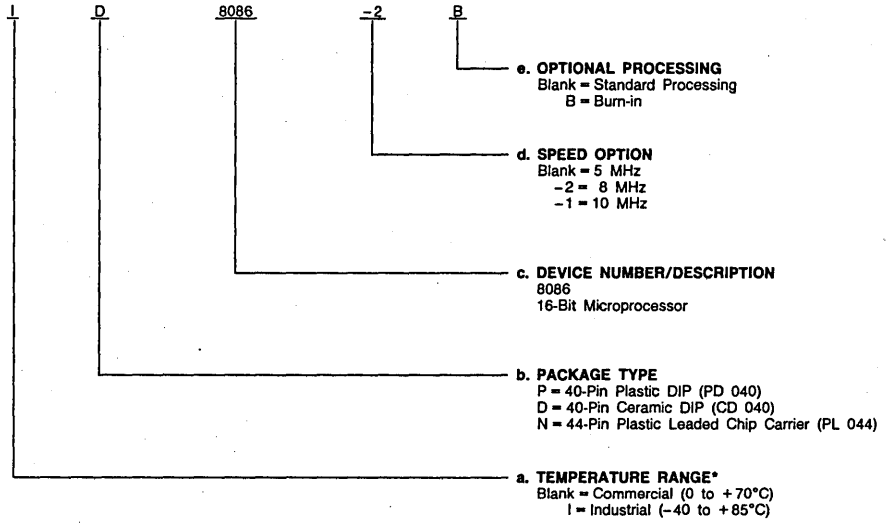
CD010701

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commercial Products

AMD commercial products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
P, D, N	8086
	8086-2
	8086-1
D, ID	8086-2B
D	8086-1
ID	8086B

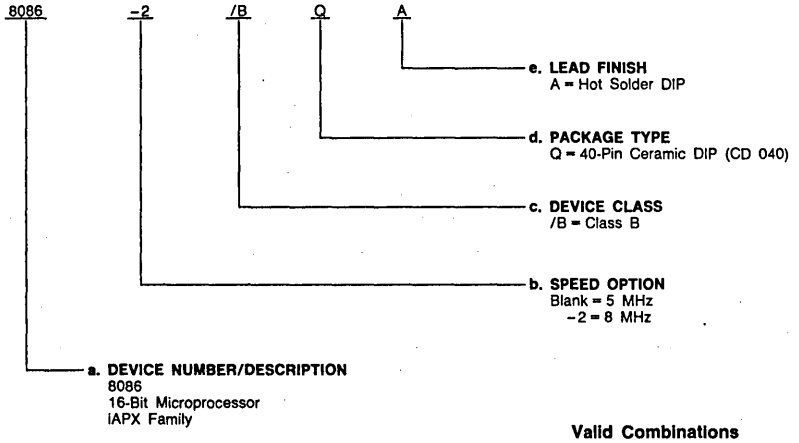
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION

Military Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
8086	/BQA
8086-2	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

The following pin function descriptions are for 8086 systems in either minimum or maximum mode. The "Local Bus" in these descriptions is the direct multiplexed bus interface connection to the 8086 (without regard to additional bus buffers).

Pin No.*	Name	I/O	Description
39, 2–16	AD_{15}–AD_0	I/O	Address Data Bus. These lines constitute the time multiplexed memory/IO address (T_1) and data (T_2, T_3, T_W, T_4) bus. A_0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D_7–D_0. It is LOW during T_1 when a byte is to be transferred on the lower portion of the bus in memory or I/O operations. Eight-bit oriented devices tied to the lower half would normally use A_0 to condition chip select functions. (See $\overline{BHE}$.) These lines are active HIGH and float to three-state OFF during interrupt acknowledge and local bus "hold acknowledge."
35–38	A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3	O	Address/Status. During T_1 these are the four most significant address lines for memory operations. During I/O operations these lines are LOW. During memory and I/O operations, status information is available on these lines during T_2, T_3, T_W, and T_4. The status of the interrupt enable FLAG bit (S_5) is updated at the beginning of each CLK cycle. A_{17}/S_4 and A_{16}/S_3 are encoded as shown. This information indicates which relocation register is presently being used for data accessing. These lines float to three-state OFF during local bus "hold acknowledge."
34	$\overline{BHE}/S_7$	O	Bus High Enable/Status. During T_1 the bus high enable signal ($\overline{BHE}$) should be used to enable data onto the most significant half of the data bus, pins D_{15}–D_8. Eight-bit oriented devices tied to the upper half of the bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is LOW during T_1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the high portion of the bus. The S_7 status information is available during T_2, T_3, and T_4. The signal is active LOW and floats to three-state OFF in "hold." It is LOW during T_1 for the first interrupt acknowledge cycle.
32	$\overline{RD}$	O	Read. Read strobe indicates that the processor is performing a memory of I/O read cycle, depending on the state of the S_2 pin. This signal is used to read devices which reside on the 8086 local bus. $\overline{RD}$ is active LOW during T_2, T_3, and T_W of any read cycle and is guaranteed to remain HIGH in T_2 until the 8086 local bus has floated. This signal floats to three-state OFF in "hold acknowledge."
22	READY	I	READY. Is the acknowledgment from the addressed memory or I/O device that it will complete the data transfer. The READY signal from memory/IO is synchronized by the 8284A Clock Generator to form READY. This signal is active HIGH. The 8086 READY input is not synchronized. Correct operation is not guaranteed if the set-up and hold times are not met.
18	INTR	I	Interrupt Request. Is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
23	$\overline{TEST}$	I	$\overline{TEST}$. Input is examined by the "Wait" instruction. If the $\overline{TEST}$ input is LOW, execution continues; otherwise, the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
17	NMI	I	Non-Maskable Interrupt. An edge-triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
21	RESET	I	Reset. Causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the Instruction Set description, when RESET returns LOW. RESET is internally synchronized.
19	CLK	I	Clock. Provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
40	V_{CC}		V_{CC}. The + 5 V power supply pin.
1, 20	GND		Ground. The ground pin.
33	MN/$\overline{MX}$	I	Minimum/Maximum. Indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

A_{17}/S_4	A_{16}-S_3	Characteristics
0 (LOW)	0	Alternate Data
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data
S_6 is 0 (LOW)		

$\overline{BHE}$	A_0	Characteristics
0	0	Whole word
0	1	Upper byte from/ to odd address
1	0	Lower byte from/ to even address
1	1	None

*Pin numbers correspond to DIPs only.

1

PIN DESCRIPTION (continued)

Pin No.*	Name	I/O	Description
28-26	$\overline{S_2}$, $\overline{S_1}$, $\overline{S_0}$	O	Status. Active during T_4, T_1, and T_2 and is returned to the passive state (1, 1, 1) during T_3 or during T_W when READY is HIGH. This status is used by the 8288 Bus Controller to generate all memory and I/O access control signals. Any change by S_2, S_1, or S_0 during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle. These signals float to three-state OFF in "hold acknowledge." These status lines are encoded as shown.

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1 (HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

31, 30	$\overline{RQ}/\overline{GT_0}$, $\overline{RQ}/\overline{GT_1}$	I/O	Request/Grant. Pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT_0}$ having higher priority than $\overline{RQ}/\overline{GT_1}$. $\overline{RQ}/\overline{GT}$ has an internal pull-up resistor so it may be left unconnected. The request/grant sequence is as follows: 1. A pulse of 1 CLK wide from another local bus master indicates a local bus request ("hold") to the 8086 (pulse 1). 2. During a T_4 or T_1 clock cycle, a pulse 1 CLK wide from the 8086 to the requesting master (pulse 2), indicates that the 8086 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge." 3. A pulse 1 CLK wide from the requesting master indicates to the 8086 (pulse 3) that the "hold" request is about to end and that the 8086 can reclaim the local bus at the next CLK. Each master-master exchange of the local bus is a sequence of 3 pulses. There must be one dead CLK cycle after each bus exchange. Pulses are active LOW. If the request is made while the CPU is performing a memory cycle, it will release the local bus during T_4 of the cycle when all the following conditions are met: 1. Request occurs on or before T_2. 2. Current cycle is not the low byte of a word (on an odd address). 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. If the local bus is idle when the request is made, two possible events will follow: 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.
29	$\overline{LOCK}$	O	$\overline{LOCK}$. Output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and floats to three-state OFF in "hold acknowledge."
24, 25	QS_1, QS_0	O	Queue Status. The queue status is valid during the CLK cycle after which the queue operation is performed. QS_1 and QS_0 provide status to allow external tracking of the internal 8086 instruction queue.
28	M/$\overline{IO}$	O	Status line. Logically equivalent to S_2 in the maximum mode. It is used to distinguish a memory access from an I/O access. M/IO becomes valid in the T_4 preceding a bus cycle and remains valid until the final T_4 of the cycle (M = HIGH, IO = LOW). M/$\overline{IO}$ floats to three-state OFF in local bus "hold acknowledge."
29	$\overline{WR}$	O	Write. Indicates that the processor is performing a write memory or write I/O cycle, depending on the state of M/IO signal. WR is active for T_2, T_3, and T_W of any write cycle. It is active LOW and floats to three-state OFF in local bus "hold acknowledge."
24	$\overline{INTA}$	O	$\overline{INTA}$. Is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T_2, T_3, and T_W of each interrupt acknowlegde cycle.
25	ALE	O	Address Latch Enable. Provided by the processor to latch the address into 8282/8283 address latch. It is a HIGH pulse active during T_1 of any bus cycle. Note that ALE is never floated.
27	DT/$\overline{R}$	O	Data Transmit/Receive. Needed in minimum system that desires to use an 8286/8287 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically DT/$\overline{R}$ is equivalent to $\overline{S_1}$ in the maximum mode, and its timing is the same as for M/$\overline{IO}$. (T = HIGH, R = LOW.) This signal floats to three-state OFF in local bus "hold acknowledge."
26	$\overline{DEN}$	O	Data Enable. Provided as an output enable for the 8286/8287 in a minimum system which uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access and for INTA cycles. For a read or $\overline{INTA}$ cycle, it is active from the middle of T_2 until the middle of T_4, while for a write cycle, it is active from the beginning of T_2 until the middle of T_4. $\overline{DEN}$ floats to three-state OFF in local bus "hold acknowledg

*Pin numbers correspond to DIPs only.

PIN DESCRIPTION (continued)

Pin No.*	Name	I/O	Description
31, 30	HOLD, HLDA	I/O	HOLD. Indicates that another master is requesting a local bus "hold." To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgement in the middle of a T_4 or T_1 clock cycle. Simultaneous with the issuance of HLDA, the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor will LOWer HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. The same rules as for $\overline{RQ}/\overline{GT}$ apply, regarding when the local bus will be released. HOLD is not asynchroneous input. External synchronization should be provided if the system cannot otherwise guarantee the set-up time.

*Pin numbers correspond to DIPs only.

DETAILED DESCRIPTION

The 8086 CPU is internally organized into two processing units. These two units are the Bus Interface Unit (BIU) and the Execution Unit (EU). A block diagram of this organization is shown on page 1.

The BIU performs instruction fetch and queuing, operand fetch and store, address relocation, and basic bus control. The EU receives operands and instructions from the BIU and processes them on a 16-bit ALU. The EU accesses memory and peripheral devices through requests to the BIU. The BIU generates physical addresses in memory using the 4 segment registers and offset values.

The BIU and EU usually operate asynchronously. This permits the 8086 to overlap execution fetch and execution. Up to 6 instruction bytes can be queued. The instruction queue acts as a FIFO buffer for instructions, from which the EU extracts instruction bytes as required.

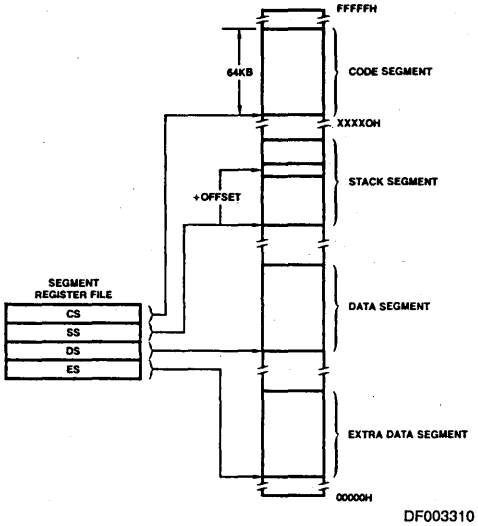
Memory Organization

The 8086 addresses up to 1 megabyte of memory. The address space is organized as a linear array, from 00000 to FFFFF in hexadecimal. Memory is subdivided into segments of 64K bytes each. There are 4 segments: code, stack, data, and extra (usually employed as an extra data segment). Each segment thus contains information of a similar type. Selection of a destination segment is automatically performed using the rules in the table below. This segmentation makes memory more easily relocatable and supports a more structured programming style.

Physical addresses in memory are generated by selecting the appropriate segment, obtaining the segment "base" address from the segment register, shifting the base address 4 digits to the left, and then adding this base to the "offset" address. For programming code, the offset address is obtained from the instruction pointer. For operands, the offset address is calculated in several ways, depending upon information contained in the addressing mode. Memory organization and address generation are shown in Figure 1a.

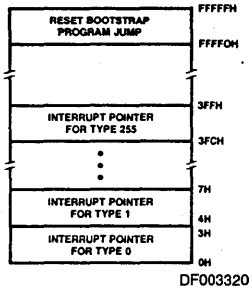
Certain memory locations are reserved for specific CPU operations. These are shown in Figure 1b. Addresses FFFFOH through FFFFFH are reserved for operations which include a jump to the initial program loading routine. After RESET, the CPU will always begin execution at location FFFFOH, where the jump must be located.

Addresses 00000H through 003FFH are reserved for interrupt operations. The service routine of each of the 256 possible interrupt types is signaled by a 4-byte pointer. The pointer elements must be stored in reserved memory addresses before the interrupts are invoked.



DF003310

Figure 1a. Memory Organization



DF003320

Figure 1b. Reserved Memory Locations

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic for all prefetching of instructions.
Stack	STACK (SS)	All stack pushes and pops, and all memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references which are relative to the stack, the destination of a string operation, or explicitly overriden.
External (Global) Data	EXTRA (ES)	Destination of string operations, when they are explicitly selected using a segment override.

Minimum and Maximum Modes

The 8086 has two system configurations, minimum and maximum mode. The CPU has a strap pin, MN/$\overline{MX}$, which defines the system configuration. The status of this strap pin defines the function of pin numbers 24 through 31.

When MN/$\overline{MX}$ is strapped to GND, the 8086 operates in maximum mode. The operations of pins 24 through 31 are redefined. In maximum mode, several bus timing and control functions are "off-loaded" to the 8288 bus controller, thus freeing up the CPU. The CPU communicates status information to the 8288 through pins S_0, S_1, and S_2. In maximum mode, the 8086 can operate in a multiprocessor system, using the LOCK signal within a Multibus format.

When MN/$\overline{MX}$ is strapped to V_{CC}, the 8086 operates in minimum mode. The CPU sends bus control signals itself through pins 24 through 31. This is shown in the Connection Diagrams (in parentheses). Examples of minimum and maximum mode systems are shown in Figure 2.

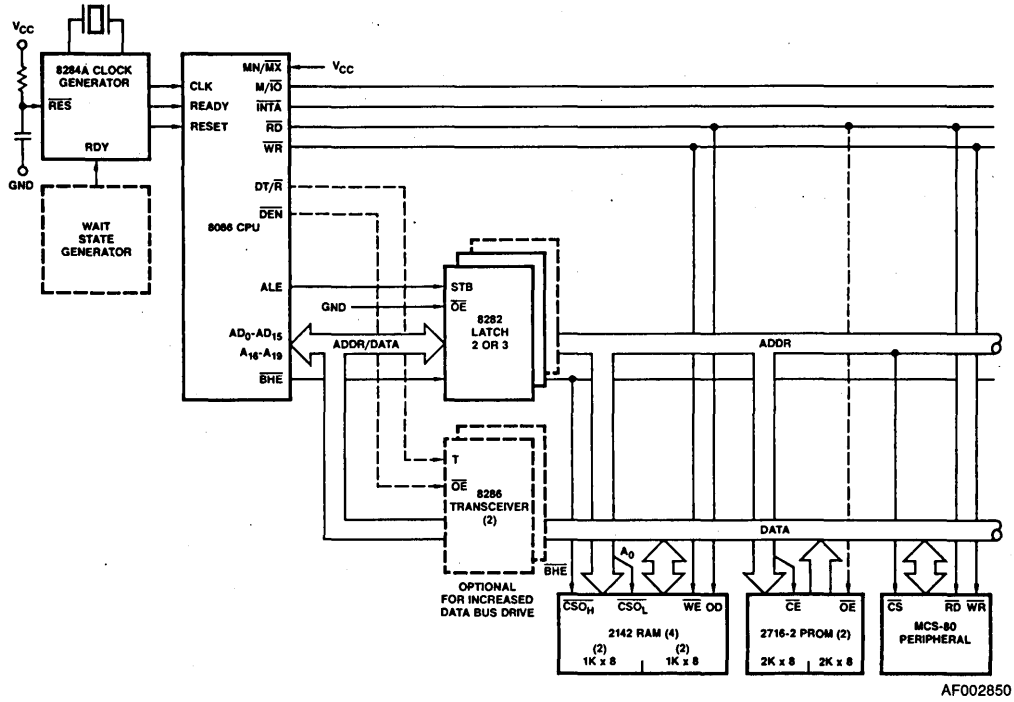
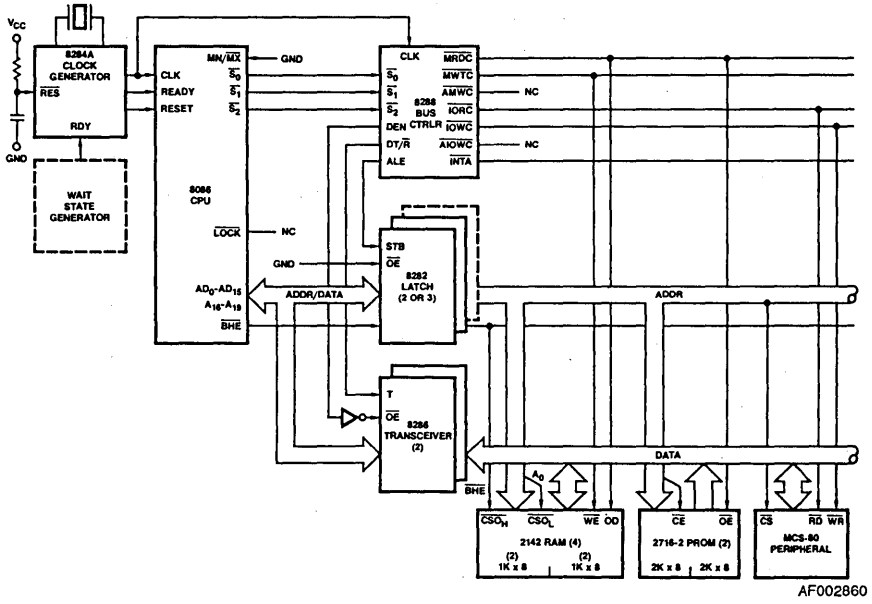


Figure 2a. Minimum Mode 8086 Typical Configuration



AF002860

Figure 2b. Maximum Mode 8086 Typical Configuration

Bus Operation

The 8086 has a combined address and data bus, commonly referred to as "a time multiplexed bus." This technique provides the most efficient use of pins on the processor while permitting the use of a standard 40-lead package. This bus can be used throughout the system with address latching provided on memory and I/O modules. The bus can also be demultiplexed at the processor with a single set of address latches if a standard non-multiplexed bus is desired for the system.

Each bus cycle consists of at least four CLK cycles. These are referred to as T_1, T_2, T_3 and T_4 (see Figure 5). The address is sent from the processor during T_1. Data transfer occurs on the bus during T_3 and T_4. T_2 is used for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "Wait" states (T_W) are inserted between T_3 and T_4. Each inserted "Wait" state is of the same duration as a CLK cycle. "Idle" states (T_I) or inactive CLK cycles can occur between 8086 bus cycles. The processor uses these cycles for internal housekeeping.

During T_1 of any bus cycle, the ALE (Address Latch Enable) signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/$\overline{MX}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S_0}$, $\overline{S_1}$, and $\overline{S_2}$ are used, in maximum mode, by the bus controller to identify the type of bus transaction according to the following table:

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Characteristics
0(LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1(HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S_3 through S_7 are multiplexed with high-order address bits and the $\overline{BHE}$ signal, and are therefore valid during T_2 through T_4. S_3 and S_4 indicate which segment register (see Instruction Set description) was used for this bus cycle in forming the address, according to the following table:

S_4	S_3	Characteristics
0(LOW)	0	Alternate Data (extra segment)
0	1	Stack
1(HIGH)	0	Code or None
1	1	Data

S_5 is a reflection of the PSW interrupt enable bit. S_6 = 0 and S_7 is a spare status bit.

I/O Addressing

8086 I/O operations can address up to a maximum of 64K I/O byte registers or 32K I/O word registers. The I/O address appears in the same format as the memory address on bus lines A_{15}–A_0. The address lines A_{19}–A_{16} are zero in I/O operations. I/O instructions which use register DX as a pointer have full address capability. Direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

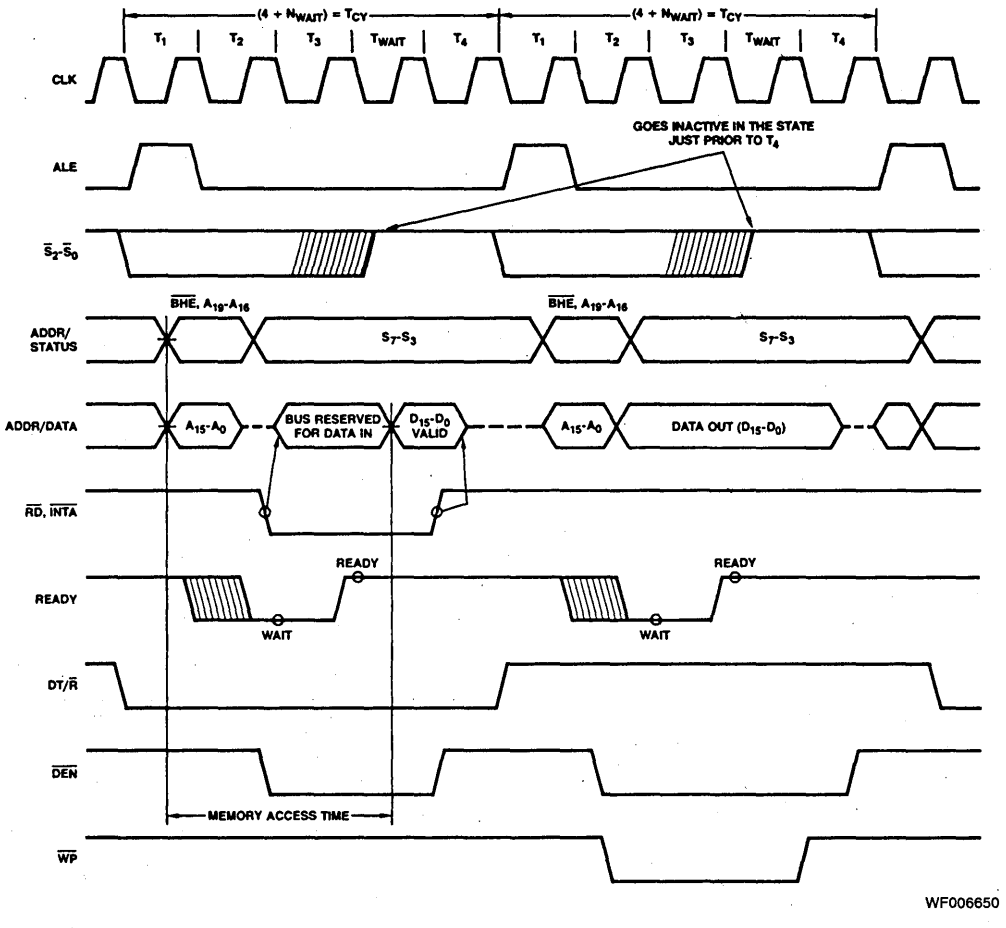


Figure 3. Basic System Timing

EXTERNAL INTERFACE

Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 8086 RESET is required to be HIGH for greater than 4 CLK cycles. The 8086 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 10 CLK cycles. After this interval the 8086 operates normally beginning with the instruction in absolute location FFFF0H (see Figure 1b). The details of this operation are explained in the Instruction Set description of the MCS-86 Family User's Manual. The RESET input is internally synchronized to the processor clock. At initialization the HIGH-to-LOW transition of RESET must occur no sooner than 50μs after power-up, to allow complete initialization of the 8086.

NMI may not be asserted prior to the 2nd CLK cycle following the end of RESET.

Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are described in the Instruction Set description. Hardware interrupts are either non-maskable or maskable.

Interrupts transfer control to a new program location. A 256-element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see Figure 1b), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type." An interrupting device supplies an 8-bit type number during the interrupt acknowledge sequence, which is used to "vector" through the appropriate element to the new interrupt service program location.

Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt pin (NMI) which has higher priority than the maskable interrupt request pin (INTR). A typical use would be to activate a power

failure routine. The NMI is edge-triggered on a LOW-to-HIGH transition. The activation of this pin causes a type 2 interrupt. (See Instruction Set description.)

NMI is required to have a duration in the HIGH state of greater than two CLK cycles, but is not required to be synchronized to the clock. Any high-going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves of a block-type instruction. Worst case response to NMI would be to multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

Maskable Interrupt (INTR)

The 86/10 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable FLAG status bit. The interrupt request signal is level-triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block-type instruction. During the interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single-step), although the FLAGS register, which is automatically pushed onto the stack, reflects the state of the processor prior to the Interrupt. Until the old FLAGS register is restored, the enable bit will be zero unless specifically set by an instruction.

During the response sequence (Figure 4), the processor executes two successive (back-to-back) interrupt acknowledge cycles. The 8086 emits the LOCK signal from T_2 of the first bus cycle until T_2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interupt system (e.g., 8259A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The INTERRUPT RETURN instruction includes a FLAGS pop, which returns the status of the original interrupt enable bit when it restores the FLAGS.

HALT

When a software "HALT" instruction is executed, the processor indicates that it is entering the "HALT" state in one of two ways depending upon which mode is strapped. In minimum mode, the processor issues one ALE with no qualfying bus control signals. In Maximum Mode, the processor issues appropiate HALT status on $\overline{S_2}\overline{S_1}\overline{S_0}$, and the 8288 bus controller issues one ALE. The 8086 will not leave the "HALT" state when a local bus "hold" is entered while in "HALT." In this case, the processor reissues the HALT indicator. An interrupt request or RESET will force the 8086 out of the "HALT" state.

Read/Modify/Write (Semaphore) Operation Via Lock

The $\overline{LOCK}$ status information is provided by the processor when directly consecutive bus cycles are required during the execution of an instruction. This provides the processor with the capability of performing read/modify/write operations on memory (via the Exchange Register With Memory Instruction, for example) without the possibility of another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The $\overline{LOCK}$ signal is activated (forced LOW) in the clock cycle following the one in which the software "LOCK" prefix instruction is decoded by the EU. It is deactivated at the end of the last bus cycle of the instruction following the "LOCK" prefix instruction. While $\overline{LOCK}$ is active, a request on a RQ/GT pin will be recorded and then honored at the end of the LOCK.

External Synchronization Via Test

As an alternative to the interrupts and general I/O capabilities, the 8086 provides a single software-testable input known as the $\overline{TEST}$ signal. At any time, the program may execute a WAIT instruction. If at that time the $\overline{TEST}$ signal is inactive (HIGH), program execution becomes suspended while the processor waits for $\overline{TEST}$ to become active. It must remain active for at least 5 CLK cycles. The WAIT instruction is re-executed repeatedly until that time. This activity does not consume bus cycles. The processor remains in an idle state while waiting. All 8086 drivers go to three-state OFF if bus "HOLD" is entered. If interrupts are enabled, they may occur while the processor is waiting. When this occurs, the processor fetches the WAIT instruction one extra time, processes the interrupt, and then re-fetches and re-executes the WAIT instruction upon returning from the interrupt.

Basic System Timing

Typical system configurations for the processor operating in minimum mode and in maximum mode are shown in Figures 2a and 2b, respectively. In minimum mode, the processor emits bus control signals in a manner similar to the 8085. In maximum mode, the processor emits coded status information which the 8288 bus controller uses to generate MULTIBUS compatible bus control signals. Figure 3 illustrates the signal timing relationships.

System Timing – Minimum System

The read cycle begins in T_1 with the assertion of the Address Latch Enable (ALE) signal. The trailing (low-going) edge of this signal is used to latch the address information, which is valid on the local bus at this time, into the 8282/8283 latch. The $\overline{BHE}$ and A_0 signals address the low, high, or both bytes. From T_1 to T_4, the M/$\overline{IO}$ signal indicates a memory or I/O operation. At T_2 the address is removed from the local bus and the bus goes to a high impedance state. The read control signal is also asserted at T_2. The read ($\overline{RD}$) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver (8286/8287) is required to buffer the 8086 local bus, signals DT/$\overline{R}$ and $\overline{DEN}$ are provided by the 8086.

A write cycle also begins with the assertion of ALE and the emission of the address. The M/$\overline{IO}$ signal is again asserted to indicate a memory or I/O write operation. In the T_2 immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until the middle of T_4. During T_2, T_3, and T_W, the processor asserts the write control signal. The write ($\overline{WR}$) signal becomes active at the beginning of T_2 as opposed to the read which is delayed somewhat into T_2 to provide time for the bus to float.

The $\overline{BHE}$ and A_0 signals are used to select the proper byte(s) of the memory/IO word to be read or written according to the following table.

$\overline{BHE}$	A_0	Characteristics
0	0	Whole word
0	1	Upper byte from/to odd address
1	0	Lower byte from/to even address
1	1	None

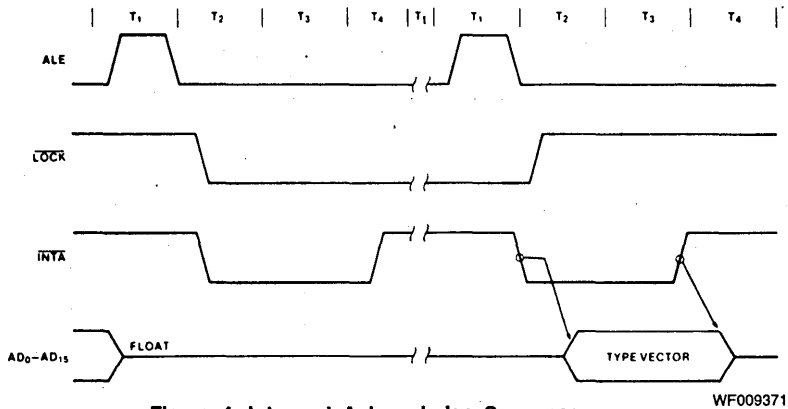
I/O ports are addressed in the same manner as memory location. Even addressed bytes are transferred on the D_7–D_0 bus lines and odd addressed bytes on D_{15}–D_8.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge signal ($\overline{INTA}$) is asserted in place of the read ($\overline{RD}$) signal and the address bus is floated. (See Figure 6.) In the second of two successive INTA cycles, a byte of information is read from bus lines D_7–D_0 as supplied by the interrupt system logic (i.e., 8259A Priority Interrupt Controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into a interrupt vector lookup table, as described earlier.

Bus Timing — Medium Size Systems

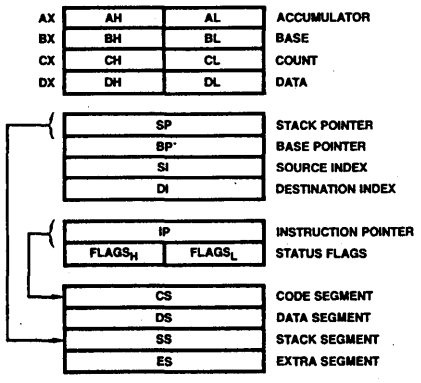
For medium size systems, the MN/$\overline{MX}$ pin is connected to V_{SS}, and the 8288 Bus Controller is added to the system as well as an 8282/8283 latch for latching the system address and a 8286/8287 transceiver to allow for bus loading greater than the 8086 is capable of handling. Signals ALE, DEN, and DT/$\overline{R}$ are generated by the 8288 instead of the processor in this configuration, although their timing remains relatively the same. The 8086 status ($\overline{S}_2$, $\overline{S}_1$, and $\overline{S}_0$) provide type-of-cycle information and become 8288 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 8288 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data isn't valid at the leading edge of write. The 8286/8287 transceiver receives the usual T and OE inputs from the 8288's DT/$\overline{R}$ and DEN.

The pointer into the interrupt vector table, which is passed during the second INTA cycle, can derive from an 8259A located on either the local bus or the system bus. If the master 8259A Priority Interrupt Controller is positioned on the local bus, a TTL gate is required to disable the 8286/8287 transceiver when reading from the master 8259A during the interrupt acknowledge sequence and software "poll."



WF009371

Figure 4. Interrupt Acknowledge Sequence



DF003330

Figure 5. 8086 Register Model

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature Under Bias ..... 0 to 70°C
Voltage on any Pin
with Respect to Ground ..... −1 to +7.0 V
Power Dissipitation ..... 2.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC})
8086 ..... 5 V ± 10%
8086-1, 8086-2 ..... 5 V ± 5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC})
8086 ..... 5 V ± 10%
8086-1, 8086-2 ..... 5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating range

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	+0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 2.5 mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400 μA	2.4		V
I_{CC}	Power Supply Current	All Speeds		340	mA
I_{LI}	Input Leakage Current	$0V \leq V_{IN} \leq V_{CC}$		±10	μA
I_{LO}	Output Leakage Current	$0.45V \leq V_{OUT} \leq V_{CC}$		±10	μA
V_{CL}	Clock Input Low Voltage		−0.5	+0.6	V
V_{CH}	Clock Input High Voltage		3.9	V_{CC} + 1.0	V
C_{IN}	Capacitance of Input Buffer (All input except AD_0-AD_{15}, $\overline{RQ}/\overline{GT}$)	fc = 1 MHz		15	pF
C_{IO}	Capacitance of I/O Buffer (AD_0-AD_{15}, $\overline{RQ}/\overline{GT}$)	fc = 1 MHz		15	pF

SWITCHING CHARACTERISTICS over COMMERCIAL operating range
MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284A (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284A (See Notes 1, 2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8086		118		68		53		ns
TCHRYX	READY Hold Time into 8086		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 3)		−8		−8		−10		ns
THVCH	HOLD Set-up Time		35		20		20		ns
TINVCH	INTR, NMI, $\overline{TEST}$ Set-up Time (See Note 2)		30		15		15		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0 V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8 V		12		12		12	ns

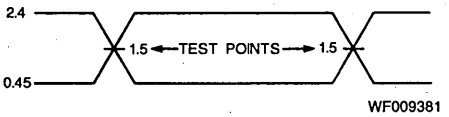
Notes: 1. Signal at 8284A shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T2 state (8ns into T3).

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

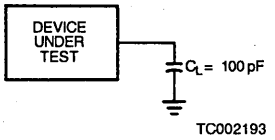
TIMING RESPONSES

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLAV	Address Valid Delay	*C_L = 20-100 pF for all 8086 Outputs (in addition to 8086 self-load). Typical C_L = 100 pF.	10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TLHLL	ALE Width		TCLCH-20		TCLCH-10		TCLCH-10		ns
TCLLH	ALE Active Delay			80		50		40	ns
TCHLL	ALE Inactive Delay			85		55		45	ns
TLLAX	Address Hold Time to ALE Inactive		TCHCL-10		TCHCL-10		TCHCL-10		ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TWHDX	Data Hold Time After WR		TCLCH-30		TCLCH-30		TCLCH-25		ns
TCVCTV	Control Active Delay 1		10	110	10	70	10	50	ns
TCHCTV	Control Active Delay 2		10	110	10	60	10	45	ns
TCVCTX	Control Inactive Delay		10	110	10	70	10	50	ns
TAZRL	Address Float to READ active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL-45		TCLCL-40		TCLCL-35		ns
TCLHAV	HLDA Valid Delay		10	160	10	100	10	60	ns
TRLRH	$\overline{RD}$ Width		2TCLCL-75		2TCLCL-50		2TCLCL-40		ns
TWLWH	$\overline{WR}$ Width		2TCLCL-60		2TCLCL-40		2TCLCL-35		ns
TAVAL	Address Valid to ALE Low		TCLCH-60		TCLCH-40		TCLCH-35		ns
TOLOH	Output Rise Time	From 0.8 to 2.0 V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8 V		12		12		12	ns

SWITCHING TEST INPUT/OUTPUT WAVEFORM



SWITCHING TEST LOAD CIRCUIT



AC Testing inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0." Timing measurements are made at 1.5 V for both a logic "1" and "0."

C_L includes jig capacitance

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating range (continued)
MAX MODE SYSTEM (USING 8288 BUS CONTROLLER)
TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5 V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0 V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284A (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284A (See Notes 1, 2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8086		118		68		53		ns
TCHRYX	READY Hold Time into 8086		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 4)		-8		-8		-10		ns
TINVCH	Set-up Time for Recognition (INTR, NMI, $\overline{TEST}$ (See Note 2)		30		15		15		ns
TGVCH	$\overline{RQ}/\overline{GT}$ Set-up Time		30		15		12		ns
TCHGX	$\overline{RQ}$ Hold Time into 8066		40		30		20		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0 V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8 V		12		12		12	ns

Notes: 1. Signal at 8284A or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8ns into T3).

SWITCHING CHARACTERISTICS over **COMMERCIAL** and **INDUSTRIAL** ranges (continued)
TIMING RESPONSES

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLML	Command Active Delay (See Note 1)	C_L = 20-100 pF for all 8086 Outputs (In addition to 8086 self-load)	10	35	10	35	10	35	ns
TCLMH	Command Inactive Delay (See Note 1)		10	35	10	35	10	35	ns
TRYHSH	READY Active to Status Passive (See Note 3)			110		65		45	ns
TCHSV	Status Active Delay		10	110	10	60	10	45	ns
TCLSH	Status Inactive Delay		10	130	10	70	10	55	ns
TCLAV	Address Valid Delay		10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TSVLH	Status Valid to ALE High (See Note 1)			15		15		15	ns
TSVMCH	Status Valid to MCE High (See Note 1)			15		15		15	ns
TCLLH	CLK Low to ALE Valid (See Note 1)			15		15		15	ns
TCLMCH	CLK Low to MCE High (See Note 1)			15		15		15	ns
TCHLL	ALE Inactive Delay (See Note 1)			15		15		15	ns
TCLMCL	MCE Inactive Delay (See Note 1)			15		15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TCVNV	Control Active Delay (See Note 1)		5	45	5	45	5	45	ns
TCVNX	Control Inactive Delay (See Note 1)		10	45	10	45	10	45	ns
TAZRL	Address Float to Read Active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL-45		TCLCL-40		TCLCL-35		ns
TCHDTL	Direction Control Active Delay (See Note 1)			50		50		50	ns
TCHDTH	Direction Control Inactive Delay (See Note 1)			30		30		30	ns
TCLGL	GT Active Delay		0	85	0	50	0	38	ns
TCLGH	GT Inactive Delay		0	85	0	50	0	45	ns
TRLRH	$\overline{RD}$ Width		2TCLCL-75		2TCLCL-50		2TCLCL-40		ns
TOLOH	Output Rise Time	From 0.8 to 2.0 V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8 V		12		12		12	ns

Notes: 1. Signal at 8284A or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8ns into T3).

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature Under Bias .....0 to 70°C
Voltage on any Pin
with Respect to Ground .....−1 to +7.0 V
Power Dissipation .....2.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Military (M) Devices
Temperature (T_C) ..... −55 to + 125°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over **MILITARY** operating range (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IL} †	Input LOW Voltage	V_{CC} = Min. & Max.	−0.5*	+0.8	V
V_{IH} †	Input HIGH Voltage	V_{CC} = Min. & Max.	2.0	V_{CC} + 0.5*	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA, V_{CC} = Min.		0.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = −400 μA, V_{CC} = Min.	2.4		V
I_{CC}	Power Supply Current (Note 1)	T_C = 25°C, V_{CC} = Max.		340	mA
I_{LI}	Input Leakage Current	V_{CC} = Max., V_{IN} = 5.5 V & 0 V	−10	10	μA
I_{LO} ††	Output Leakage Current	V_{CC} = Max., V_{OUT} = 5.5 V & 0.45 V	−10	10	μA
V_{CL} †	Clock Input LOW Voltage	V_{CC} = Min. & Max.	−0.5*	+0.6	V
V_{CH} †	Clock Input HIGH Voltage	V_{CC} = Min. & Max.	3.9	V_{CC} + 1.0*	V
C_{IN} †††	Capacitance of Input Buffer (All Input Except AD_0–AD_{15}, $\overline{RQ}/\overline{GT}$)	fc = 1 MHz		20*	pF
C_{IO} †††	Capacitance of I/O Buffer (AD_0–AD_{15}, $\overline{RQ}/\overline{GT}$)	fc = 1 MHz		20*	pF

* Guaranteed by design; not tested.
† Group A, Subgroups 7 and 8 only are tested.
†† Group A, Subgroups 1 and 2 only are tested.
††† Not included in Group A tests.

Notes: 1. I_{CC} is measured while running a functional pattern with spec value I_{OL}/I_{OH} loads applied.

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	8086		8086-2		Unit
			Min.	Max.	Min.	Max.	
TCLCL	CLK Cycle Period (Note 11)		200	500	125	500	ns
TCLCH	CLK LOW Time		118		68		ns
TCHCL	CLK HIGH Time		69		44		ns
TCH1CH2	CLK Rise Time (Note 5)	From 1.0 to 3.5 V		10		10	ns
TCL2CL1	CLK Fall Time (Note 5)	From 3.5 to 1.0 V		10		10	ns
TDVCL	Data in Setup Time		30		20		ns
TCLDX	Data in Hold Time		10		10		ns
TR1VCL	RDY Setup Time into 8284A (Notes 1 & 2)		35		35		ns
TCLR1X	RDY Hold Time into 8284A (Notes 1 & 2)		0		0		ns
TRYHCH	READY Setup Time into 8086		118		68		ns
TCHRYX	READY Hold Time into 8086		30		20		ns
TRYLCL	READY Inactive to CLK (Note 4)		-8		-8		ns
THVCH	HOLD Setup Time		35		20		ns
TINVCH	INTR, NMI, $\overline{\text{TEST}}$ Setup Time (Note 2)		30		15		ns
TILIH	Input Rise Time (Except CLK) (Note 5)	From 0.8 to 2.0 V		2 0		20	ns
TIHIL	Input Fall Time (Except CLK) (Note 5)	From 2.0 to 0.8 V		12		12	ns

Notes:
1. Signal at 8284A and 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8 ns into T3).
5. Not tested; these specs are controlled by the Teradyne J941 tester.
6. V_{CC} = 4.5 V, 5.5 V V_{IH} = 2.4 V
   V_{IL} = .45 V V_{IHC} = 4.3 V
   V_{ILC} = .25 V V_{OH} = 1.6 V
   V_{OL} = 1.4 V
7. Minimum spec tested at V_{CC} Max. (5.5 V) only.
8. Maximum spec tested at V_{CC} Min. (4.5 V) only.
9. Tested at V_{CC} Max. (5.5 V) only.
10. Tested at V_{CC} Min. (4.5 V) only.
11. Test conditions for TCLCL Max. are:
    V_{CC} = 4.5 V V_{OL} = 1 V
    V_{IL} = 0 V V_{IH} = 4 V
    V_{ILC} = 0 V V_{IHC} = 5 V

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)

TIMING RESPONSES

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	8086		8086-2		Unit
			Min.	Max.	Min.	Max.	
TCLAV	Address Valid Delay	C_L = 100 pF for all 8086 Outputs (in addition to 8086 internal loads)	10	110	10	60	ns
TCLAX	Address Hold Time (Notes 7 & 8)		10		10		ns
TCLAZ	Address Float Delay (Note 8)		10	80	10	50	ns
TLHLL	ALE Width (Note 10)		98		58		ns
TCLLH	ALE Active Delay (Note 8)			80		50	ns
TCHLL	ALE Inactive Delay (Note 8)			85		55	ns
TLLAX	Address Hold Time to ALE Inactive (Note 7)		59		34		ns
TCLDV	Data Valid Delay (Note 8)		10	110	10	60	ns
TCHDX	Data Hold Time (Note 10)		10		10		ns
TWHDX	Data Hold Time After WR (Note 9)		88		38		ns
TCVCTV	Control Active Delay 1 (Note 8)		10	110	10	70	ns
TCHCTV	Control Active Delay 2 (Note 8)		10	110	10	60	ns
TCVCTX	Control Inactive Delay (Note 8)		10	110	10	70	ns
TAZRL	Address Float to READ Active (Note 9)		0		0		ns
TCLRL	$\overline{RD}$ Active Delay (Note 8)		10	165	10	100	ns
TCLRH	$\overline{RD}$ Inactive Delay (Note 8)		10	150	10	80	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active (Note 10)		155		85		ns
TCLHAV	HLDA Valid Delay (Note 8)		10	160	10	100	ns
TRLRH	$\overline{RD}$ Width (Note 10)		325		200		ns
TWLWH	$\overline{WR}$ Width (Note 10)		340		210		ns
TAVAL	Address Valid to ALE LOW (Note 9)		58		28		ns
TOLOH	Output Rise Time (Note 9)	From 0.8 to 2.0 V		20		20	ns
TOHOL	Output Fall Time (Note 9)	From 2.0 to 0.8 V		12		12	ns

Notes:
1. Signal at 8284A and 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8 ns into T3).
5. Not tested; these specs are controlled by the Teradyne J941 tester.
6. V_{CC} = 4.5 V, 5.5 V V_{IH} = 2.4 V
   V_{IL} = .45 V V_{IHC} = 4.3 V
   V_{ILC} = .25 V V_{OH} = 1.6 V
   V_{OL} = 1.4 V
7. Minimum spec tested at V_{CC} Max. (5.5 V) only.
8. Maximum spec tested at V_{CC} Min. (4.5 V) only.
9. Tested at V_{CC} Max. (5.5 V) only.
10. Tested at V_{CC} Min. (4.5 V) only.
11. Test conditions for TCLCL Max. are:
    V_{CC} = 4.5 V V_{OL} = 1 V
    V_{IL} = 0 V V_{IH} = 4 V
    V_{ILC} = 0 V V_{IHC} = 5 V

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)

MAX MODE SYSTEM (USING 8288 BUS CONTROLLER) TIMING REQUIREMENTS

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	8086		8086-2		Unit
			Min.	Max.	Min.	Max.	
TCLCL	CLK Cycle Period (Note 11)		200	500	125	500	ns
TCLCH	CLK LOW Time		118		68		ns
TCHCL	CLK HIGH Time		69		44		ns
TCH1CH2	CLK Rise Time (Note 5)	From 1.0 to 3.5 V		10		10	ns
TCL2CL1	CLK Fall Time (Note 5)	From 3.5 to 1.0 V		10		10	ns
TDVCL	Data in Setup Time		30		20		ns
TCLDX	Data in Hold Time		10		10		ns
TR1VCL	RDY Setup Time into 8284A (Notes 1 & 2)		35		35		ns
TCLR1X	RDY Hold Time into 8284A (Notes 1 & 2)		0		0		ns
TRYHCH	READY Setup Time into 8086		118		68		ns
TCHRYX	READY Hold Time into 8086		30		20		ns
TRYLCL	READY Inactive to CLK (Note 4)		-8		-8		ns
TINVCH	Setup Time for Recognition (INTR, NMI, $\overline{TEST}$) (Note 2)		30		15		ns
TGVCH	$\overline{RQ}/\overline{GT}$ Setup Time		30		15		ns
TCHGX	$\overline{RQ}$ Hold Time into 8066		40		30		ns
TILIH	Input Rise Time (Except CLK) (Note 5)	From 0.8 to 2.0 V		20		20	ns
TIHIL	Input Fall Time (Except CLK) (Note 5)	From 2.0 to 0.8 V		12		12	ns

Notes:
1. Signal at 8284A and 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8 ns into T3).
5. Not tested; these specs are controlled by the Teradyne J941 tester.
6. V_{CC} = 4.5 V, 5.5 V V_{IH} = 2.4 V
   V_{IL} = .45 V V_{IHC} = 4.3 V
   V_{ILC} = .25 V V_{OH} = 1.6 V
   V_{OL} = 1.4 V
7. Minimum spec tested at V_{CC} Max. (5.5 V) only.
8. Maximum spec tested at V_{CC} Min. (4.5 V) only.
9. Tested at V_{CC} Max. (5.5 V) only.
10. Tested at V_{CC} Min. (4.5 V) only.
11. Test conditions for TCLCL Max. are:
    V_{CC} = 4.5 V V_{OL} = 1 V
    V_{IL} = 0 V V_{IH} = 4 V
    V_{ILC} = 0 V V_{IHC} = 5 V

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)

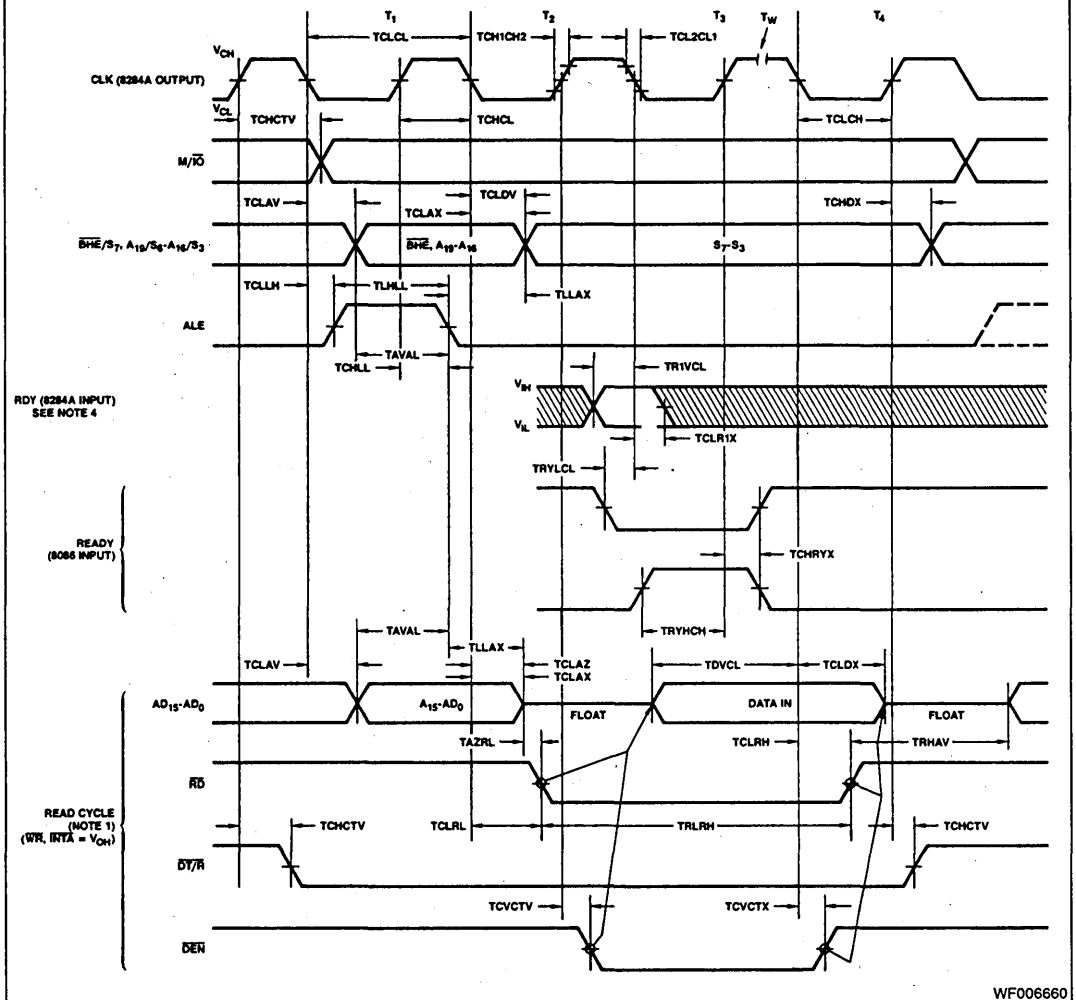
TIMING RESPONSES

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	8086		8086-2		Unit
			Min.	Max.	Min.	Max.	
TCLML	Command Active Delay (Note 1)	C_L = 100 pF for all 8086 Outputs (In addition to 8086 internal loads)	10	35	10	35	ns
TCLMH	Command Inactive Delay (Note 1)		10	35	10	35	ns
TRYHSH	READY Active to Status Passive (Note 3)			110		65	ns
TCHSV	Status Active Delay (Notes 7 & 8)		10	110	10	60	ns
TCLSH	Status Inactive Delay		10	130	10	70	ns
TCLAV	Address Valid Delay		10	110	10	60	ns
TCLAX	Address Hold Time		10		10		ns
TCLAZ	Address Float Delay		10	80	10	50	ns
TSVLH	Status Valid to ALE HIGH (Note 1)			15		15	ns
TSVMCH	Status Valid to MCE HIGH (Note 1)			15		15	ns
TCLLH	CLK LOW to ALE Valid (Note 1)			15		15	ns
TCLMCH	CLK LOW to MCE HIGH (Note 1)			15		15	ns
TCHLL	ALE Inactive Delay (Note 1)			15		15	ns
TCLMCL	MCE Inactive Delay (Note 1)			15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	ns
TCHDX	Data Hold Time		10		10		ns
TCVNV	Control Active Delay (Note 1)		5	45	5	45	ns
TCVNX	Control Inactive Delay (Note 1)		10	45	10	45	ns
TAZRL	Address Float to Read Active		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		155		85		ns
TCHDTL	Direction Control Active Delay (Note 1)			50		50	ns
TCHDTH	Direction Control Inactive Delay (Note 1)			30		30	ns
TCLGL	GT Active Delay (Note 8)		0	85	0	50	ns
TCLGH	GT Inactive Delay (Note 8)		0	85	0	50	ns
TRLRH	$\overline{RD}$ Width		325		200		ns
TOLOH	Output Rise Time	From 0.8 to 2.0 V		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8 V		12		12	ns

Notes:
1. Signal at 8284A and 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8 ns into T3).
5. Not tested; these specs are controlled by the Teradyne J941 tester.
6. V_{CC} = 4.5 V, 5.5 V V_{IH} = 2.4 V
   V_{IL} = .45 V V_{IHC} = 4.3 V
   V_{ILC} = .25 V V_{OH} = 1.6 V
   V_{OL} = 1.4 V
7. Minimum spec tested at V_{CC} Max. (5.5 V) only.
8. Maximum spec tested at V_{CC} Min. (4.5 V) only.
9. Tested at V_{CC} Max. (5.5 V) only.
10. Tested at V_{CC} Min. (4.5 V) only.
11. Test conditions for TCLCL Max. are:
    V_{CC} = 4.5 V V_{OL} = 1 V
    V_{IL} = 0 V V_{IH} = 4 V
    V_{ILC} = 0 V V_{IHC} = 5 V

SWITCHING WAVEFORMS

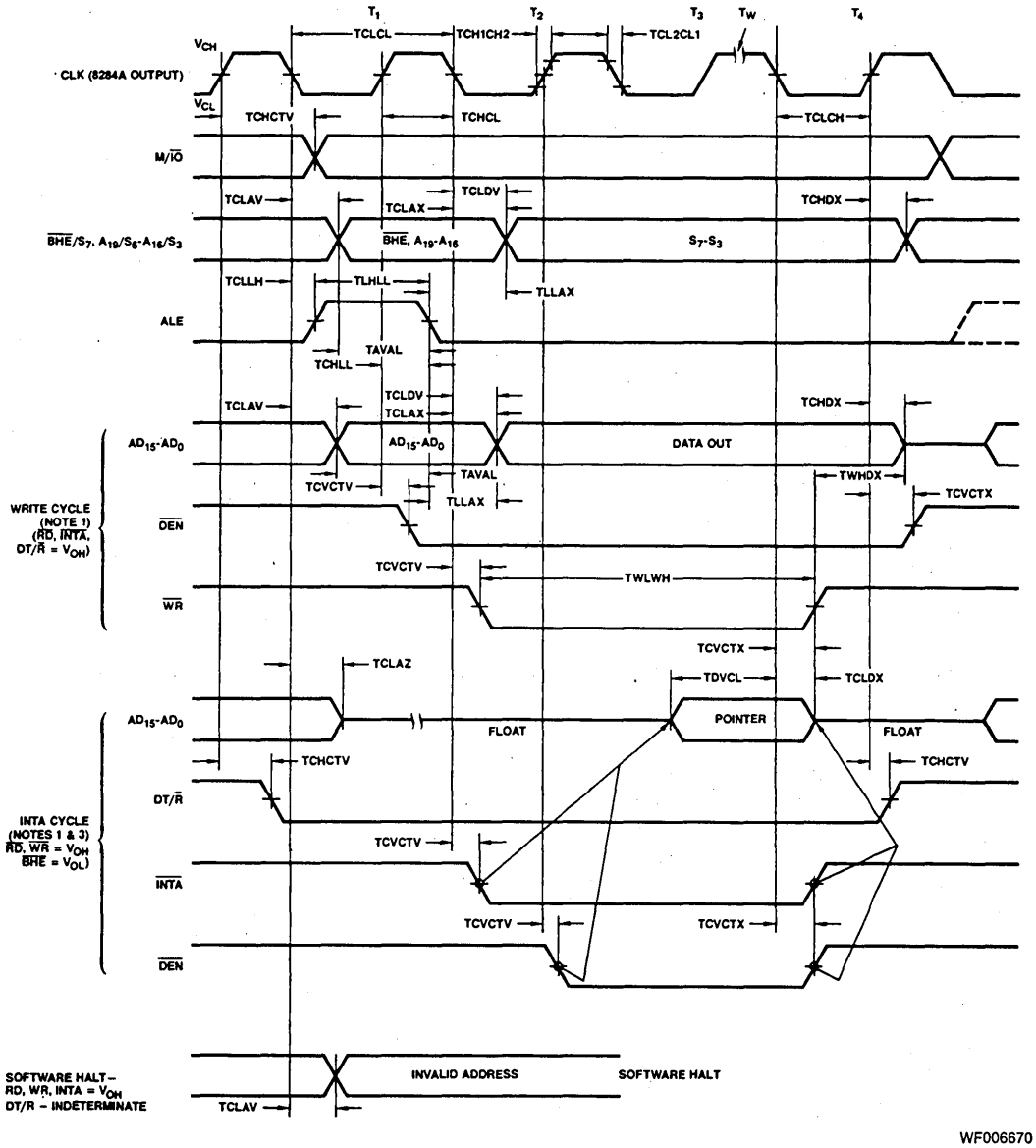
MINIMUM MODE



WF006660

SWITCHING WAVEFORMS (continued)

MINIMUM MODE

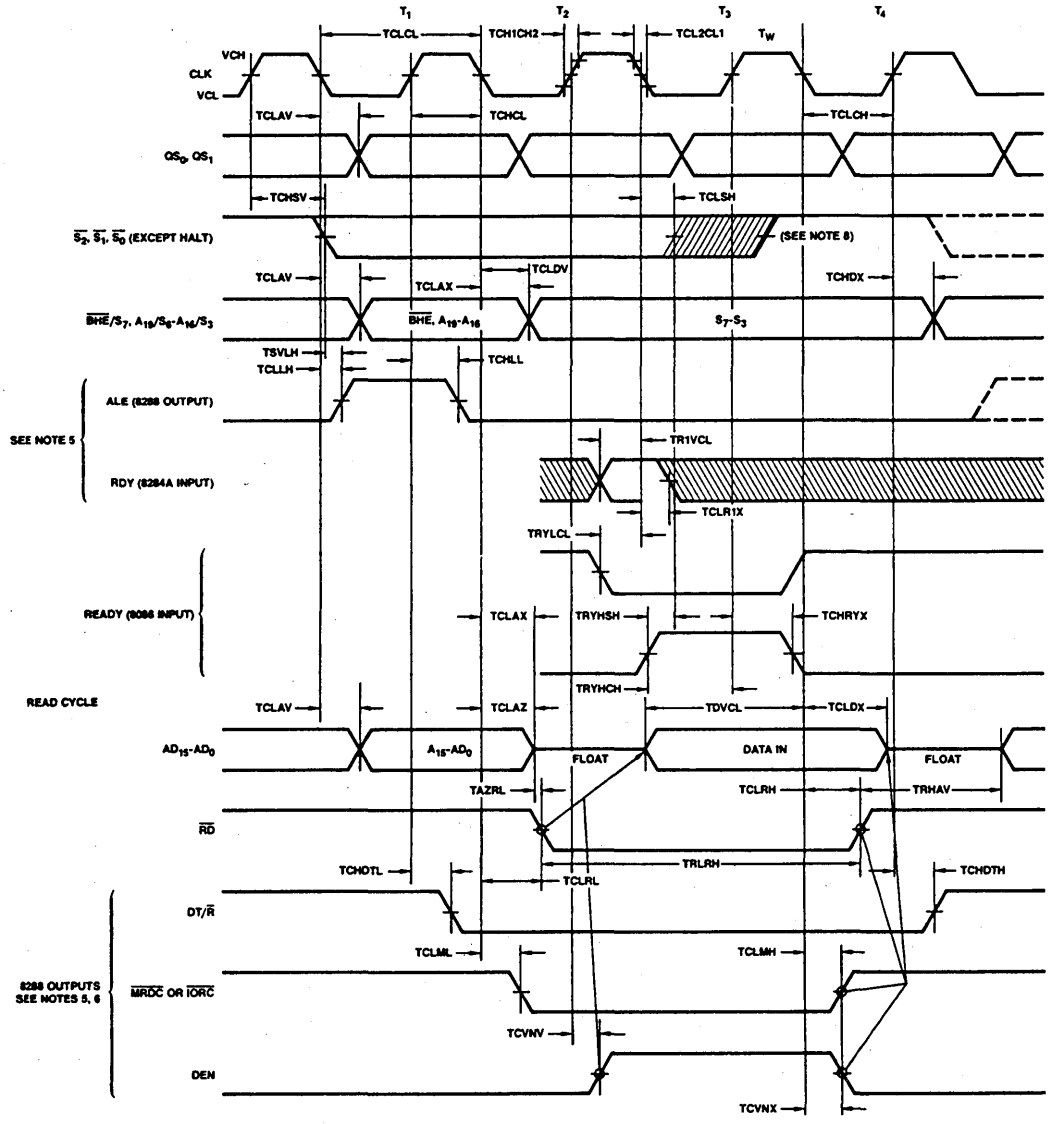


WF006670

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Two INTA cycles run back-to-back. The 8086 LOCAL ADDR/DATA BUS is floating during both INTA cycles. Control signals are shown for second INTA cycle.
4. Signals at 8284A are shown for reference only.
5. All timing measurements are made at 1.5 V unless otherwise noted.

SWITCHING WAVEFORMS (continued)

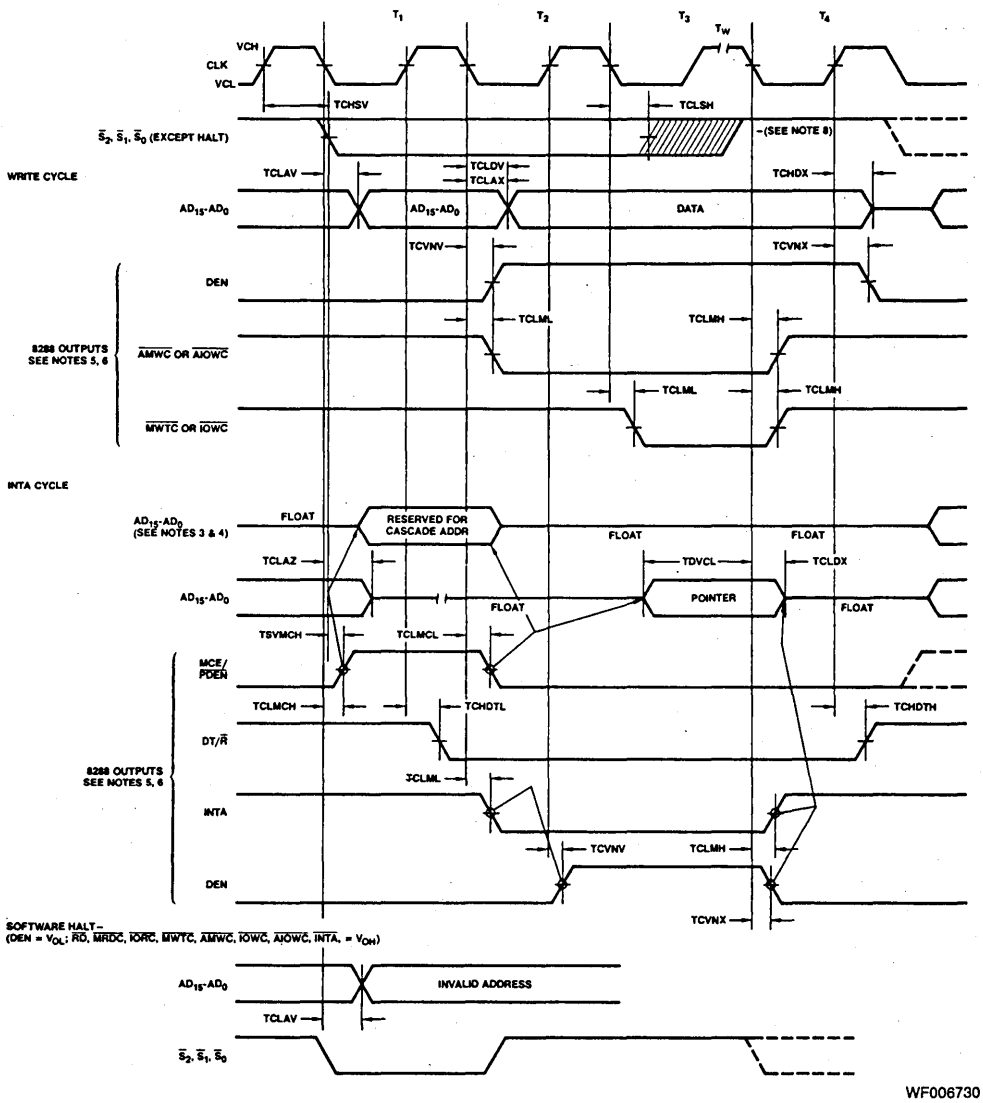
MAXIMUM MODE



WF006680

SWITCHING WAVEFORMS (continued)

MAXIMUM MODE (continued)

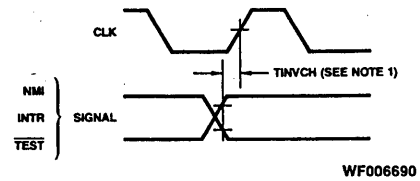


WF006730

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Cascade address is valid between first and second INTA cycle.
4. Two INTA cycles run back-to-back. The 8086 LOCAL ADDR/DATA BUS is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.
5. Signals at 8284A or .8288 are shown for reference only.
6. The issuance of the 8288 command and control signals ($\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$, $\overline{INTA}$ and DEN) lags the active high 8288 CEN.
7. All timing measurements are made at 1.5 V unless otherwise noted.
8. Status inactive in state just prior to T_4.

SWITCHING WAVEFORMS (continued)

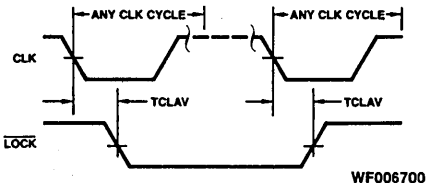
ASYNCHRONOUS SIGNAL RECOGNITION



WF006690

Note: Set-up Requirements for Asynchronous signals only to guarantee recognition at next CLK.

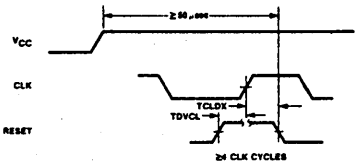
BUS LOCK SIGNAL TIMING



WF006700

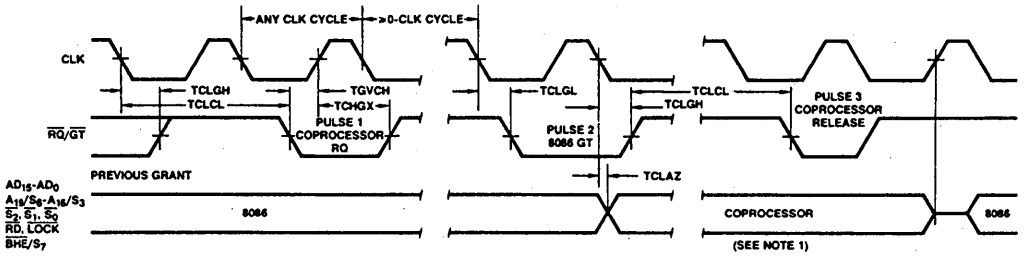
(MAXIMUM MODE ONLY)

RESET TIMING



WF009530

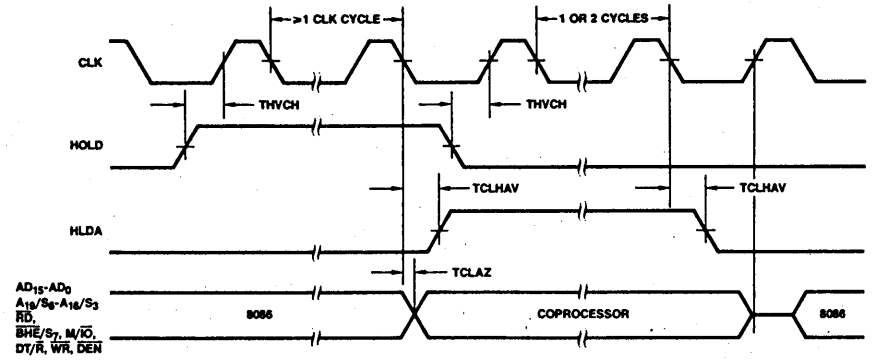
REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)



WF006710

Note: The Coprocessor may not drive the buses outside the region shown without risking contention.

HOLD/HOLD ACKNOWLEDGE TIMING (MINIMUM MODE ONLY)



WF006720

8086/8088
INSTRUCTION SET SUMMARY

DATA TRANSFER

MOV = Move	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/memory to /from register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate to register	1 0 1 1 w reg	data	data if w = 1	
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
PUSH = Push:				
Register/memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment register	0 0 0 reg 1 1 0			
POP = Pop:				
Register/memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment register	0 0 0 reg 1 1 1			
XCHG = Exchange:				
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m		
Register with accumulator	1 0 0 1 0 reg			
IN = Input from:				
Fixed port	1 1 1 0 0 1 0 w	port		
Variable port	1 1 1 0 1 1 0 w			
OUT = Ouput to:				
Fixed port	1 1 1 0 0 1 1 w	port		
Variable port	1 1 1 0 1 1 1 w			
XLAT = Transtate byte to AL	1 1 0 1 0 1 1 1			
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m		
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m		
LANF = Load AH with flags	1 0 0 1 1 1 1 1			
SANF = Store AH into flags	1 0 0 1 1 1 1 0			
PUSHF = Push flags	1 0 0 1 1 1 0 0			
POPF = Pop flags	1 0 0 1 1 1 0 1			

.Mnemonics ©Intel, 1978

INSTRUCTION SET SUMMARY (continued)

ARITHMETIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
ADD = Add				
Reg/memory with register to either	0 0 0 0 0 0 d w	mod reg r/m		
Immediate to register / memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w = 1	
ADC = Add with carry:				
Reg/memory with register to either	0 0 0 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w = 1	
INC = Increment:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
Register	0 1 0 0 0 reg			
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1			
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1			
SUB = Subtract:				
Reg/memory and register to either	0 0 1 0 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 1 0 1 1 0 w	data	data if w = 1	
SBB = Subtract with borrow:				
Reg/memory and register to either	0 0 0 1 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w = 1	
DEC = Decrement:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m		
Register	0 1 0 0 1 reg			
NEG Change sign	1 1 1 1 0 1 1 w	mod 0 1 1 r/m		
CMP = Compare:				
Register/memory with register	0 0 1 1 1 0 1 w	mod reg r/m		
Register with register/memory	0 0 1 1 1 0 0 w	mod reg r/m		
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s:w = 01
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w = 1	
AAS ASCII adjust for subtract	0 0 1 1 1 1 1 1			
DAS Decimal adjust for subtract	0 0 1 0 1 1 1 1			
MUL Mulitiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m		
IMUL Integer multiply (signed):	1 1 1 1 0 1 1 w	mod 1 0 1 r/m		
AAM ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0		
DIV Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m		
IDIV Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m		
AAD ASCH adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0		
CBW Convert byte to word	1 0 0 1 1 0 0 0			
CWD Convert word to double word	1 0 0 1 1 0 0 1			

Mnemonics ©Intel, 1978

INSTRUCTION SET SUMMARY (continued)

LOGIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
NOT Invert	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		
SHL/SAL Shift logical/arithmetic left	1 1 0 1 0 0 v w	mod 1 0 0 r/m		
SHR Shift logical right	1 1 0 1 0 0 v w	mod 1 0 1 r/m		
SAR Shift arithmetic right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
ROL Rotate left	1 1 0 1 0 0 v w	mod 0 0 0 r/m		
ROR Rotate right	1 1 0 1 0 0 v w	mod 0 0 1 r/m		
RCL Rotate through carry flag left	1 1 0 1 0 0 v w	mod 0 1 0 r/m		
RCR Rotate through carry right	1 1 0 1 0 0 v w	mod 0 1 1 r/m		
AND = And:				
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1	
TEST = And function to flags, no result:				
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m		
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1	
OR = Or:				
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1	
XOR = Exclusive or:				
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1	

STRING MANIPULATION:

REP = Repeat	1 1 1 1 0 0 1 z
MOVS = Move byte/word	1 0 1 0 0 1 0 w
CMPS = Compare byte/word	1 0 1 0 0 1 1 w
SCAS = Scan byte/word	1 0 1 0 1 1 1 w
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w

Mnemonics ©Intel, 1978

INSTRUCTION SET SUMMARY (continued)

CONTROL TRANSFER

CALL = Call	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp-low	disp-high	
indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0 r/m		
Direct intersegment	1 0 0 1 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m		
JMP = Unconditional jump:				
Direct within segment	1 1 1 0 1 0 0 1	disp-low	disp-high	
Direct within segment-short	1 1 1 0 1 0 1 1	disp		
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0 r/m		
Direct intersegment	1 1 1 0 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m		
RET = Return from CALL:				
Within segment	1 1 0 0 0 0 1 1			
Within seg adding immed to SP	1 1 0 0 0 0 1 0	data-low	data-high	
Intersegment	1 1 0 0 1 0 1 1			
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low	data-high	
JE/JZ = Jump on equal/zero	0 1 1 1 0 1 0 0	disp		
JL/JNGE = Jump on less/not greater or equal	0 1 1 1 1 1 0 0	disp		
JLE/JNG = Jump on less or equal/not greater	0 1 1 1 1 1 1 0	disp		
JB/JNAE = Jump on below/not above or equal	0 1 1 1 0 0 1 0	disp		
JBE/JNA = Jump on below or equal/not above	0 1 1 1 0 1 1 0	disp		
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0	disp		
JO = Jump on overflow	0 1 1 1 0 0 0 0	disp		
JS = Jump on sign	0 1 1 1 1 0 0 0	disp		
JNE/JNZ = Jump on not equal/not zero	0 1 1 1 0 1 0 1	disp		
JNL/JGE = Jump on not less/greater or equal	0 1 1 1 1 1 0 1	disp		
JNLE/JG = Jump on not less or equal/greater	0 1 1 1 1 1 1 1	disp		
JNB/JAE = Jump on not below/above or equal	0 1 1 1 0 0 1 1	disp		
JNBE/JA = Jump on not below or equal/above	0 1 1 1 0 1 1 1	disp		
JNP/JPO = Jump on not par/par odd	0 1 1 1 1 0 1 1	disp		
JNO = Jump on not overflow	0 1 1 1 0 0 0 1	disp		
JNS = Jump on not sign	0 1 1 1 1 0 0 1	disp		
LOOP = Loop CX times	1 1 1 0 0 0 1 0	disp		
LOOPZ/LOOPE = Loop while zero/equal	1 1 1 0 0 0 0 1	disp		
LOOPNZ/LOOPNE = Loop while not zero/equal	1 1 1 0 0 0 0 0	disp		
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1	disp		

Mnemonics ©Intel, 1978

INSTRUCTION SET SUMMARY (continued)

CONTROL TRANSFER (Cont'd.)

INT = Interrupt	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Type specified	1 1 0 0 1 1 0 1	type		
Type 3	1 1 0 0 1 1 0 0			
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0			
IRET = Interrupt return	1 1 0 0 1 1 1 1			

PROCESSOR CONTROL

CLC = Clear carry	1 1 1 1 1 0 0 0	
CMC = Complement carry	1 1 1 1 0 1 0 1	
STC = Set carry	1 1 1 1 1 0 0 1	
CLD = Clear direction	1 1 1 1 1 1 0 0	
STD = Set direction	1 1 1 1 1 1 0 1	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	
STI = Set interrupt	1 1 1 1 1 0 1 1	
HLT = Halt	1 1 1 1 0 1 0 0	
WAIT = Wait	1 0 0 1 1 0 1 1	
ESC = Processor Extension Escape	1 1 0 1 1 x x x	mod x x x r/m
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	

Footnotes:

AL = 8-bit accumulator
AX = 16-bit accumulator
CX = Count register
DS = Data segment
ES = Extra segment
Above/below refers to unsigned value.
Greater = more positive.
Less = less positive (more negative) signed values
if d = 1 then "to" reg; if d = 0 then "from" reg
w = 1 then word instruction; if w = 0 then byte instruction

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

if s:w = 01 then 16 bits of immediate data form the operand.
if s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand.
if v = 0 then "count" = 1; if v = 1 then "count" in (CL)
x = don't care
z is used for string primitives for comparison with ZF Flag.

SEGMENT OVERRIDE PREFIX

0	0	1	reg	1	1	0

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register files as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS = X:X:X:X:(OF):(DF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

Mnemonics ©Intel, 1978

8088

8-Bit Microprocessor CPU
iAPX86 Family
FINAL

DISTINCTIVE CHARACTERISTICS

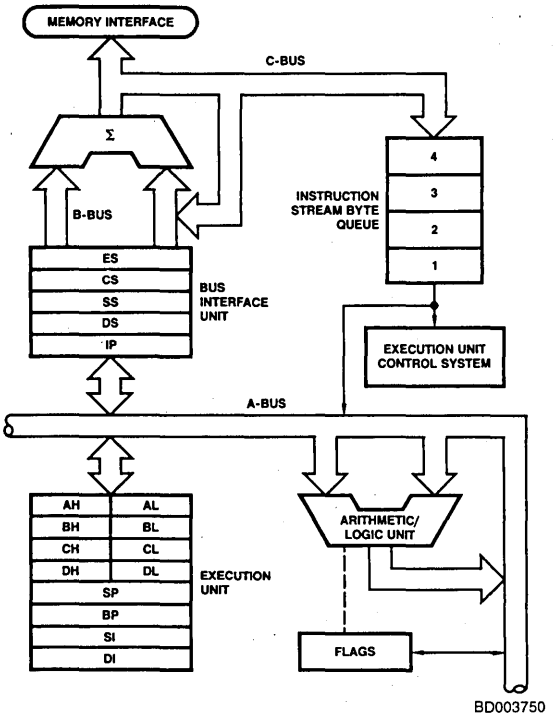
- 8-bit data bus, 16-bit internal architecture
- Directly addresses 1 Mbyte of memory
- Software compatible with 8086 CPU
- Byte, word, and block operations
- 24 operand addressing modes
- Powerful instruction set
- Efficient high level language implementation
- Three speed options: 5MHz 8088
  8MHz 8088-2
  10MHz 8088-1

GENERAL DESCRIPTION

The 8088 CPU is an 8-bit processor designed around the 8086 internal structure. Most functions of the 8088 are identical to the equivalent 8086 functions. The pinout is slightly different. The 8088 handles the external bus the same way the 8086 does, but it handles only 8 bits at a time. Sixteen-bit words are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time.

The 8088 is made with N-channel silicon gate technology and is packaged in a 40-pin Plastic dip, CERDIP or Plastic Leaded Chip Carrier.

BLOCK DIAGRAM

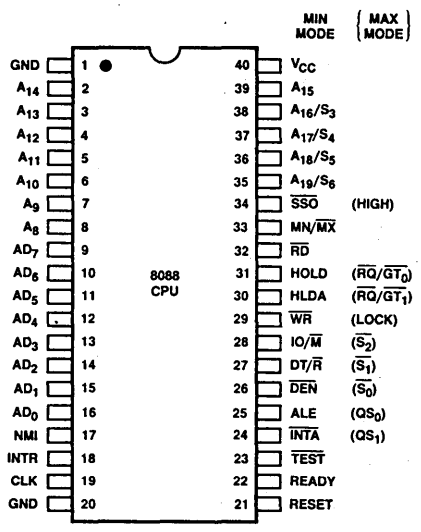


Publication # 02338 | Rev. D | Amendment /0
Issue Date: August 1989

CONNECTION DIAGRAMS

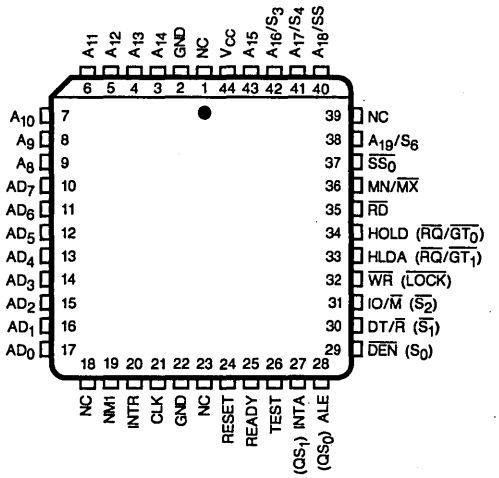
Top View

DIPs



CD005520

PLCC



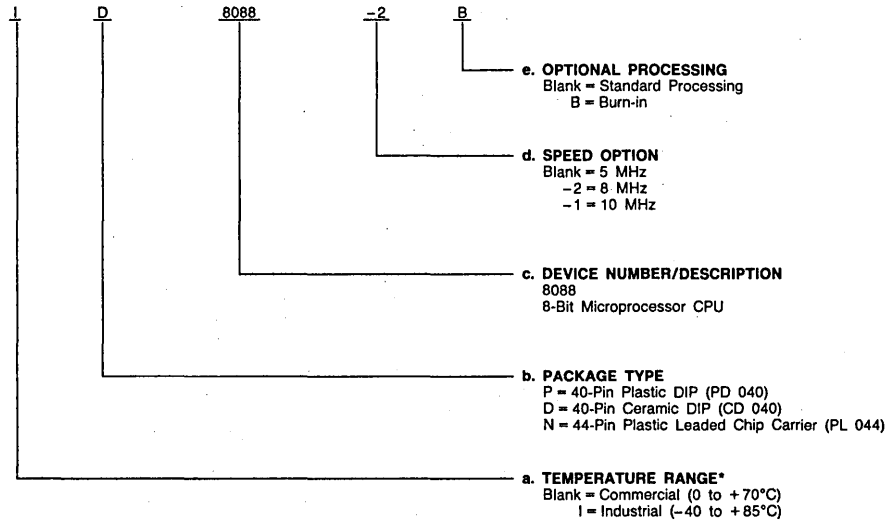
CD010680

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



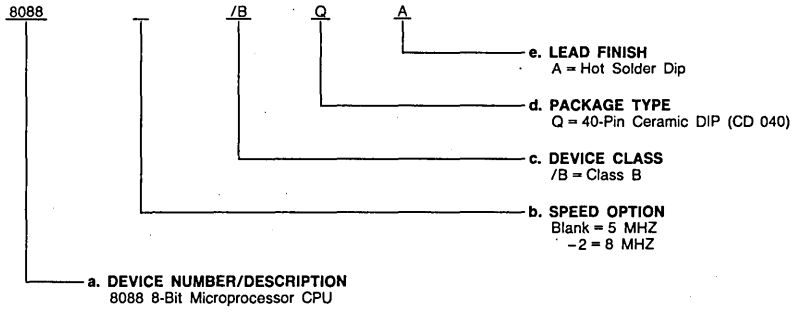
Valid Combinations	
P, N	8088
	8088-2
	8088-1
D	8088B, 8088
	8088-2B, 8088-2
	8088-1B
ID	8088B
	8088-2B

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

MILITARY ORDERING INFORMATION
APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
8088	/BQA
8088-2	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

The following pin function descriptions are for 8088 systems in either minimum or maximum mode. The "local bus" in these descriptions is the direct multiplexed bus interface connection to the 8088 (without regard to additional bus buffers).

Pin No.*	Name	I/O	Description
9-16	AD_7–AD_0	I/O	Address Data Bus. These lines constitute the time multiplexed memory/IO address (T_1) and data (T_2, T_3, T_W and T_4) bus. These lines are active HIGH and float to three-state OFF during interrupt acknowledge and local bus "hold acknowledge."
39, 2-8	A_{15}–A_8	O	Address Bus. These lines provide address bits 8 through 15 for the entire bus cycle (T_1–T_4). These lines do not have to be latched by ALE to remain valid. A_{15}–A_8 are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge."
35-38	A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3	O	Address/Status. During T_1, these are the four most significant address lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information is available on these lines during T_2, T_3, T_W and T_4. S_6 is always LOW. The status of the interrupt enable flat bit (S_5) is updated at the beginning of each clock cycle. S_4 and S_3 are encoded as shown. This information indicates which segment register is presently being used for data accessing. These lines float to three-state OFF during local bus "hold acknowledge."
32	$\overline{RD}$	O	Read. Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the IO/$\overline{M}$ pin or S_2. This signal is used to read devices which reside on the 8088 local bus. $\overline{RD}$ is active LOW during T_2, T_3 and T_W of any read cycle and is guaranteed to remain HIGH in T_2 until the 8088 local bus has floated. This signal floats to 3-state OFF in "hold acknowledge."
22	READY	I	READY. The acknowledgment from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 8284 clock generator to form READY. This signal is active HIGH. The 8088 READY input is not synchronized. Correct operation is not guaranteed if the set-up and hold times are not met.
18	INTR	I	Interrupt Request. A level-triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
23	$\overline{TEST}$	I	$\overline{TEST}$. Input is examined by the "wait for test" instruction. If the $\overline{TEST}$ input is LOW, execution continues; otherwise, the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
17	NMI	I	Non-Maskable Interrupt. An edge-triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
21	RESET	I	RESET. Causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.
19	CLK	I	Clock. Provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
40	V_{CC}		V_{CC}. The +5 V ±10% power supply pin.
1, 20	GND		GND. The ground pins.
33	MIN/$\overline{MX}$	I	Minimum/Maximum. Indicates what mode the processor is to operate in. The two modes are discussed in the following sections.
28	IO/$\overline{M}$	O	Status Line. An inverted maximum mode $\overline{S_2}$. It is used to distinguish a memory access from an I/O access. IO/$\overline{M}$ becomes valid in the T_4 preceding a bus cycle and remains valid until the final T_4 of the cycle (I/O = HIGH, M = LOW). IO/$\overline{M}$ floats to three-state OFF in local bus "hold acknowledge."
29	$\overline{WR}$	O	Write. Strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/$\overline{M}$ signal. WR is active for T_2, T_3 and T_W of any write cycle. It is active LOW and floats to 3-state OFF in local bus "hold acknowledge."
24	$\overline{INTA}$	O	INTA. Used as a read strobe for interrupt acknowledge cycles. It is active LOW during T_2, T_3 and T_W of each interrupt acknowledge cycle.
25	ALE	O	Address Latch Enable. Provided by the processor to latch the address into 8282/8283 address latch. It is a HIGH pulse active during clock low of T_1 of any bus cycle. Note that ALE is never floated.
27	DT/$\overline{R}$	O	Data Transmit/Receive. Needed in a minimum system that desires to use an 8286/8287 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically DT/$\overline{R}$ is equivalent to $\overline{S_1}$ in the maximum mode, and its timing is the same as for IO/$\overline{M}$ (T = HIGH, R = LOW.) This signal floats to three-state OFF in local bus "hold acknowledge."
26	$\overline{DEN}$	O	Data Enable. Provided as an output enable for the 8286/8287 in a minimum system that uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access and for $\overline{INTA}$ cycles. For a read or $\overline{INTA}$ cycle, it is active from the middle of T_2 until the middle of T_4; while for a write cycle, it is active from the beginning of T_2 until the middle of T_4. $\overline{DEN}$ floats to 3-state OFF during local bus "hold acknowledge."

S_4	S_3	Characteristics
0 (LOW)	0	Alternate Data
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data
S_6 is 0 (LOW)		

*Pin numbers correspond to DIPs only.

PIN DESCRIPTION (continued)

Pin No.*	Name	I/O	Description
31, 30	HOLD, HLDA	I/O	HOLD. Indicates that another master is requesting a local bus "hold." To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgment in the middle of a T_4 or T_1 clock cycle. Simultaneous with the issuance of HLDA, the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. HOLD is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the set-up time.
34	$\overline{SSO}$	O	Status Line. Logically equivalent to $\overline{SO}$ in the maximum mode. The combination of $\overline{SSO}$, IO/$\overline{M}$ and DT/$\overline{R}$ allows the system to completely decode the current bus cycle status.

IO/$\overline{M}$	DT/$\overline{R}$	$\overline{SSO}$	Characteristics
1 (HIGH)	0	0	Interrupt Acknowledge
1	0	1	Read I/O port
1	1	0	Write I/O port
1	1	1	Halt
0 (LOW)	0	0	Code Access
0	0	1	Read memory
0	1	0	Write memory
0	1	1	Passive

28-26	$\overline{S}_2$, $\overline{S}_1$, $\overline{S}_0$	O	Status. Active during clock high of T_4, T_1 and T_2 and is returned to the passive state (1, 1, 1) during T_3 or during T_W when READY is HIGH. This status is used by the 8288 bus controller to generate all memory and I/O access control signals. Any change by $\overline{S}_2$, $\overline{S}_1$ or $\overline{S}_0$ during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle. These signals float to three-state OFF during "hold acknowledge." During the first clock cycle after RESET becomes active, these signals are active HIGH. After this first clock, they float to three-state OFF.

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1 (HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

31, 30	$\overline{RQ}/\overline{GT}_0$, $\overline{RQ}/\overline{GT}_1$	I/O	Request/Grant. Pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT}_0$ having higher priority than $\overline{RQ}/\overline{GT}_1$. $\overline{RQ}/\overline{GT}$ has an internal pull-up resistor, so may be left unconnected. The request/grant sequence is as follows: 1. A pulse of one CLK wide from another local bus master indicates a local bus request ("hold") to the 8088 (pulse 1). 2. During a T_4 or T_1 clock cycle, a pulse one clock wide from the 8088 to the requesting master (pulse 2), indicates that the 8088 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge." The same rules as for HOLD/HLDA apply as for when the bus is released. 3. A pulse one CLK wide from the requesting master indicates to the 8088 (pulse 3) that the "hold" request is about to end and that the 8088 can reclaim the local bus at the next CLK. The CPU then enters T_4. Each master-master exchange of the local bus is a sequence of three pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active LOW. If the request is made while the CPU is performing a memory cycle, it will release the local bus during T_4 of the cycle when all the following conditions are met: 1. Request occurs on or before T_2. 2. Current cycle is not the low bit of a word. 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. If the local bus is Idle when the request is made, two possible events will follow: 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.
29	$\overline{LOCK}$	O	$\overline{LOCK}$. Indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active (LOW). The $\overline{LOCK}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW and floats to 3-state off in "hold acknowledge."

*Pin numbers correspond to DIPs only.

PIN DESCRIPTION (continued)

Pin No.*	Name	I/O	Description
24, 25	QS_1, QS_0	O	Queue Status. Provides status to allow external tracking of the internal 8088 instruction queue. The queue status is valid during the CLK cycle after which the queue operation is performed.
34	–	O	Pin 34 is always HIGH in the maximum mode.

QS_1	QS_0	Characteristics
0 (LOW)	0	No Operation
0	1	First Byte of Opcode from Queue
1 (HIGH)	0	Empty the Queue
1	1	Subsequent Byte from Queue

*Pin numbers correspond to DIPs only.

DETAILED DESCRIPTION

The 8088 Compared to the 8086

- The queue length is 4 bytes in the 8088; whereas, the 8086 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8 bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The 8088 BIU will fetch a new instruction to load into the queue each time there is a 1 byte hole (space available) in the queue. The 8086 waits until a 2-byte space is available.
- The internal execution time of the instruction set is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four clock cycles. The CPU is also limited by the speed of instruction fetches. This latter problem only occurs when a series of simple operations occurs. When the more sophisticated instructions of the 8088 are being used, the queue has time to fill and the execution proceeds as fast as the execution unit will allow.

The 8088 and 8086 are completely software compatible by virtue of their identical execution units. Software that is system dependent may not be completely transferable, but software that is not system dependent will operate equally as well on an 8088 or an 8086.

The hardware interface of the 8088 contains the major differences between the two CPUs. The pin assignments are nearly identical, however, with the following functional changes:

- $A_8 - A_{15}$ — These pins are only address outputs on the 8088. These address lines are latched internally and remain valid throughout a bus cycle in a manner similar to the 8085 upper address lines.
- $\overline{BHE}$ has no meaning on the 8088 and has been eliminated.
- $\overline{SSO}$ provides the $\overline{SO}$ status information in the minimum mode. This output occurs on pin 34 in minimum mode only. DT/$\overline{R}$, IO/$\overline{M}$, and $\overline{SSO}$ provide the complete bus status in minimum mode.
- IO/$\overline{M}$ has been inverted to be compatible with the MCS-85 bus structure.
- ALE is delayed by one clock cycle in the minimum mode when entering HALT, to allow the status to be latched with ALE.

I/O Addressing

In the 8088, I/O operations can address up to a maximum of 64K I/O registers. The I/O address appears in the same format as the memory address on bus lines $A_{15} - A_0$. The address lines $A_{19} - A_{16}$ are zero in I/O operations. The variable I/O instructions, which use register DX as a pointer, have full address capability, while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

Designers familiar with the 8085 or upgrading an 8085 design should note that the 8085 addresses I/O with an 8-bit address on both halves of the 16-bit address bus. The 8088 uses a full 16-bit address of its lower 16 address lines.

Bus Operation

The 8088 address/data bus is broken into three parts — the lower eight address/data bits ($AD_0 - AD_7$), the middle eight address bits ($A_8 - A_{15}$) and the upper four address bits ($A_{16} - A_{19}$). The address/data bits and the highest four address bits are time multiplexed. This technique provides the most efficient use of pins on the processor, permitting the use of a standard 40 lead package. The middle eight address bits are not multiplexed; i.e., they remain valid throughout each bus cycle. In addition, the bus can be demultiplexed at the processor with a single address latch if a standard, non-multiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T1, T2, T3 and T4. The address is emitted from the processor during T1 and data transfer occurs on the bus during T3 and T4. T2 is used primarily for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "wait" states (Tw) are inserted between T3 and T4. Each inserted "wait" state is of the same duration as a CLK cycle. Periods can occur between 8088 driven bus cycles. These are referred to as "idle" states (Ti) or inactive CLK cycles. The processor uses these cycles for internal housekeeping.

During T1 of any bus cycle, the ALE (address latch enable), signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/$\overline{MX}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$ are used by the bus controller, in maximum mode, to identify the type of bus transaction according to the following table:

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1 (HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S3 through S6 are multiplexed with high order address bits and are therefore valid during T2 through T4. S3 and S4 indicate which segment register was used for this bus cycle in forming the address according to the following table:

S_4	S_3	Characteristics
0 (LOW)	0	Alternate Data (extra segment)
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data

S5 is a reflection of the PSW interrupt enable bit. S6 is always equal to 0.

External Interface

Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 8088 RESET is required to be HIGH for greater than four clock cycles. The 8088 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 7 clock cycles. After this interval the 8088 operates normally, beginning with the instruction in absolute location FFFF0H (see Figure 3). The RESET input is internally synchronized to the processor clock. At initialization, the HIGH to LOW transition of RESET must occur no sooner than 50 μs after power up, to allow complete initialization of the 8088.

If INTR is asserted sooner than nine clock cycles after the end of RESET, the processor may execute one instruction before responding to the interrupt.

All three-state outputs float to three-state OFF during RESET. Status is active in the idle state for the first clock after RESET becomes active and then floats to three-state OFF.

Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are specified in the instruction set description in the iAPX 88 book or the iAPX 86, 88 User's Manual. Hardware interrupts can be classified as nonmaskable or maskable.

Interrupts result in a transfer of control to a new program location. A 256 element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see Figure 3), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type." An interrupting device supplies an 8-bit type number, during the interrupt acknowledge sequence, which is used to vector through the appropriate element to the new interrupt service program location.

Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt (NMI) pin which has higher priority than the maskable interrupt request (INTR) pin. A typical use would be to activate a power failure routine. The NMI is edge-triggered on a LOW to HIGH transition. The activation of this pin causes a type 2 interrupt.

NMI is required to have a duration in the HIGH state of greater than two clock cycles but is not required to be synchronized to the clock. Any higher going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves (2 bytes in the case of word moves) of a block type instruction. Worst case response to NMI would be for multiply, divide and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

Maskable Interrupt (INTR)

The 8088 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable (IF) flag bit. The interrupt request signal is level triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block type instruction. During interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single step), although the FLAGS register which is automatically pushed onto the stack reflects the state of the processor prior to the interrupt. Until the old FLAGS register is restored, the enable bit will be zero unless specifically set by an instruction.

During the response sequence (see Figure 1), the processor executes two successive (back to back) interrupt acknowledge cycles. The 8088 emits the LOCK signal (maximum mode only) from T2 of the first bus cycle until T2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interrupt system (e.g., 8259A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The interrupt return instruction includes a flags pop which returns the status of the original interrupt enable bit when it restores the flags.

HALT

When a software HALT instruction is executed, the processor indicates that it is entering the HALT state in one of two ways, depending upon which mode is strapped. In minimum mode, the processor issues ALE, delayed by one clock cycle, to allow the system to latch the halt status. Halt status is available on IO/$\overline{M}$, DT/$\overline{R}$ and $\overline{SSO}$. In maximum mode, the processor issues appropriate HALT status on $\overline{S2}$, $\overline{S1}$ and $\overline{S0}$, and the 8288 bus controller issues one ALE. The 8088 will not leave the HALT state when a local bus hold is entered while in HALT. In this case, the processor reissues the HALT indicator at the end of the local bus hold. An interrupt request or RESET will force the 8088 out of the HALT state.

Read/Modify/Write (Semaphore) Operations via LOCK

The LOCK status information is provided by the processor when consecutive bus cycles are required during the execution of an instruction. This allows the processor to perform read/modify/write operations on memory (via the "exchange register with memory" instruction), without another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The $\overline{LOCK}$ signal is activated (LOW) in the clock cycle following decoding of the LOCK prefix instruction. It is deactivated at the end of the last bus cycle of the instruction following the LOCK prefix. While $\overline{LOCK}$ is active, a request on a $\overline{RQ}/\overline{GT}$ pin will be recorded, and then honored at the end of the LOCK.

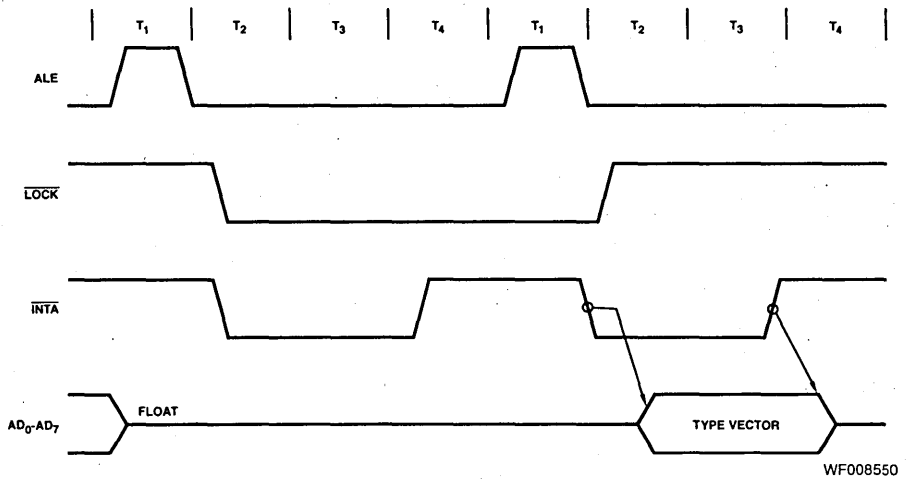


Figure 1. Interrupt Acknowledge Sequence

External Synchronization via TEST

As an alternative to interrupts, the 8088 provides a single software-testable input pin (TEST). This input is utilized by executing a WAIT instruction. The single WAIT instruction is repeatedly executed until the TEST input goes active (LOW). The execution of WAIT does not consume bus cycles once the queue is full.

If a local bus request occurs during WAIT execution, the 8088 three-states all output drivers. If interrupts are enabled, the 8088 will recognize interrupts and process them. The WAIT instruction is then refetched, and reexecuted.

Basic System Timing

In minimum mode, the MN/MX pin is strapped to V_{CC} and the processor emits bus control signals compatible with the 8085 bus structure. In maximum mode, the MN/MX pin is strapped to GND, and the processor emits coded status information, which the 8288 bus controller uses to generate MULTIBUS compatible bus control signals.

System Timing — Minimum System

The read cycle begins in T1 with the assertion of the address latch enable (ALE) signal. The trailing (low going) edge of this signal is used to latch the address information, which is valid on the address/data bus (AD0 – AD7) at this time, into the 8282/8283 latch. Address lines A8 through A15 do not need to be latched because they remain valid throughout the bus cycle. From T1 to T4 the IO/M signal indicates a memory or I/O operation. At T2 the address is removed from the address/data bus, and the bus goes to a high impedance state. The read control signal is also asserted at T2. The read (RD) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later, valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again three-state its bus drivers. If a transceiver (8286/8287) is required to buffer the 8088 local bus, signals DT/R and DEN are provided by the 8088.

A write cycle also begins with the assertion of ALE and the emission of the address. The IO/M signal is again asserted to indicate a memory or I/O write operation. In T2, immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until at least the middle of T4. During T2, T3 and T_W, the processor asserts the write control signal. The write (WR) signal becomes active at the beginning of T2, as opposed to the read, which is delayed somewhat into T2 to provide time for the bus to float.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge (INTA) signal is asserted in place of the read (RD) signal and the address bus is floated (see Figure 1). In the second of two successive INTA cycles, a byte of information is read from the data bus, as supplied by the interrupt system logic (i.e., 8259A priority interrupt controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into the interrupt vector lookup table, as described earlier.

Bus Timing — Medium Complexity Systems

For medium complexity systems, the MN/MX pin is connected to GND and the 8288 bus controller is added to the system, as well as an 8282/8283 latch for latching the system address, and an 8286/8287 transceiver to allow for bus loading greater than the 8088 is capable of handling. Signals ALE, DEN and DT/R are generated by the 8288 instead of the processor in this configuration, although their timing remains relatively the same. The 8088 status outputs (S2, S1 and S0) provide type of cycle information and become 8288 inputs. This bus cycle information specifies read (code, data or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 8288 thus issues control signals specifying memory read or write, I/O read or write or interrupt acknowledge. The 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data is not valid at the leading edge of write. The 8286/8287 transceiver receives

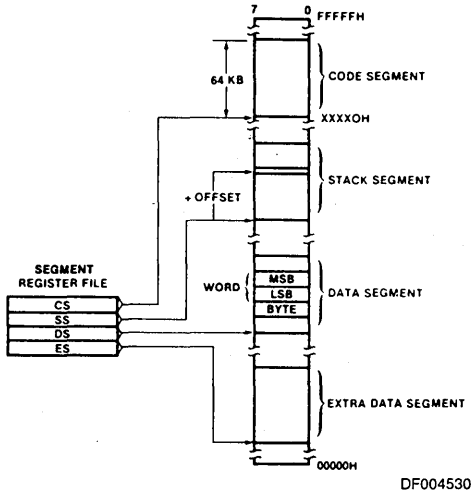
the usual T and $\overline{OE}$ inputs from the 8288's DT/$\overline{R}$ and $\overline{DEN}$ outputs.

The pointer into the interrupt vector table, which is passed during the second $\overline{INTA}$ cycle, can derive from an 8259A located on either the local bus or the system bus. If the master 8289A priority interrupt controller is positioned on the local bus, a TTL gate is required to disable the 8286/8287 transceiver when reading from the master 8259A during the interrupt acknowledge sequence and software "poll."

Memory Organization

The processor provides a 20-bit address to memory which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra data and stack segments of up to 64K bytes each, with each segment falling on 16-byte boundaries (see Figure 2).

All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to the rules of the following table. All information in one segment type share the same logical attributes (e.g., code or data). By structuring memory into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster and more structured.



DF004530

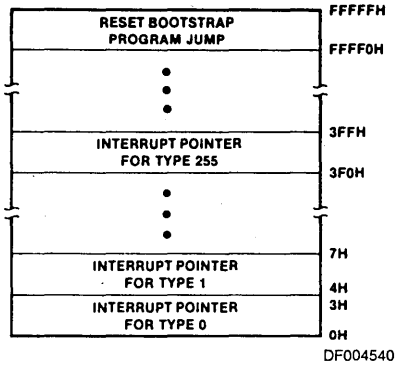
Figure 2. Memory Organization

Word (16-bit) operands can be located on even or odd address boundaries. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in the next higher address location. The BIU will automatically execute two fetch or write cycles for 16-bit operands.

Certain locations in memory are reserved for specific CPU operations (see Figure 3). Locations from addresses FFFF0H through FFFFFH are reserved for operations including a jump to the initial system initialization routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be located. Locations 00000H through 003FFH are reserved for interrupt operations. Four-byte pointers consisting of a 16-bit segment address and a 16-bit offset address direct program flow to one of the 256 possible interrupt service routines. The pointer elements are assumed to have been stored at their respective places in reserved memory prior to the occurrence of interrupts.

Minimum and Maximum Modes

The requirements for supporting minimum and maximum 8088 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the 8088 is equipped with a strap pin (MN/$\overline{MX}$) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/$\overline{MX}$ pin is strapped to GND, the 8088 defines pins 24 through 31 and 34 in maximum mode. When the MN/$\overline{MX}$ pin is strapped to V_{CC}, the 8088 generates bus control signals itself on pins 24 through 31 and 34.



DF004540

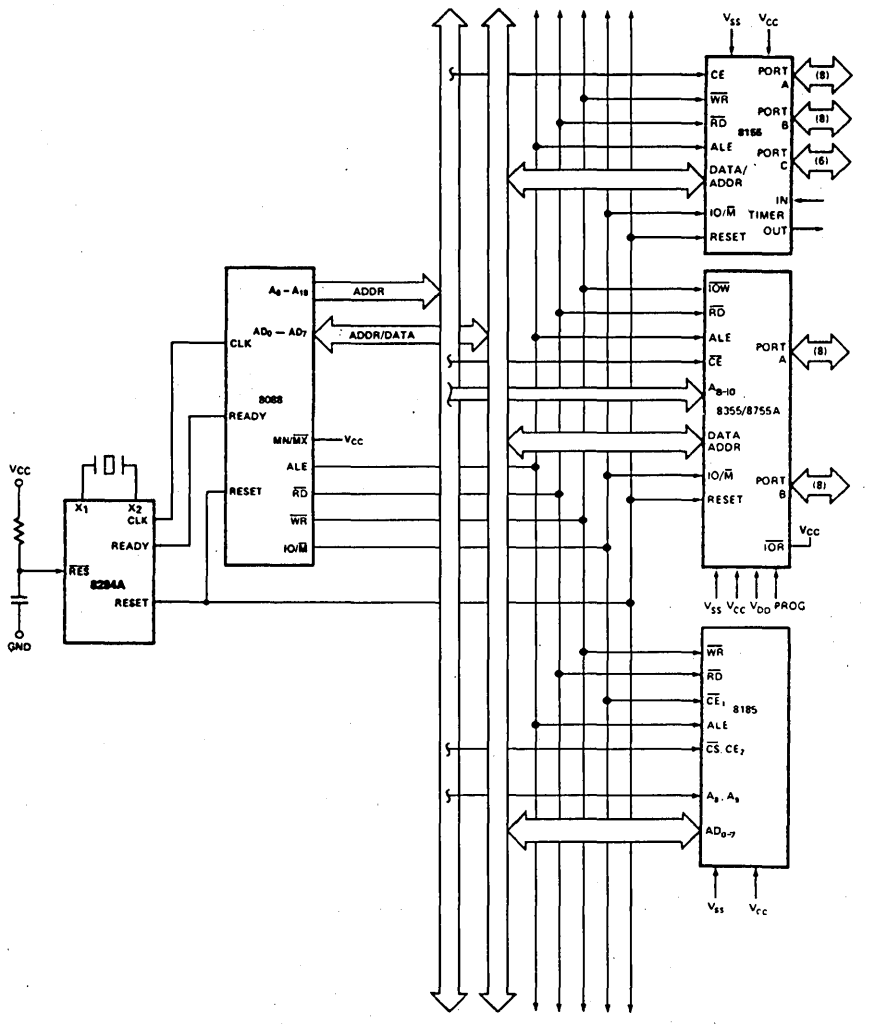
Figure 3. Reserved Memory Locations

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch.
Stack	STACK (SS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when: relative to stack, destination of string operation, or explicitly overridden.
External (Global) Data	EXTRA (ES)	Destination of string operations: Explicitly selected using a segment override.

The minimum mode 8088 can be used with either a multiplexed or demultiplexed bus. The multiplexed bus configuration is compatible with the MCS-85™ multiplexed bus peripherals (8155, 8156, 8355, 8755A, and 8185). This configuration (see Figure 4) provides the user with a minimum chip count system. This architecture provides the 8088 processing power in a highly integrated form.

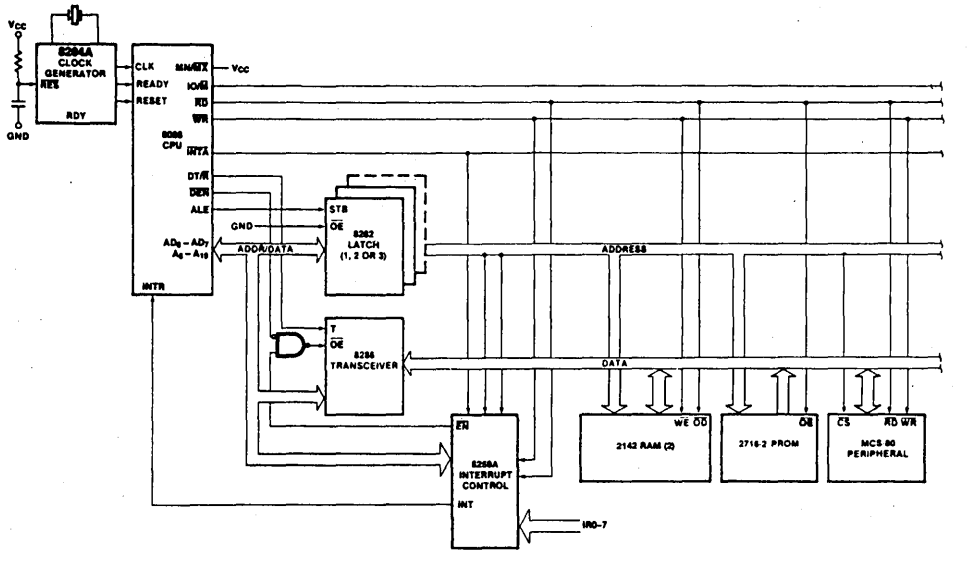
The demultiplexed mode requires one latch (for 64K addressability) or two latches (for a full megabyte of addressing). A third latch can be used for buffering if the address bus loading requires it. An 8286 or 8287 transceiver can also be used if data bus buffering is required (see Figure 5). The 8088 provides $\overline{DEN}$ and DT/$\overline{R}$ to control the transceiver, and ALE to latch the addresses. This configuration of the minimum mode provides the standard demultiplexed bus structure with heavy bus buffering and relaxed bus timing requirements.

The maximum mode employs the 8288 bus controller (see Figure 6). The 8288 decodes status lines $\overline{S0}$, $\overline{S1}$ and $\overline{S2}$ and provides the system with all bus control signals. Moving the bus control to the 8288 provides better source and sink current capability to the control lines and frees the 8088 pins for extended large system features. Hardware lock, queue status and two request/grant interfaces are provided by the 8088 in maximum mode. These features allow co-processors in local bus and remote bus configurations.



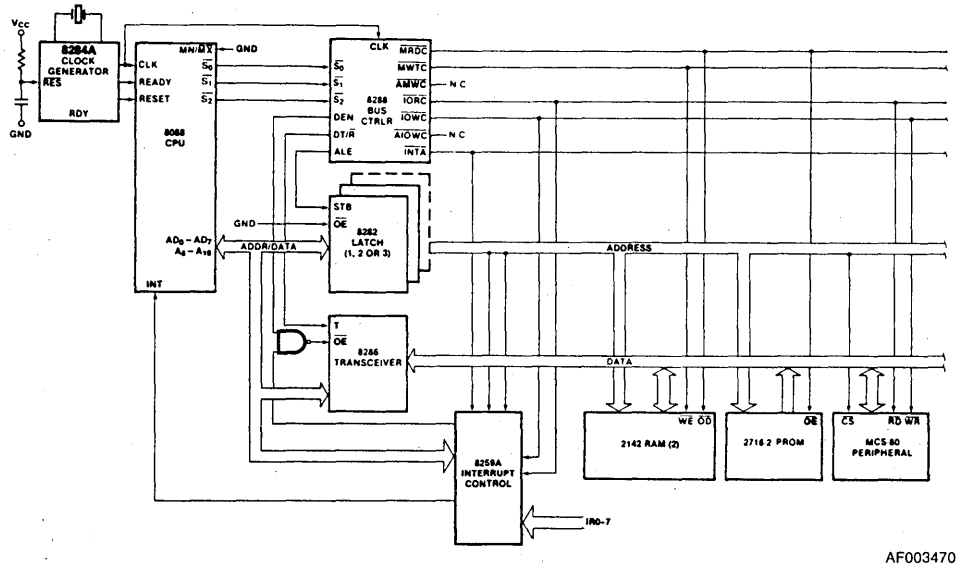
AF003451

Figure 4. Multiplexed Bus Configuration



AF003461

Figure 5. Demultiplexed Bus Configuration



AF003470

Figure 6. Fully Buffered System Using Bus Controller

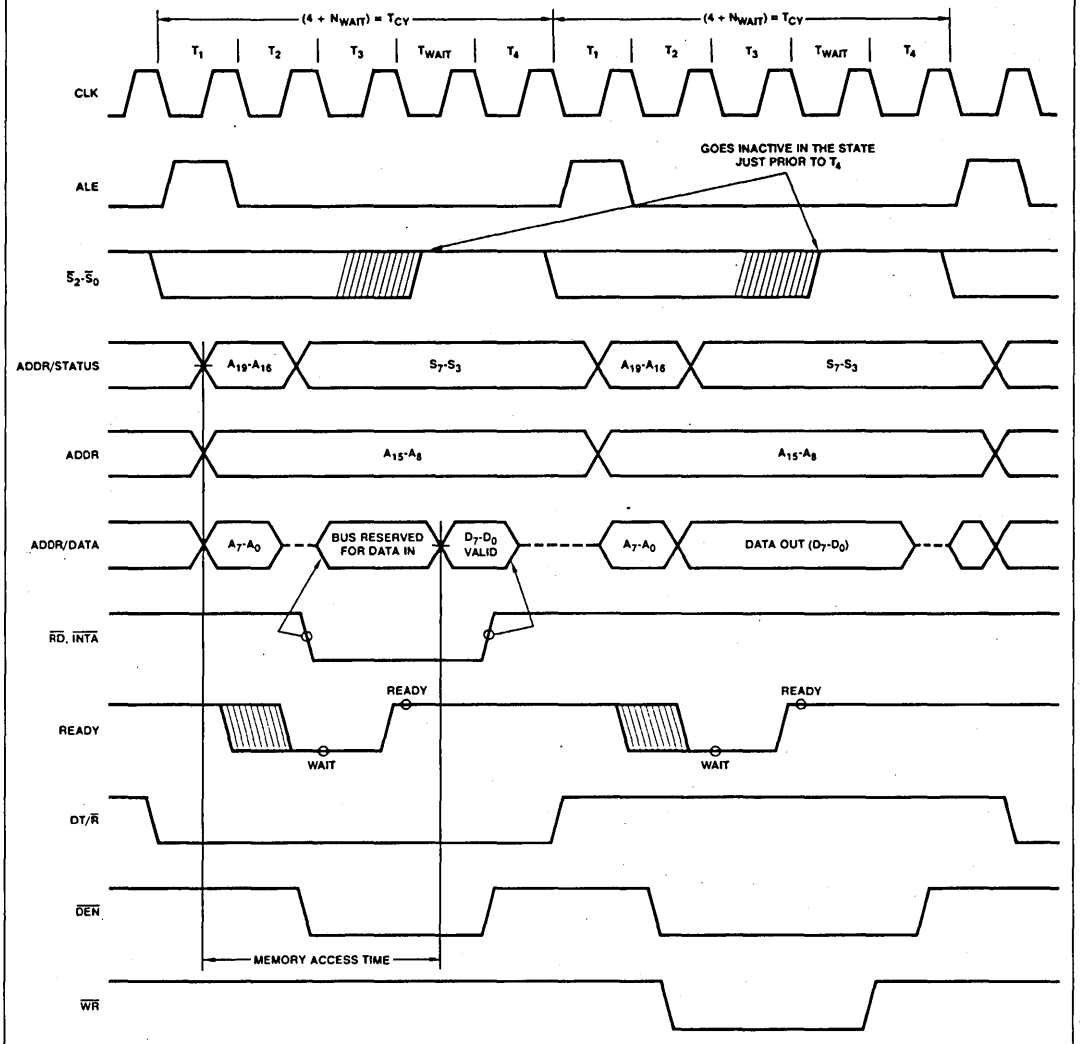


Figure 7. Basic System Timing

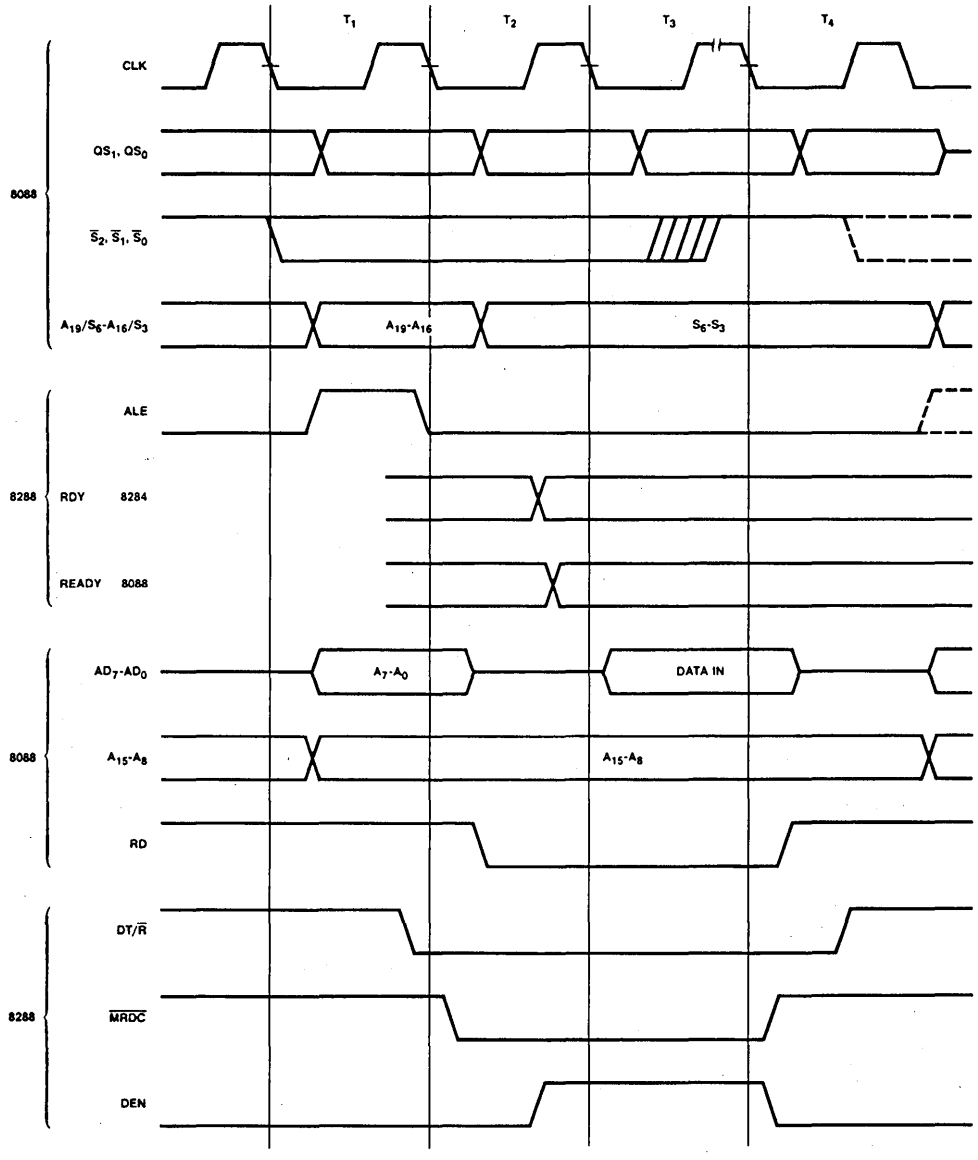


Figure 8. Medium Complexity System Timing

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on any Pin
with Respect to Ground ..... −1.0 to +7.0 V
Power Dissipation ..... 2.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC})
8088 ..... 5 V ± 10%
8088-1, 8088-2 ..... 5 V ± 5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC})
8088 ..... 5 V ± 10%
8088-1, 8088-2 ..... 5 V ± 5%

Military (M) Devices
Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... 5 V ± 10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (for APL, Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions		Min	Max	Units
V_{IL}†	Input Low Voltage	COML: see Note 1		−0.5*	+0.8	V
		MIL: V_{CC} = Min. & Max.				
V_{IH}†	Input High Voltage	COML: see Notes 1 & 2		2.0	V_{CC} +0.5*	V
		MIL: V_{CC} = Min. & Max.				
V_{OL}	Output Low Voltage	COML: I_{OL} = 2.0 mA			0.45	V
		MIL: I_{OL} = 2.0 mA, V_{CC} = Min.				
V_{OH}	Output High Voltage	COML: I_{OH} = −400 µA		2.4		V
		MIL: I_{OH} = −400 µA, V_{CC} = Min.				
I_{CC}	Power Supply Current (Note 6)	MIL: T_C = 25°C, V_{CC} = Max.			340	mA
I_{LI}	Input Leakage Current	COML: 0 V ≤ V_{IN} ≤ V_{CC}			±10	µA
		MIL: V_{CC} = Max., V_{IN} = 5.5 V & 0 V		−10	10	
I_{LO}††	Output Leakage Current	COML: 0.45 V ≤ V_{OUT} ≤ V_{CC}			COML ±10	µA
		MIL: V_{CC} = Max., V_{OUT} = 5.5 V & 0.45 V		MIL −10	MIL 10	
V_{CL}	Clock Input Low Voltage			−0.5	+0.6	V
V_{CH}	Clock Input High Voltage			3.9	V_{CC} +1.0	V
C_{IN}	Capacitance of Input Buffer (All input except AD_0–AD_7, RQ/GT)	fc = 1 MHz			15	pF
C_{IO}	Capacitance of I/O Buffer (AD_0–AD_7, RQ/GT)	fc = 1 MHz			15	pF
I_{CC}	Power Supply Current	T_A = 25°C	8088		340	mA
			8088-1, -2		350	
			P8088		250	

Notes: 1. V_{IL} tested with MN/$\overline{MX}$ pin = 0 V; V_{IH} tested with MN/$\overline{MX}$ pin = 5 V; MN/$\overline{MX}$ is a strap pin.
2. Not applicable to $\overline{RQ}$/$\overline{GT0}$ and $\overline{RQ}$/$\overline{GT1}$ pins (pins 30 and 31).
3. Signal at 8284 or 8288 shown for reference only.
4. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
5. Applies only to T_3 and Wait states.
6. I_{CC} is measured while running a functional pattern with spec value I_{OL}/I_{OH} loads applied.

* Guaranteed by design; not tested.
† Group A, Subgroups 7 and 8 only are tested.
†† Group A, Subgroups 1 and 2 only are tested.
††† Not included in Group A test.

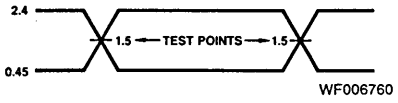
SWITCHING CHARACTERISTICS over COMMERCIAL operating range
MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Parameter Symbol	Parameter Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5 V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0 V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284 (See Notes 3, 4)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284 (See Notes 3, 4)		0		0		0		ns
TRYHCH	READY Set-up Time into 8088		118		68		53		ns
TCHRYX	READY Hold Time into 8088		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 5)		-8		-8		-10		ns
THVCH	HOLD Set-up Time		35		20		20		ns
TINVCH	INTR, NMI, $\overline{\text{TEST}}$ Set-up Time (See Note 4)		30		15		15		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0 V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8 V		12		12		12	ns

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)
TIMING RESPONSES

Parameter Symbol	Parameter Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLAV	Address Valid Delay	C_L = 20-100 pF for all 8088 Outputs (in addition to internal loads)	10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TLHLL	ALE Width		TCLCH -20		TCLCH -10		TCLCH -10		ns
TCLLH	ALE Active Delay			80		50		40	ns
TCHLL	ALE Inactive Delay			85		55		45	ns
TLLAX	Address Hold Time to ALE Inactive		TCHCL -10		TCHCL -10		TCHCL -10		ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TWHDX	Data Hold Time After $\overline{\text{WR}}$		TCLCH -30		TCLCH -30		TCLCH -25		ns
TCVCTV	Control Active Delay 1		10	110	10	70	10	50	ns
TCHCTV	Control Active Delay 2		10	110	10	60	10	45	ns
TCVCTX	Control Inactive Delay		10	110	10	70	10	50	ns
TAZRL	Address Float to READ Active		0		0		0		ns
TCLRL	$\overline{\text{RD}}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{\text{RD}}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{\text{RD}}$ Inactive to Next Address Active		TCLCL -45		TCLCL -40		TCLCL -35		ns
TCLHAV	HLDA Valid Delay		10	160	10	100	10	60	ns
TRLRH	$\overline{\text{RD}}$ Width		2TCLCL -75		2TCLCL -50		2TCLCL -40		ns
TWLWH	$\overline{\text{WR}}$ Width		2TCLCL -60		2TCLCL -40		2TCLCL -35		ns
TAVAL	Address Valid to ALE Low		TCLCH -60		TCLCH -40		TCLCH -35		ns
TOLOH	Output Rise Time	From 0.8 to 2.0 V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8 V		12		12		12	ns

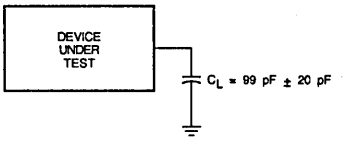
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006760

AC testing inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0." The clock is driven at 4.3 V and 0.25 V. Timing measurements are made at 1.5 V for both a logic "1" and "0."

SWITCHING TEST LOAD CIRCUIT



WF006771

C_L Includes JIG Capacitance.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

MAX MODE SYSTEM (USING 8288 BUS CONTROLLER) TIMING REQUIREMENTS

Parameter Symbol	Parameter Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5 V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0 V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284 (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284 (See Notes 1,2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8088		118		68		53		ns
TCHRYX	READY Hold Time into 8088		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 3)		-8		-8		-10		ns
TINVCH	Set-up Time for Recognition (INTR, NMI, TEST) (See Note 2)		30		15		15		ns
TGVCH	RQ/GT Set-up Time		30		15		12		ns
TCHGX	RQ Hold Time into 8086		40		30		20		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0 V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8 V		12		12		12	ns

Notes: 1. Signal at 8284 or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T_3 and Wait states.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

TIMING RESPONSES

Parameter Symbol	Parameter Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLML	Command Active Delay (See Note 1)	C_L = 20-100 pF for all 8088 outputs (in addition to internal loads)	10	35	10	35	10	35	ns
TCLMH	Command Inactive Delay (See Note 1)		10	35	10	35	10	35	ns
TRYHSH	READY Active to Status Passive (See Note 3)			110		65		45	ns
TCHSV	Status Active Delay		10	110	10	60	10	45	ns
TCLSH	Status Inactive Delay		10	130	10	70	10	55	ns
TCLAV	Address Valid Delay		10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TSVLH	Status Valid to ALE High (See Note 1)			15		15		15	ns
TSVMCH	Status Valid to MCE High (See Note 1)			15		15		15	ns
TCLLH	CLK Low to ALE Valid (See Note 1)			15		15		15	ns
TCLMCH	CLK Low to MCE High (See Note 1)			15		15		15	ns
TCHLL	ALE Inactive Delay (See Note 1)			15		15		15	ns
TCLMCL	MCE Inactive Delay (See Note 1)			15		15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TCVNV	Control Active Delay (See Note 1)		5	45	5	45	5	45	ns
TCVNX	Control Inactive Delay (See Note 1)		10	45	10	45	10	45	ns
TAZRL	Address Float to Read Active		0		0		0		ns
TCLRL	RD Active Delay		10	165	10	100	10	70	ns
TCLRH	RD Inactive Delay		10	150	10	80	10	60	ns
TRHAV	RD Inactive to Next Address Active		TCLCL -45		TCLCL -40		TCLCL -35		ns
TCHDTL	Direction Control Active Delay (See Note 1)			50		50		50	ns
TCHDTH	Direction Control Inactive Delay (See Note 1)			30		30		30	ns
TCLGL	GT Active Delay			85		50	0	45	ns
TCLGH	GT Inactive Delay			85		50	0	45	ns
TRLRH	RD Width		2TCLCL -75		2TCLCL -50		2TCLCL -40		ns
TOLOH	Output Rise Time	From 0.8 to 2.0 V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8 V		12		12		12	ns

Notes: 1. Signal at 8284 or 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T_2 state (8 ns into T_3 state).

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	8088		8088-2		Unit
			Min.	Max.	Min.	Max.	
TCLCL	CLK Cycle Period (Note 11)		200	500	125	500	ns
TCLCH	CLK LOW Time		118		68		ns
TCHCL	CLK HIGH Time		69		44		ns
TCH1CH2	CLK Rise Time (Note 5)	From 1.0 to 3.5 V		10		10	ns
TCL2CL1	CLK Fall Time (Note 5)	From 3.5 to 1.0 V		10		10	ns
TDVCL	Data in Setup Time		30		20		ns
TCLDX	Data in Hold Time		10		10		ns
TR1VCL	RDY Setup Time into 8284A (Notes 1 & 2)		35		35		ns
TCLR1X	RDY Hold Time into 8284A (Notes 1 & 2)		0		0		ns
TRYHCH	READY Setup Time into 8088		118		68		ns
TCHRYX	READY Hold Time into 8088		30		20		ns
TRYLCL	READY Inactive to CLK (Note 3)		-8		-8		ns
THVCH	HOLD Setup Time		35		20		ns
TINVCH	INTR, NMI, $\overline{\text{TEST}}$ Setup Time (Note 2)		30		15		ns
TILIH	Input Rise Time (Except CLK) (Note 5)	From 0.8 to 2.0 V		20		20	ns
TIHIL	Input Fall Time (Except CLK) (Note 5)	From 2.0 to 0.8 V		12		12	ns

Notes:
1. Signal at 8284A and 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8 ns into T3).
5. Not tested; these specs are controlled by the Teradyne J941 tester.
6. V_{CC} = 4.5 V, 5.5 V; V_{IH} = 2.4 V; V_{IL} = .45 V; V_{IHC} = 4.3 V; V_{ILC} = .25 V; V_{OH} = 1.6 V; V_{OL} = 1.4 V
7. Minimum spec tested at V_{CC} Max. (5.5 V) only.
8. Maximum spec tested at V_{CC} Min. (4.5 V) only.
9. Tested at V_{CC} Max. (5.5 V) only.
10. Tested at V_{CC} Min. (4.5 V) only.
11. Test conditions for TCLCL Max. are: V_{CC} = 4.5 V; V_{OL} = 1 V; V_{IL} = 0 V; V_{IH} = 4 V; V_{ILC} = 0 V; V_{IHC} = 5 V

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)
TIMING RESPONSES

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	8088		8088-2		Unit
			Min.	Max.	Min.	Max.	
TCLAV	Address Valid Delay	C_L = 100 pF for all 8088 Outputs (in addition to internal loads).	10	110	10	60	ns
TCLAX	Address Hold Time (Notes 7 & 8)		10		10		ns
TCLAZ	Address Float Delay (Note 8)		10	80	10	50	ns
TLHLL	ALE Width (Note 10)		98		58		ns
TCLLH	ALE Active Delay (Note 8)			80		50	ns
TCHLL	ALE Inactive Delay (Note 8)			85		55	ns
TLLAX	Address Hold Time to ALE Inactive (Note 7)		59		34		ns
TCLDV	Data Valid Delay (Note 8)		10	110	10	60	ns
TCHDX	Data Hold Time (Note 10)		10		10		ns
TWHDX	Data Hold Time After WR (Note 9)		88		38		ns
TCVCTV	Control Active Delay 1 (Note 8)		10	110	10	70	ns
TCHCTV	Control Active Delay 2 (Note 8)		10	110	10	60	ns
TCVCTX	Control Inactive Delay (Note 8)		10	110	10	70	ns
TAZRL	Address Float to READ Active (Note 9)		0		0		ns
TCLRL	$\overline{RD}$ Active Delay (Note 8)		10	165	10	100	ns
TCLRH	$\overline{RD}$ Inactive Delay (Note 8)		10	150	10	80	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active (Note 10)		155		85		ns
TCLHAV	HLDA Valid Delay (Note 8)		10	160	10	100	ns
TRLRH	$\overline{RD}$ Width (Note 10)		325		200		ns
TWLWH	$\overline{WR}$ Width (Note 10)		340		210		ns
TAVAL	Address Valid to ALE Low (Note 9)		58		28		ns
TOLOH	Output Rise Time (Note 9)	From 0.8 to 2.0 V		20		20	ns
TOHOL	Output Fall Time (Note 9)	From 2.0 to 0.8 V		12		12	ns

Notes:
1. Signal at 8284A and 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8 ns into T3).
5. Not tested; these specs are controlled by the Teradyne J941 tester.
6. V_{CC} = 4.5 V, 5.5 V; V_{IH} = 2.4 V; V_{IL} = .45 V; V_{IHC} = 4.3 V; V_{ILC} = .25 V; V_{OH} = 1.6 V; V_{OL} = 1.4 V
7. Minimum spec tested at V_{CC} Max. (5.5 V) only.
8. Maximum spec tested at V_{CC} Min. (4.5 V) only.
9. Tested at V_{CC} Max. (5.5 V) only.
10. Tested at V_{CC} Min. (4.5 V) only.
11. Test conditions for TCLCL Max. are: V_{CC} = 4.5 V; V_{OL} = 1 V; V_{IL} = 0 V; V_{IH} = 4 V; V_{ILC} = 0 V; V_{IHC} = 5 V

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)
MAX MODE SYSTEM (USING 8288 BUS CONTROLLER) TIMING REQUIREMENTS

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	8088		8088-2		Unit
			Min.	Max.	Min.	Max.	
TCLCL	CLK Cycle Period (Note 11)		200	500	125	500	ns
TCLCH	CLK LOW Time		118		68		ns
TCHCL	CLK HIGH Time		69		44		ns
TCH1CH2	CLK Rise Time (Note 5)	From 1.0 to 3.5 V		10		10	ns
TCL2CL1	CLK Fall Time (Note 5)	From 3.5 to 1.0 V		10		10	ns
TDVCL	Data in Setup Time		30		20		ns
TCLDX	Data in Hold Time		10		10		ns
TR1VCL	RDY Setup Time into 8284A (Notes 1 & 2)		35		35		ns
TCLR1X	RDY Hold Time into 8284A (Notes 1 & 2)		0		0		ns
TRYHCH	READY Setup Time into 8088		118		68		ns
TCHRYX	READY Hold Time into 8088		30		20		ns
TRYLCL	READY Inactive to CLK (Note 3)		-8		-8		ns
TINVCH	Setup Time for Recognition (INTR, NMI, $\overline{TEST}$ (Note 2)		30		15		ns
TGVCH	$\overline{RQ}/\overline{GT}$ Setup Time		30		15		ns
TCHGX	$\overline{RQ}$ Hold Time into 8086		40		30		ns
TILIH	Input Rise Time (Except CLK) (Note 5)	From 0.8 to 2.0 V		20		20	ns
TIHIL	Input Fall Time (Except CLK) (Note 5)	From 2.0 to 0.8 V		12		12	ns

Notes:
1. Signal at 8284A and 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8 ns into T3).
5. Not tested; these specs are controlled by the Teradyne J941 tester.
6. V_{CC} = 4.5 V, 5.5 V V_{IH} = 2.4 V
   V_{IL} = .45 V V_{IHC} = 4.3 V
   V_{ILC} = .25 V V_{OH} = 1.6 V
   V_{OL} = 1.4 V
7. Minimum spec tested at V_{CC} Max. (5.5 V) only.
8. Maximum spec tested at V_{CC} Min. (4.5 V) only.
9. Tested at V_{CC} Max. (5.5 V) only.
10. Tested at V_{CC} Min. (4.5 V) only.
11. Test conditions for TCLCL Max. are:
    V_{CC} = 4.5 V V_{OL} = 1 V
    V_{IL} = 0 V V_{IH} = 4 V
    V_{ILC} = 0 V V_{IHC} = 5 V

SWITCHING CHARACTERISTICS over **MILITARY** operating range (continued)

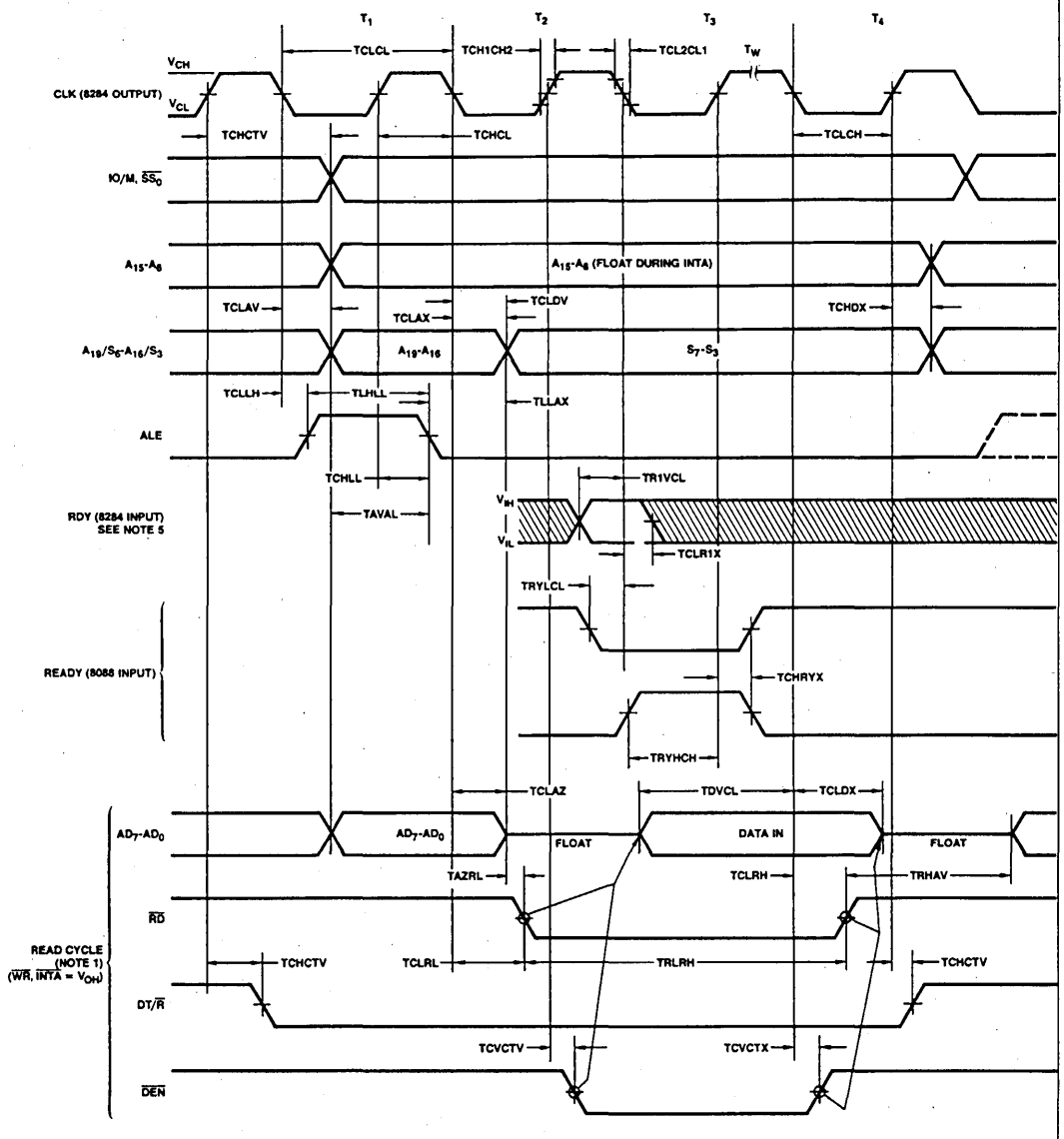
TIMING RESPONSES

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	8088		8088-2		Unit
			Min.	Max.	Min.	Max.	
TCLML	Command Active Delay (Note 1)	C_L = 100 pF for all 8088 Outputs (In addition to internal loads)	10	35	10	35	ns
TCLMH	Command Inactive Delay (Note 1)		10	35	10	35	ns
TRYHSH	READY Active to Status Passive (Note 4)			110		65	ns
TCHSV	Status Active Delay (Notes 7 & 8)		10	110	10	60	ns
TCLSH	Status Inactive Delay		10	130	10	70	ns
TCLAV	Address Valid Delay		10	110	10	60	ns
TCLAX	Address Hold Time		10		10		ns
TCLAZ	Address Float Delay		10	80	10	50	ns
TSVLH	Status Valid to ALE HIGH (Note 1)			15		15	ns
TSVMCH	Status Valid to MCE HIGH (Note 1)			15		15	ns
TCLLH	CLK LOW to ALE Valid (Note 1)			15		15	ns
TCLMCH	CLK LOW to MCE HIGH (Note 1)			15		15	ns
TCHLL	ALE Inactive Delay (Note 1)			15		15	ns
TCLMCL	MCE Inactive Delay (Note 1)			15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	ns
TCHDX	Data Hold Time		10		10		ns
TCVNV	Control Active Delay (Note 1)		5	45	5	45	ns
TCVNX	Control Inactive Delay (Note 1)		10	45	10	45	ns
TAZRL	Address Float to Read Active		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		155		85		ns
TCHDTL	Direction Control Active Delay (Note 1)			50		50	ns
TCHDTH	Direction Control Inactive Delay (Note 1)			30		30	ns
TCLGL	GT Active Delay (Note 8)			110		50	ns
TCLGH	GT Inactive Delay (Note 8)			85		50	ns
TRLRH	$\overline{RD}$ Width		325		200		ns
TOLOH	Output Rise Time	From 0.8 to 2.0 V		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8 V		12		12	ns

Notes:
1. Signal at 8284A and 8288 shown for reference only.
2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8 ns into T3).
5. Not tested; these specs are controlled by the Teradyne J941 tester.
6. V_{CC} = 4.5 V, 5.5 V V_{IH} = 2.4 V
   V_{IL} = .45 V V_{IHC} = 4.3 V
   V_{ILC} = .25 V V_{OH} = 1.6 V
   V_{OL} = 1.4 V
7. Minimum spec tested at V_{CC} Max. (5.5 V) only.
8. Maximum spec tested at V_{CC} Min. (4.5 V) only.
9. Tested at V_{CC} Max. (5.5 V) only.
10. Tested at V_{CC} Min. (4.5 V) only.
11. Test conditions for TCLCL Max. are:
    V_{CC} = 4.5 V V_{OL} = 1 V
    V_{IL} = 0 V V_{IH} = 4 V
    V_{ILC} = 0 V V_{IHC} = 5 V

SWITCHING WAVEFORMS

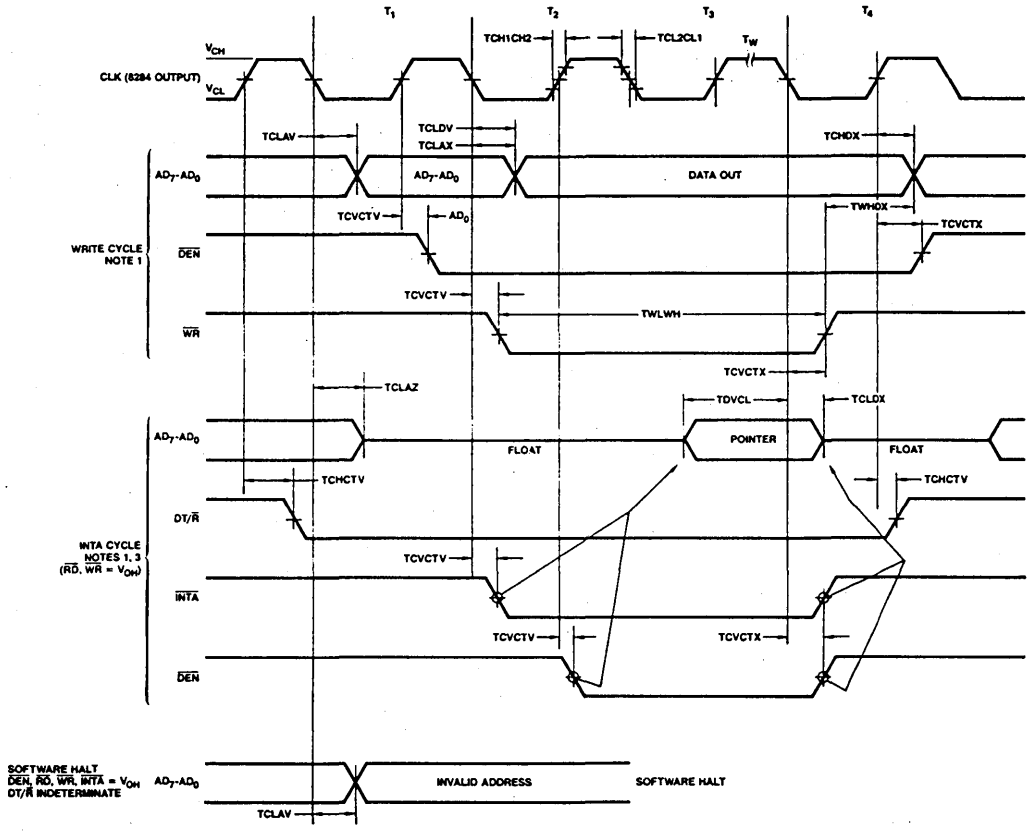
BUS TIMING – MINIMUM MODE SYSTEM



WF006791

SWITCHING WAVEFORMS

BUS TIMING – MINIMUM MODE SYSTEM (continued)

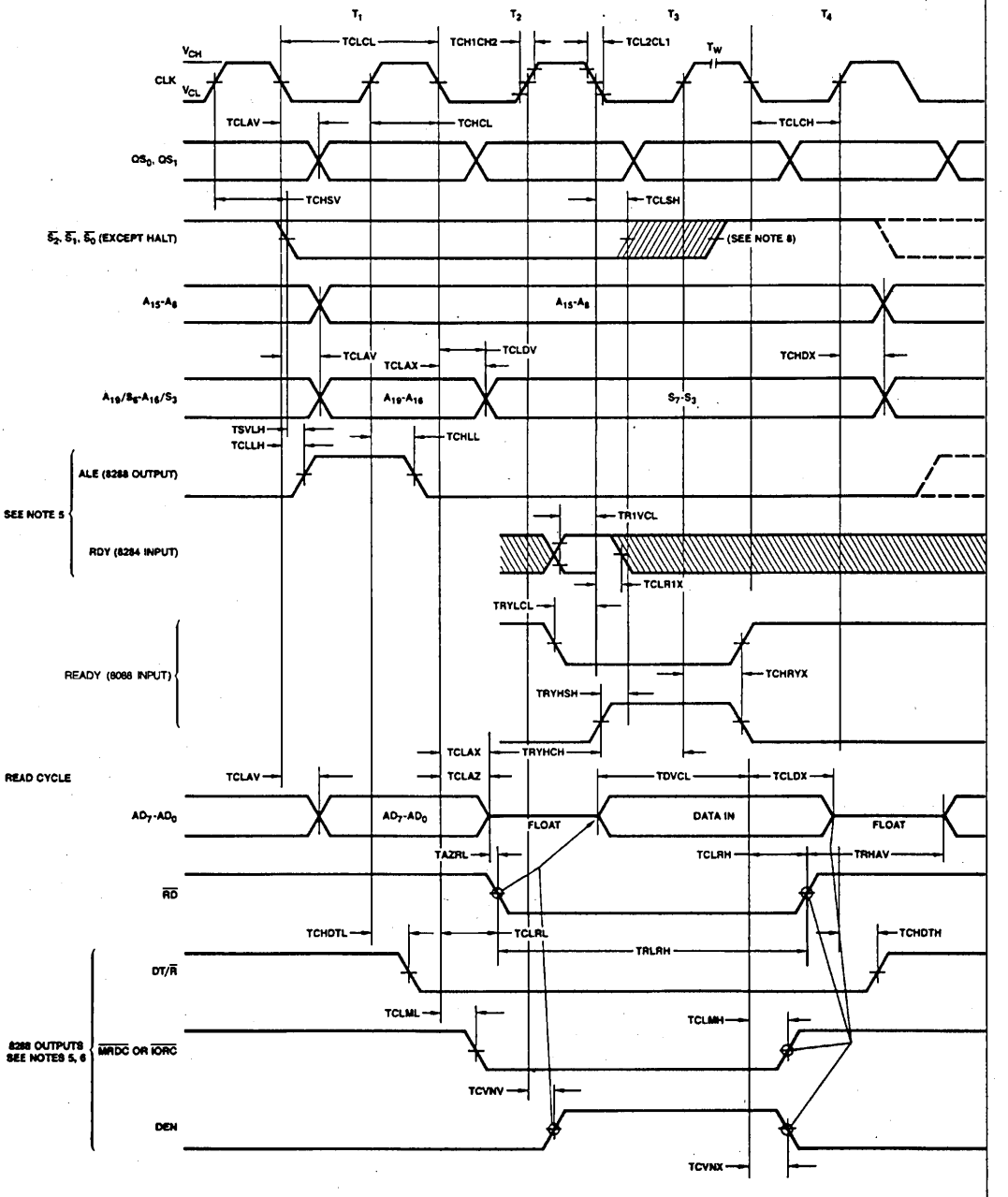


WF006780

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Two INTA cycles run back-to-back. The 8088 local ADDR/DATA bus is floating during both INTA cycles. Control signals are shown for the second INTA cycle.
4. Signals at 8284 are shown for reference only.
5. All timing measurements are made at 1.5 V unless otherwise noted.

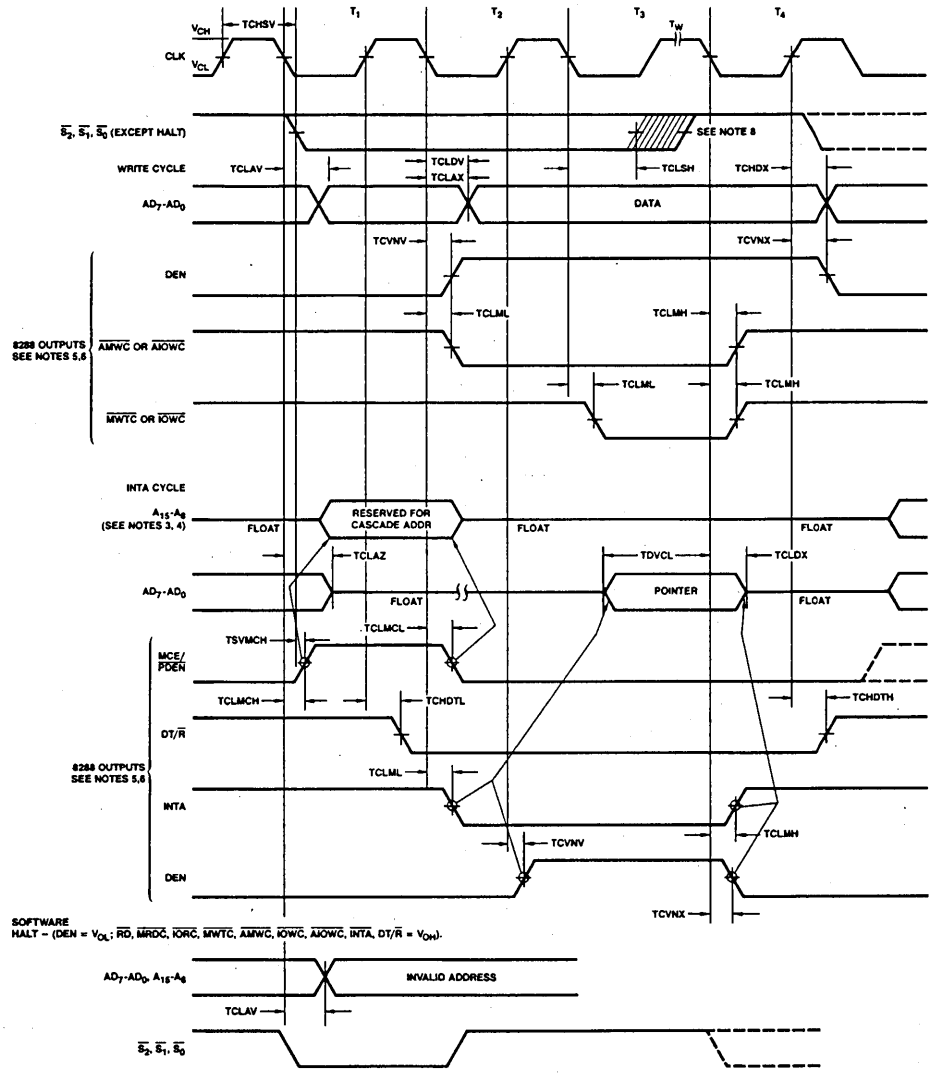
SWITCHING WAVEFORMS (continued)

BUS TIMING – MAXIMUM MODE



SWITCHING WAVEFORMS (continued)

BUS TIMING – MAXIMUM MODE SYSTEM (USING 8288)

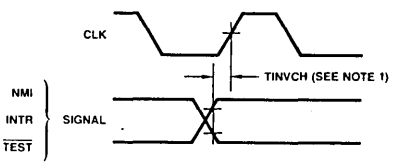


WF006801

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Cascade address is valid between first and second INTA cycles.
4. Two INTA cycles run back-to-back. The 8088 local ADDR/DATA bus is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.
5. Signals at 8284 or 8288 are shown for reference only.
6. The issuance of the 8288 command and control signals ($\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$, $\overline{INTA}$, and DEN) lags the active high 8288 CEN.
7. All timing measurements are made at 1.5 V unless otherwise noted.
8. Status inactive in state just prior to T_4.

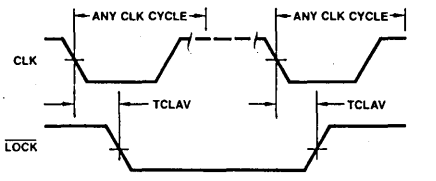
SWITCHING WAVEFORMS (continued)

ASYNCHRONOUS SIGNAL RECOGNITION



WF006820

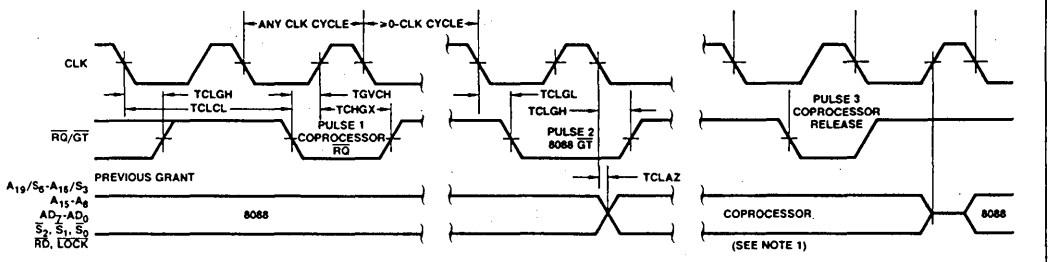
BUS LOCK SIGNAL TIMING (MAXIMUM MODE ONLY)



WF006830

Note: Set-up requirements for asynchronous signals only to guarantee recognition at next CLK.

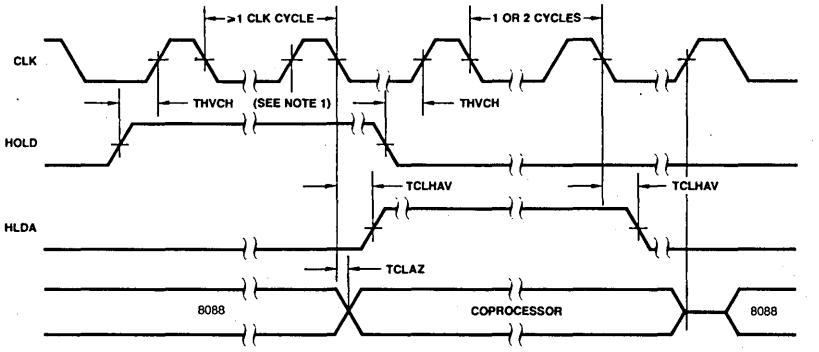
REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)



WF006840

Note: The coprocessor may not drive the buses outside the region shown without rising contention.

HOLD/HOLD ACKNOWLEDGE TIMING (MIMIMUM MODE ONLY)



WF006851

Note: All signals switch between V_{OH} and V_{OL} unless otherwise specified.

8086/8088
INSTRUCTION SET SUMMARY

DATA TRANSFER

MOV = Move	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/memory to/from register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate to register	1 0 1 1 w reg	data	data if w = 1	
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
PUSH = Push:				
Register/memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment register	0 0 0 reg 1 1 0			
POP = Pop:				
Register/memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment register	0 0 0 reg 1 1 1			
XCHG = Exchange:				
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m		
Register with accumulator	1 0 0 1 0 reg			
IN = Input from:				
Fixed port	1 1 1 0 0 1 0 w	port		
Variable port	1 1 1 0 1 1 0 w			
OUT = Output to:				
Fixed port	1 1 1 0 0 1 1 w	port		
Variable port	1 1 1 0 1 1 1 w			
XLAT = Transtate byte to AL	1 1 0 1 0 1 1 1			
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m		
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m		
LANF = Load AH with flags	1 0 0 1 1 1 1 1			
SANF = Store AH into flags	1 0 0 1 1 1 1 0			
PUSHF = Push flags	1 0 0 1 1 1 0 0			
POPF = Pop flags	1 0 0 1 1 1 0 1			

Mnemonics ©Intel, 1978

INSTRUCTION SET SUMMARY (continued)

ARITHMETIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
ADD = Add				
Reg/memory with register to either	0 0 0 0 0 0 d w	mod reg r/m		
Immediate to register / memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w = 1	
ADC = Add with carry:				
Reg/memory with register to either	0 0 0 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w = 1	
INC = Increment:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
Register	0 1 0 0 0 reg			
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1			
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1			
SUB = Subtract:				
Reg/memory and register to either	0 0 1 0 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 1 0 1 1 0 w	data	data if w = 1	
SBB = Subtract with borrow:				
Reg/memory and register to either	0 0 0 1 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w = 1	
DEC = Decrement:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m		
Register	0 1 0 0 1 reg			
NEG Change sign	1 1 1 1 0 1 1 w	mod 0 1 1 r/m		
CMP = Compare:				
Register/memory with register	0 0 1 1 1 0 1 w	mod reg r/m		
Register with register/memory	0 0 1 1 1 0 0 w	mod reg r/m		
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s:w = 01
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w = 1	
AAS ASCII adjust for subtract	0 0 1 1 1 1 1 1			
DAS Decimal adjust for subtract	0 0 1 0 1 1 1 1			
MUL Mulitiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m		
IMUL Integer multiply (signed):	1 1 1 1 0 1 1 w	mod 1 0 1 r/m		
AAM ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0		
DIV Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m		
IDIV Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m		
AAD ASCH adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0		
CBW Convert byte to word	1 0 0 1 1 0 0 0			
CWD Convert word to double word	1 0 0 1 1 0 0 1			

Mnemonics ©Intel, 1978

INSTRUCTION SET SUMMARY (continued)

LOGIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
NOT Invert	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		
SHL/SAL Shift logical/arithmetic left	1 1 0 1 0 0 v w	mod 1 0 0 r/m		
SHR Shift logical right	1 1 0 1 0 0 v w	mod 1 0 1 r/m		
SAR Shift arithmetic right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
ROL Rotate left	1 1 0 1 0 0 v w	mod 0 0 0 r/m		
ROR Rotate right	1 1 0 1 0 0 v w	mod 0 0 1 r/m		
RCL Rotate through carry flag left	1 1 0 1 0 0 v w	mod 0 1 0 r/m		
RCR Rotate through carry right	1 1 0 1 0 0 v w	mod 0 1 1 r/m		
AND = And:				
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1	
TEST = And function to flags, no result:				
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m		
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1	
OR = Or:				
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1	
XOR = Exclusive or:				
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1	

STRING MANIPULATION:

REP = Repeat	1 1 1 1 0 0 1 z
MOVS = Move byte/word	1 0 1 0 0 1 0 w
CMPS = Compare byte/word	1 0 1 0 0 1 1 w
SCAS = Scan byte/word	1 0 1 0 1 1 1 w
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w
STOS = Store byte/wd from AL/A	1 0 1 0 1 0 1 w

Mnemonics ©Intel, 1978

INSTRUCTION SET SUMMARY (continued)

CONTROL TRANSFER

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
CALL = Call				
Direct within segment	1 1 1 0 1 0 0 0	disp-low	disp-high	
indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0 r/m		
Direct intersegment	1 0 0 1 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m		
JMP = Unconditional jump:				
Direct within segment	1 1 1 0 1 0 0 1	disp-low	disp-high	
Direct within segment-short	1 1 1 0 1 0 1 1	disp		
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0 r/m		
Direct intersegment	1 1 1 0 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m		
RET = Return from CALL:				
Within segment	1 1 0 0 0 0 1 1			
Within segment adding immediate to SP	1 1 0 0 0 0 1 0	data-low	data-high	
Intersegment	1 1 0 0 1 0 1 1			
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low	data-high	
JE/JZ = Jump on equal/zero	0 1 1 1 0 1 0 0	disp		
JL/JNGE = Jump on less/not greater or equal	0 1 1 1 1 1 0 0	disp		
JLE/JNG = Jump on less or equal/not greater	0 1 1 1 1 1 1 0	disp		
JB/JNAE = Jump on below/not above or equal	0 1 1 1 0 0 1 0	disp		
JBE/JNA = Jump on below or equal/not above	0 1 1 1 0 1 1 0	disp		
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0	disp		
JO = Jump on overflow	0 1 1 1 0 0 0 0	disp		
JS = Jump on sign	0 1 1 1 1 0 0 0	disp		
JNE/JNZ = Jump on not equal/not zero	0 1 1 1 0 1 0 1	disp		
JNL/JGE = Jump on not less/greater or equal	0 1 1 1 1 1 0 1	disp		
JNLE/JG = Jump on not less or equal/greater	0 1 1 1 1 1 1 1	disp		
JNB/JAE = Jump on not below/above or equal	0 1 1 1 0 0 1 1	disp		
JNBE/JA = Jump on not below or equal/above	0 1 1 1 0 1 1 1	disp		
JNP/JPO = Jump on not par/par odd	0 1 1 1 1 0 1 1	d isp		
JNO = Jump on not overflow	0 1 1 1 0 0 0 1	disp		
JNS = Jump on not sign	0 1 1 1 1 0 0 1	disp		
LOOP = Loop CX times	1 1 1 0 0 0 1 0	disp		
LOOPZ/LOOPE = Loop while zero/equal	1 1 1 0 0 0 0 1	disp		
LOOPNZ/LOOPNE = Loop while not zero/equal	1 1 1 0 0 0 0 0	disp		
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1	disp		

Mnemonics ©Intel, 1978

CONTROL TRANSFER (continued)

INT = Interrupt	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Type specified	1 1 0 0 1 1 0 1	type		
Type 3	1 1 0 0 1 1 0 0			
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0			
IRET = Interrupt return	1 1 0 0 1 1 1 1			

PROCESSOR CONTROL

CLC = Clear carry	1 1 1 1 1 0 0 0			
CMC = Complement carry	1 1 1 1 0 1 0 1			
STC = Set carry	1 1 1 1 1 0 0 1			
CLD = Clear direction	1 1 1 1 1 1 0 0			
STD = Set direction	1 1 1 1 1 1 0 1			
CLI = Clear interrupt	1 1 1 1 1 0 1 0			
STI = Set interrupt	1 1 1 1 1 0 1 1			
HLT = Halt	1 1 1 1 0 1 0 0			
WAIT = Wait	1 0 0 1 1 0 1 1			
ESC = Processor Extension Escape	1 1 0 1 1 x x x	mod x x x r/m		
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0			

Footnotes:

AL = 8-bit accumulator
AX = 16-bit accumulator
CX = Count register
DS = Data segment
ES = Extra segment
Above/below refers to unsigned value.
Greater = more positive.
Less = less positive (more negative) signed values
if d = 1 then "to" reg; if d = 0 then "from" reg
w = 1 then word instruction; if w = 0 then byte instruction

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0 , disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

if s:w = 01 then 16 bits of immediate data form the operand.
if s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand.
if v = 0 then "count" = 1; if v = 1 then "count" in (CL)
x = don't care
z is used for string primitives for comparison with Z.F Flag.

SEGMENT OVERRIDE PREFIX

0	0	1	reg	1	1	0

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register files as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS = X:X:X:X:(OF):(DF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

Mnemonics ©Intel, 1978

Final

80286

High-Performance Microprocessor with Memory Management and Protection

Advanced Micro Devices

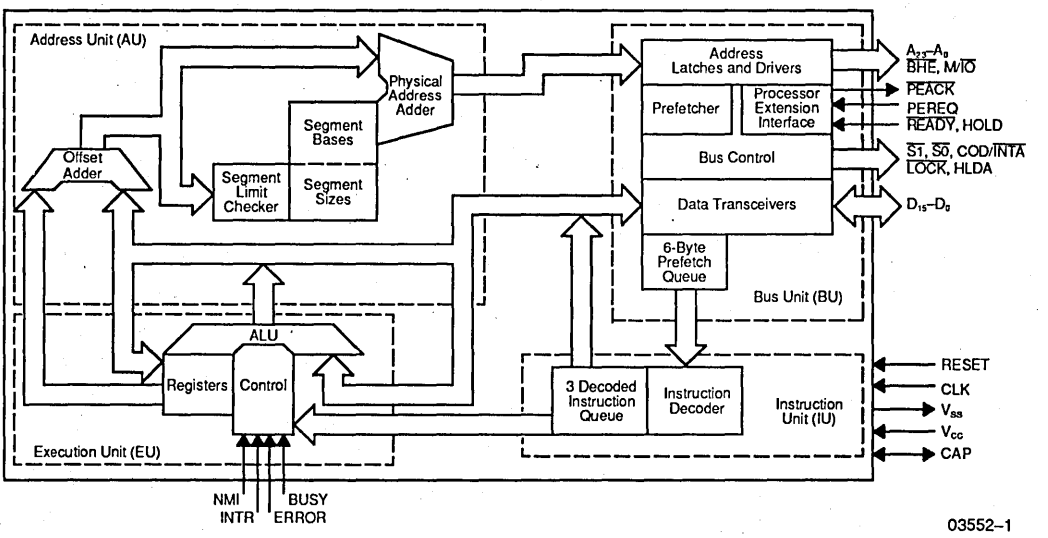
DISTINCTIVE CHARACTERISTICS

- **High-performance processor (up to 13.3 times iAPX 86 when using the 16 MHz 80286)**
- **Large address space**
  - –16 megabytes physical
  - –1 gigabyte virtual memory per task
- **Integrated memory management, four-level memory protection and support for virtual memory and operating systems**
- **Two iAPX 86 upward-compatible operating modes**
  - –iAPX 86 real address mode
  - –Protected virtual address mode
- **High bandwidth bus interface (16 megabyte/sec)**
- **Range of clock rates**
  - – 8 MHz 80286–8
  - –10 MHz 80286–10
  - –12 MHz 80286–12
  - –16 MHz 80286–16

GENERAL DESCRIPTION

The 80286 is an advanced, high-performance microprocessor with specially optimized capabilities for multiple user and multi-tasking systems. The 80286 has built-in memory protection that supports operating system and task isolation as well as program and data privacy within tasks. A 16-MHz 80286 provides up to 13.3 times greater throughput than the standard 5-MHz 8086. The 80286 includes memory management capabilities that map up to 2^{30} bytes (one gigabyte) of virtual address space per task into 2^{24} bytes (16 megabytes) of physical memory.

BLOCK DIAGRAM



03552–1

Publication # 03552 Rev. F Amendment /0
Issue Date: June 1989

GENERAL DESCRIPTION (continued)

The 80286 is upward-compatible with iAPX 86 and 88 software. Using iAPX 86 real address mode, the 80286 is object-code compatible with existing iAPX 86, 88 software.

In protected virtual address mode, the 80286 is source-code compatible with iAPX 86, 88 software and may require upgrading to use virtual addresses supported by the 80286's integrated memory management and protection mechanism. Both modes operate at full 80286 performance and execute a superset of the iAPX 86 and 88 instructions.

The 80286 provides special operations to support the efficient implementation and execution of operating systems. For example, one instruction can end execution of one task, save its state, switch to a new task, load its state, and start execution of the new task. The 80286 also supports virtual memory systems by providing a segment-not-present exception and restartable instructions.

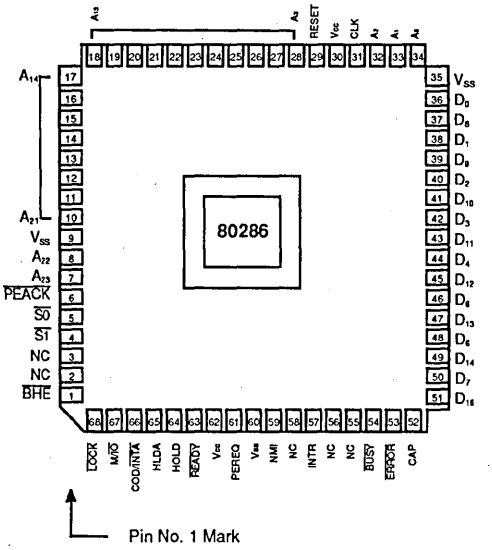
Related AMD Products

Part No.	Description
82284	Clock Driver
82C54	Programmable Interval Timer
Am9517A	DMA Controller

CONNECTION DIAGRAMS

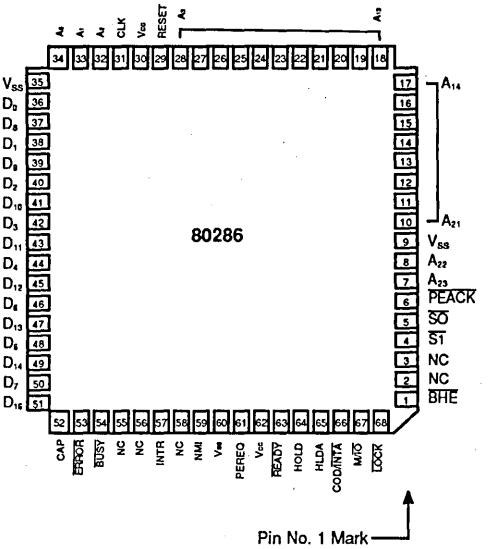
LCC

Component Pad Views—As viewed from underside of component on the PC Board



03552-2

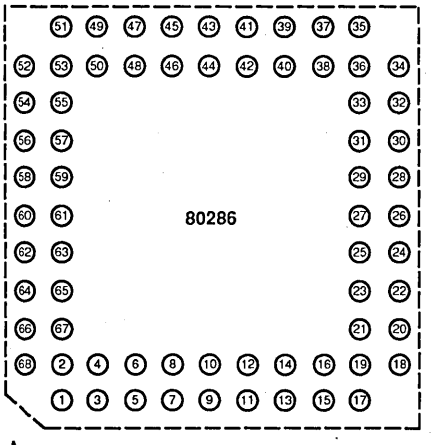
PC Board Views—As viewed from the component side of the PC Board



There are no electrical connections on the bottom of this package

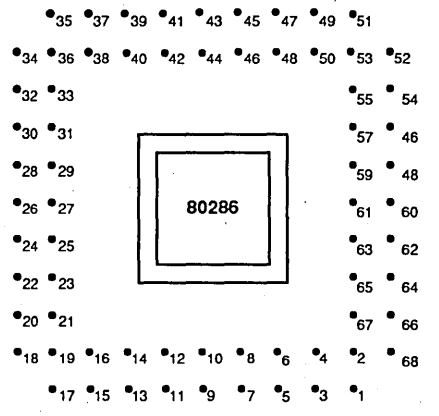
03552-3

PGA



03552-4

Pins pointing away from viewer

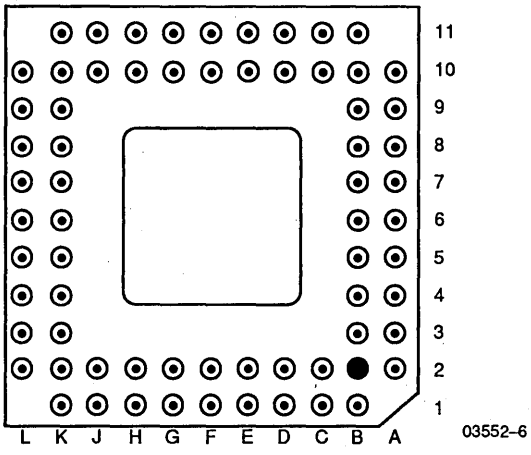


03552-5

Pins pointing toward viewer

PGA (continued)

Bottom View



PIN DESIGNATIONS
(Sorted by pin number)

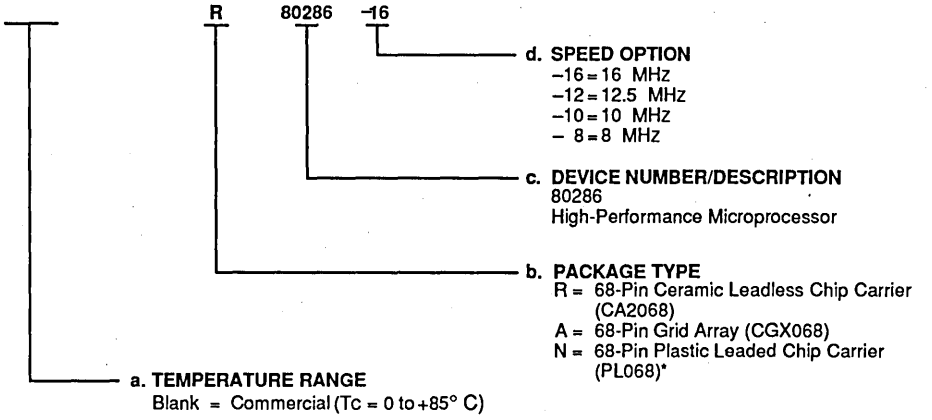
Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	$\overline{BHE}$	24	A_7	47	D_{13}
2	NC	25	A_6	48	D_6
3	NC	26	A_5	49	D_{14}
4	$\overline{S1}$	27	A_4	50	D_7
5	$\overline{S0}$	28	A_3	51	D_{15}
6	$\overline{PEACK}$	29	RESET	52	CAP
7	A_{23}	30	V_{CC}	53	$\overline{ERROR}$
8	A_{22}	31	CLK	54	$\overline{BUSY}$
9	V_{SS}	32	A_2	55	NC
10	A_{21}	33	A_1	56	NC
11	A_{20}	34	A_0	57	INTR
12	A_{19}	35	V_{SS}	58	NC
13	A_{18}	36	D_0	59	NMI
14	A_{17}	37	D_8	60	V_{SS}
15	A_{16}	38	D_1	61	PEREQ
16	A_{15}	39	D_9	62	V_{CC}
17	A_{14}	40	D_2	63	$\overline{READY}$
18	A_{13}	41	D_{10}	64	HOLD
19	A_{12}	42	D_3	65	HLDA
20	A_{11}	43	D_{11}	66	$COD/\overline{INTA}$
21	A_{10}	44	D_4	67	$M/\overline{IO}$
22	A_9	45	D_{12}	68	$\overline{LOCK}$
23	A_8	46	D_5		

ORDERING INFORMATION

Commodity Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option (if applicable)
e. Optional Processing



Valid Combinations	
A	80286–8 80286–10 80286–12
R	80286–8 80286–10 80286–12 80286–16

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*The PLCC package is not a valid ordering part number for the 80286. The PLCC package is valid for the 80L286 part number. See the 80L286 data sheet (order #08511D) for ordering information and DC and AC parametrics.

PIN DESCRIPTION

CLK

System Clock (Input; Active HIGH)

System Clock provides the fundamental timing for 80286 systems. It is divided by two inside the 80286 to generate the processor clock. The internal divide-by-two circuitry can be synchronized to an external clock generator by LOW-to-HIGH transition on the RESET input.

D_0–D_{15}

Data Bus (Input/Output; Active HIGH)

Data Bus inputs data during memory, I/O, and interrupt acknowledge read cycles; outputs data during memory and I/O write cycles. The data bus is active HIGH and floats to three-state OFF during bus hold acknowledge.

A_{23}–A_0

Address Bus (Output; Active HIGH)

Address Bus outputs physical memory and I/O port addresses. A_0 is LOW when data is to be transferred on pins D_{7-0}. A_{23}–A_{16} are LOW during I/O transfers. The address bus is active HIGH and floats to three-state OFF during bus hold acknowledge.

$\overline{BHE}$

Bus High Enable (Output; Active LOW)

Bus High Enable indicates transfer of data on the upper byte of the data bus D_{15-8}. Eight-bit oriented devices assigned to the upper byte of the data bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is active LOW and floats to three-state OFF during bus hold acknowledge.

$\overline{BHE}$ and A_0 Encodings

$\overline{BHE}$ Value	A_0 Value	Function
0	0	Word transfer
0	1	Byte transfer on upper half of data bus (D_{15-8})
1	0	Byte transfer on lower half of data bus (D_{7-0})
1	1	Reserved

$\overline{S1}$, $\overline{S0}$

Bus Cycle Status (Output; Active LOW) Bus Cycle Status indicates initiation of a bus cycle and, along with M/$\overline{IO}$ and COD/$\overline{INTA}$, defines the type of bus cycle. The bus is in a T_S state whenever one or both are LOW. $\overline{S1}$ and $\overline{S0}$ are active LOW and float to three-state OFF during bus hold acknowledge.

80286 Bus Cycle Status Definition

COD/$\overline{INTA}$	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Bus Cycle Status Definition
0 (LOW)	0	0	0	Interrupt acknowledge
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	None; not a status cycle
0	1	0	0	If $A_1 = 1$ then halt; else shutdown
0	1	0	1	Memory data read
0	1	1	0	Memory data write
0	1	1	1	None; not a status cycle
1 (HIGH)	0	0	0	Reserved
1	0	0	1	I/O Read
1	0	1	0	I/O Write
1	0	1	1	None; not a status cycle
1	1	0	0	Reserved
1	1	0	1	Memory instruction read
1	1	1	0	Reserved
1	1	1	1	None; not a status cycle

M/$\overline{IO}$

Memory/$\overline{IO}$ Select (Output)

Memory/$\overline{IO}$ Select distinguishes memory access from I/O access. If HIGH during T_S, a memory cycle or a halt/shutdown cycle is in progress. If LOW, an I/O cycle or an interrupt acknowledge cycle is in progress. M/$\overline{IO}$ floats to three-state OFF during bus hold acknowledge.

COD/$\overline{INTA}$

Code/Interrupt Acknowledge (Output)

Code/Interrupt Acknowledge distinguishes instruction fetch cycles from memory data read cycles. Also distinguishes interrupt acknowledge cycles from I/O cycles. COD/$\overline{INTA}$ floats to three-state OFF during bus hold acknowledge.

$\overline{LOCK}$

Bus Lock (Output; Active LOW)

Bus Lock indicates that other system bus masters are not to gain control of the system bus following the current bus cycle. The $\overline{LOCK}$ signal may be activated explicitly by the "LOCK" instruction prefix or automatically by 80286 hardware during memory XCHG instructions, interrupt acknowledge, or descriptor table access. $\overline{LOCK}$ is active LOW and floats to three-state OFF during hold acknowledge.

PIN DESCRIPTION (continued)

$\overline{\text{READY}}$

Bus Ready (Input; Active LOW)

Bus Ready terminates a bus cycle. Bus cycles are extended without limit until terminated by $\overline{\text{READY}}$ LOW. $\overline{\text{READY}}$ is an active LOW synchronous input requiring set-up and hold times relative to the system clock be met for correct operation. $\overline{\text{READY}}$ is ignored during bus hold acknowledge.

HOLD, HLDA

Bus Hold Request and Hold Acknowledge (Input/Output; Active HIGH)

Bus Hold Request and Hold Acknowledge control ownership of the 80286 local bus. The HOLD input allows another local bus master to request control of the local bus. When control is granted, the 80286 will float its bus drivers to three-state OFF and then active HLDA, thus entering the bus hold acknowledge condition. The local bus will remain granted to the requesting master until HOLD becomes inactive which results in the 80286 deactivating HLDA and regaining control of the local bus. This terminates the bus hold acknowledge condition. HOLD may be asynchronous to the system clock. These signals are active HIGH.

INTR

Interrupt Request (Input; Active HIGH)

Interrupt Request requests the 80286 to suspend its current program execution and service a pending external request. Interrupt requests are masked whenever the interrupt enable bit in the flag word is cleared. When the 80286 responds to an interrupt request, it performs two interrupt acknowledge bus cycles to read an 8-bit interrupt vector that identifies the source of the interrupt. To assure program interruption, INTR must remain active until the first interrupt acknowledge cycle is completed. INTR is sampled at the beginning of each processor cycle and must be active HIGH at least two processor cycles before the current instruction ends in order to interrupt before the next instruction. INTR is level sensitive, active HIGH, and may be asynchronous to the system clock.

NMI

Non-maskable Interrupt Request (Input; Active HIGH)

Non-maskable Interrupt Request interrupts the 80286 with an internally supplied vector value of 2. No interrupt acknowledge cycles are performed. The interrupt enable bit in the 80286 flag word does not affect this input. The NMI input is active HIGH, may be asynchronous to the system clock, and is edge triggered after internal synchronization. For proper recognition, the input must have been previously LOW for at least four system clock cycles and remain HIGH for at least four system clock cycles.

PEREQ, $\overline{\text{PEACK}}$

Processor Extension Operand Request and Acknowledge (Input/Output)

Processor Extension Operand Request and Acknowledge extended the memory management and protection capabilities of the 80286 to processor extensions. The PEREQ input requests the 80286 to perform a data operand transfer for a processor extension. The $\overline{\text{PEACK}}$ output signals the processor extension when the requested operand is being transferred. PEREQ is active HIGH and may be asynchronous to the system clock. $\overline{\text{PEACK}}$ is active LOW.

$\overline{\text{BUSY}}$, $\overline{\text{ERROR}}$

Processor Extension Busy and Error (Input/Input, Active Low)

Processor Extension Busy and Error indicate the operating condition of a processor extension to the 80286. An active $\overline{\text{BUSY}}$ input stops 80286 program execution on WAIT and some ESC instructions until $\overline{\text{BUSY}}$ becomes inactive (HIGH). The 80286 may be interrupted while waiting for $\overline{\text{BUSY}}$ to become inactive. An active $\overline{\text{ERROR}}$ input causes the 80286 to perform a processor extension interrupt when executing WAIT or some ESC instructions. These inputs are active LOW and may be asynchronous to the system clock.

RESET

System Reset (Input; Active HIGH)

System Reset clears the internal logic of the 80286 and is active HIGH. The 80286 may be reinitialized at any time with a LOW-to-HIGH transition on RESET which remains active for more than 16 system clock cycles. During RESET active, the output pins of the 80286 enter the state shown below:

80286 Pin State During Reset

Pin Value	Pin Names
1 (HIGH)	$\overline{S0}$, $\overline{S1}$, $\overline{\text{PEACK}}$, A_{23}–A_0, $\overline{\text{BHE}}$, $\overline{\text{LOCK}}$
0 (LOW)	M/$\overline{IO}$, COD/$\overline{INTA}$, HLDA
Three-state OFF	D_{15}–D_0

Operation of the 80286 begins after a HIGH-to-LOW transition on RESET. The HIGH-to-LOW transition of RESET must be synchronous to the system clock. Approximately 50 system clock cycles are required by the 80286 for internal initializations before the first bus cycle to fetch code from the power-on execution address is performed.

A LOW-to-HIGH transition of RESET synchronous to the system clock will begin a new processor cycle at the next HIGH-to-LOW transition of the system clock. The LOW-to-HIGH transition of RESET may be asynchronous to the system clock; however, in this case it cannot be predetermined which phase of the processor clock will occur during the next system period. Synchronous LOW-to-HIGH transitions of RESET are only required for systems where the processor clock must be phase synchronous to another clock.

V_{SS}
System Ground (Input; Active HIGH)

System Ground: 0 volts.

V_{CC}
System Power (Input; Active HIGH)

System Power: +5 volt power supply.

CAP
Substrate Filter Capacitor (Input; Active High)

A 0.047 µF ±20% 12 V capacitor must be connected between this pin and ground. This capacitor filters the output of the internal substrate bias generator. A maximum DC leakage current of 1 µA is allowed through the capacitor.

For correct operation of the 80286, the substrate bias generator must charge this capacitor to its operating voltage. The capacitor charge-up time is 5 milliseconds (max.) after V_{CC} and CLK reach their specified AC and DC parameters. RESET may be applied to prevent spurious activity by the CPU during this time. After this time, the 80286 processor clock can be phase synchronized to another clock by pulsing RESET LOW synchronous to the system clock.

FUNCTIONAL DESCRIPTION

Introduction

The 80286 is an advanced, high-performance microprocessor with specially optimized capabilities for multiple user and multi-tasking systems. Depending on the application, the 80286's performance is up to 13.3 times faster than the standard 5-MHz 8086's, while providing complete upward software compatibility with AMD's iAPX 86, 88, and 186 family of CPUs.

The 80286 operates in two modes: iAPX 86 real address mode and protected virtual address mode. Both modes execute a superset of the iAPX 86 and 88 instruction set.

In iAPX 86 real address mode programs use real addresses with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16-megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. Both modes provide the same base instruction set, registers, and addressing modes.

The following pages describe first, the base 80286 architecture common to both modes; second, iAPX 86 real address mode; and third, protected mode.

80286 Base Architecture

The iAPX 86, 88, 186, and 286 CPU family all contain the same basic set of registers, instructions, and addressing modes. The 80286 processor is upward-compatible with the 8086, 8088, and 80186 CPUs.

Register Set

The 80286 base architecture has fifteen registers as shown in Figure 1. These registers are grouped into the following four categories:

General Registers: Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used either in their entirety as 16-bit words or split into pairs of separate 8-bit registers.

Segment Registers: Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers: Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode determines the specific registers used for operand address calculations.

Status and Control Registers: Three 16-bit special purpose registers record or control certain aspects of the 80286 processor state. These include the Instruction Pointer, which contains the offset address of the next sequential instruction to be executed.

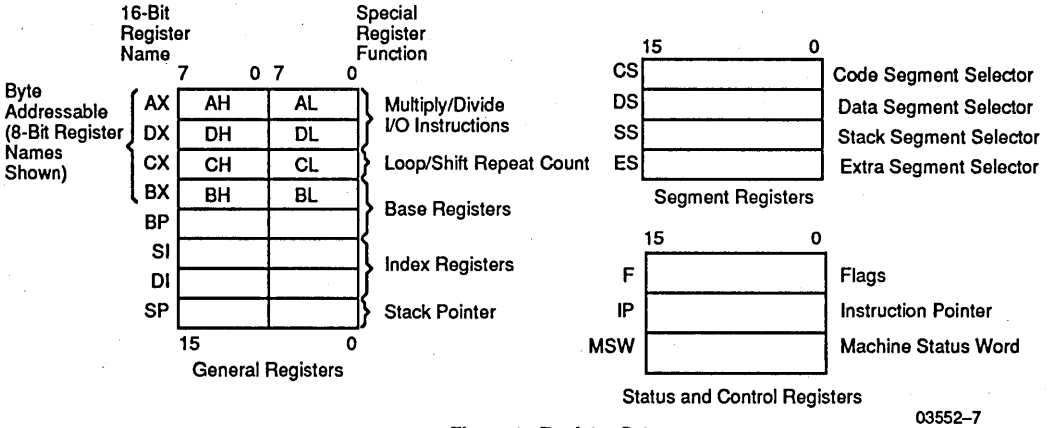


Figure 1. Register Set

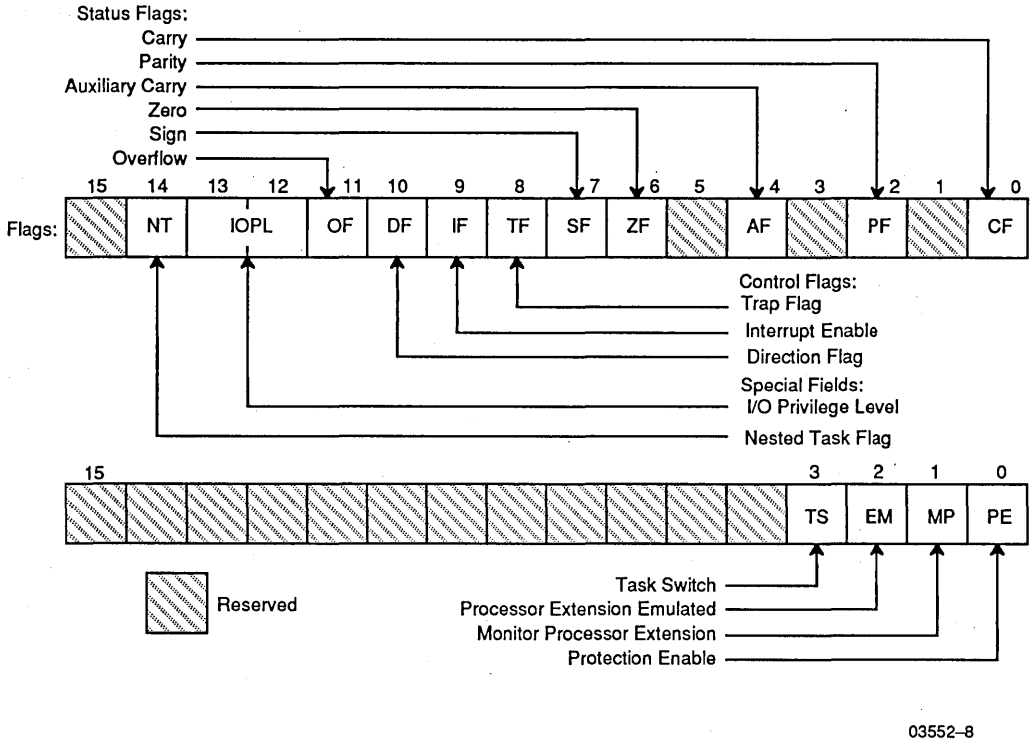


Figure 2. Status and Control Register Bit Functions

Flags Word Description

The Flags word (Flags) records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80286 within a given operating mode (bits 8 and 9). Flags is a 16-bit register. The function of the flag bits is given in Table 1.

Table 1. Flags Word Bit Functions

Bit Position	Name	Function
0	CF	Carry Flag—Set on high-order bit carry or borrow; cleared otherwise
2	PF	Parity Flag—Set if low-order 8 bits of result contain an even number of 1 bits; cleared otherwise
4	AF	Set on carry-from or borrow-to the low-order four bits of AL; cleared otherwise
6	ZF	Zero Flag—Set if result is zero; cleared otherwise
7	SF	Sign Flag—Set equal to high-order bit of result (0 if positive, 1 if negative)
11	OF	Overflow Flag—Set if result is a too-large positive number or a too-small negative number (excluding sign-bit) to fit in destination operand; cleared otherwise
8	TF	Single Step Flag—Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt
9	IF	Interrupt-Enable Flag—When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location
10	DF	Direction Flag—Causes string instructions to auto-decrement the appropriate index registers when set. Clearing DF causes auto increment.

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, program transfer, high-level instructions, and processor control. These categories are summarized in Figures 3–9.

An 80286 instruction can reference zero, one, or two operands where an operand resides in a register, in the instruction itself, or in memory. Zero-operand instructions (e.g., NOP and HLT) are usually one byte long. One-operand instructions (e.g., INC and DEC) are usually two bytes long, but some are encoded in only one byte. One-operand instructions may reference a register or memory location. Two-operand instructions permit the following six types of instruction operations:

Register to Register
Memory to Register
Immediate to Register
Memory to Memory
Register to Memory
Immediate to Memory

Two-operand instructions (e.g., MOV and ADD) are usually three to six bytes long. Memory to memory operations are provided by a special class of string instructions requiring one to three bytes. For detailed instruction formats and encodings, refer to the instruction set summary at the end of this document.

General Purpose	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
Input/Output	
IN	Input byte or word
OUT	Output byte or word
Address Object	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
Flag Transfer	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

Figure 3. Data Transfer Instructions

Addition	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
Subtraction	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
Multiplication	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
Division	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to double word

Figure 4. Arithmetic Instructions

MOVS	Move byte or word string
INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/REPNZ	Repeat while not equal/not zero

Figure 5. String Instructions

Logicals	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
Shifts	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
Rotates	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

Figure 6. Shift/Rotate/Logical

Conditional Transfers		Unconditional Transfers	
JA/JNBE	Jump if above/not below nor equal	CALL	Call procedure
JAE/JNB	Jump if above or equal/not below	RET	Return from procedure
JB/JNAE	Jump if below/not above nor equal	JMP	Jump
JBE/JNA	Jump if below or equal/not above		
JC	Jump if carry	**Iteration Controls**	
JE/JZ	Jump if equal/zero		
JG/JNLE	Jump if greater/not less nor equal	LOOP	Loop
JGE/JNL	Jump if greater or equal/not less	LOOPE/LOOPZ	Loop if equal/zero
JL/JNGE	Jump if less/not greater nor equal	LOOPNE/LOOPNZ	Loop if not equal/not zero
JLE/JNG	Jump if less or equal/not greater	JCXZ	Jump if register CX = 0
JNC	Jump if not carry		
JNE/JNZ	Jump if not equal/not zero	**Interrupts**	
JNO	Jump if not overflow		
JNP/JPO	Jump if not parity/parity odd	INT	Interrupt
JNS	Jump if not sign	INTO	Interrupt if overflow
JO	Jump if overflow	IRET	Interrupt return
JP/JPE	Jump if parity/parity even		
JS	Jump if sign		

Figure 7. Program Transfer Instructions

Flag Operations	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
External Synchronization	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{\text{BUSY}}$ not active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
No Operation	
NOP	No operation
Execution Environment Control	
LMSW	Load machine status word
SMSW	Store machine status word

Figure 8. Processor Control Instructions

ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 9. High-Level Instructions

Memory Organization

Memory is organized as sets of variable length segments. Each segment is a linear contiguous sequence of up to 64K(2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit segment selector and a 16-bit offset. The segment selector indicates the desired segment in memory. The offset component indicates the desired byte address within the segment.

All instructions that address operands in memory must specify the segment and the offset. For speed and compact instruction encoding, segment selectors are usually stored in the high speed segment registers. An instruction need specify only the desired segment register and an offset to address a memory operand.

Most instructions need not explicitly specify which segment register is used. The correct segment register is automatically chosen according to the rules of Table 2. These rules follow the way programs are written (see Figure 11) as independent modules that require areas for code and data, a stack, and access to external data areas.

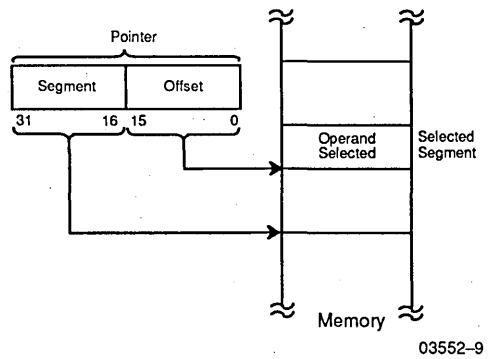


Figure 10. Two-Component Address

Table 2. Segment Register Selection Rules

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Automatic with instruction prefetch
Stack	Stack (SS)	All stack pushes and pops. Any memory reference which uses BP as a base register.
Local Data	Data (DS)	All data references except when relative to stack or string destination.
External (Global) Data	Extra (ES)	Alternate data segment and destination of string operation.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs. To access operands that do not reside in one of the four immediately available segments, either a full 32-bit pointer can be used or a new segment selector must be loaded.

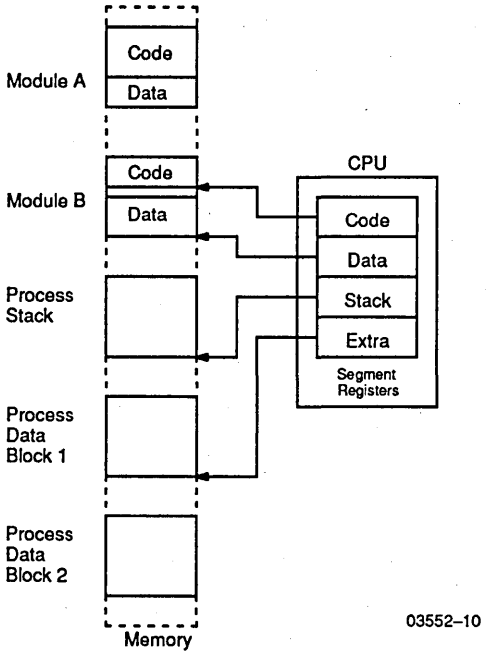


Figure 11. Segmented Memory Helps Structure Software

Addressing Modes

The 80286 provides a total of eight addressing modes for instructions to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

Register Operand Mode: The operand is located in one of the 8- or 16-bit general registers.

Immediate Operand Mode: The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: segment selector and offset. The segment selector is supplied by a segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset is calculated by summing any combination of the following three address elements:

the displacement (an 8- or 16-bit immediate value contained in the instruction)

the base (contents of either the BX or BP base registers)

the index (contents of either the SI or DI index registers)

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes here described:

Direct Mode: The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.

Register Indirect Mode: The operand's offset is in one of the registers SI, DI, BX, or BP.

Based Mode: The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).

Indexed Mode: The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).

Based Indexed Mode: The operand's offset is the sum of the contents of a base register and an index register.

Based Indexed Mode with Displacement: The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

The 80286 directly supports the following data types:

Integer: A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a two's complement representation. Signed 32- and 64-bit integers are supported using the 80287 Numeric Data Processor.

Ordinal: An unsigned binary numeric value contained in an 8-bit byte or 16-bit word.

Pointer: A 32-bit quantity, composed of a segment selector component and an offset component. Each component is a 16-bit word.

String: A contiguous sequence of bytes or words. A string may contain from 1 byte to 64K bytes.

ASCII: A byte representation of alphanumeric and control characters using the ASCII standard of character representation.

BCD: A byte (unpacked) representation of the decimal digits 0–9.

Packed BCD: A byte (packed) representation of two decimal digits 0–9 storing one digit in each nibble of the byte.

Floating Point: A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using the iAPX 287 Numeric Processor configuration.)

Figure 12 graphically represents the data types supported by the 80286.

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. I/O instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register. Eight-bit port addresses are zero extended such that A_{15}–A_8 are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Flags) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable. Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. The return address from an exception will always point at the instruction causing the exception and include any leading instruction prefixes.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0–31, some of which are used for instruction exceptions, are reserved. For each interrupt, an 8-bit vector must be supplied to the 80286 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware-initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Maskable Interrupt (INTR)

The 80286 provides a maskable hardware interrupt request pin, INTR. Software enables this input by setting the interrupt flag bit (IF) in the flag word. All 224 user-defined interrupt sources can share this input, yet they can retain separate interrupt handlers. An 8-bit vector read by the CPU during the interrupt acknowledge sequence (discussed in the System Interface section) identifies the source of the interrupt.

Further maskable interrupts are disabled while servicing an interrupt by resetting the IF but as part of the response to an interrupt or exception. The saved flag word will reflect the enable status of the processor prior to the interrupt. Until the flag word is restored to the flag register, the interrupt flag will be zero unless specifically set. The interrupt return instruction includes restoring the flag word, thereby restoring the original status of IF.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt input (NMI) is also provided. NMI has higher priority than INTR. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed.

While executing the NMI servicing procedure, the 80286 will not service further NMI requests, INTR requests, or the processor extension segment overrun interrupt until an interrupt return (IRET) instruction is executed or the CPU is reset. If NMI occurs while currently servicing an NMI, its presence will be saved for servicing after executing the first IRET instruction. IF is cleared at the beginning of an NMI interrupt to inhibit INTR interrupts.

Single Step Interrupt

The 80286 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single step interrupt and is controlled by the single step flag bit (TF) in the flag word. Once this bit is set, an internal single step interrupt will occur after the next instruction

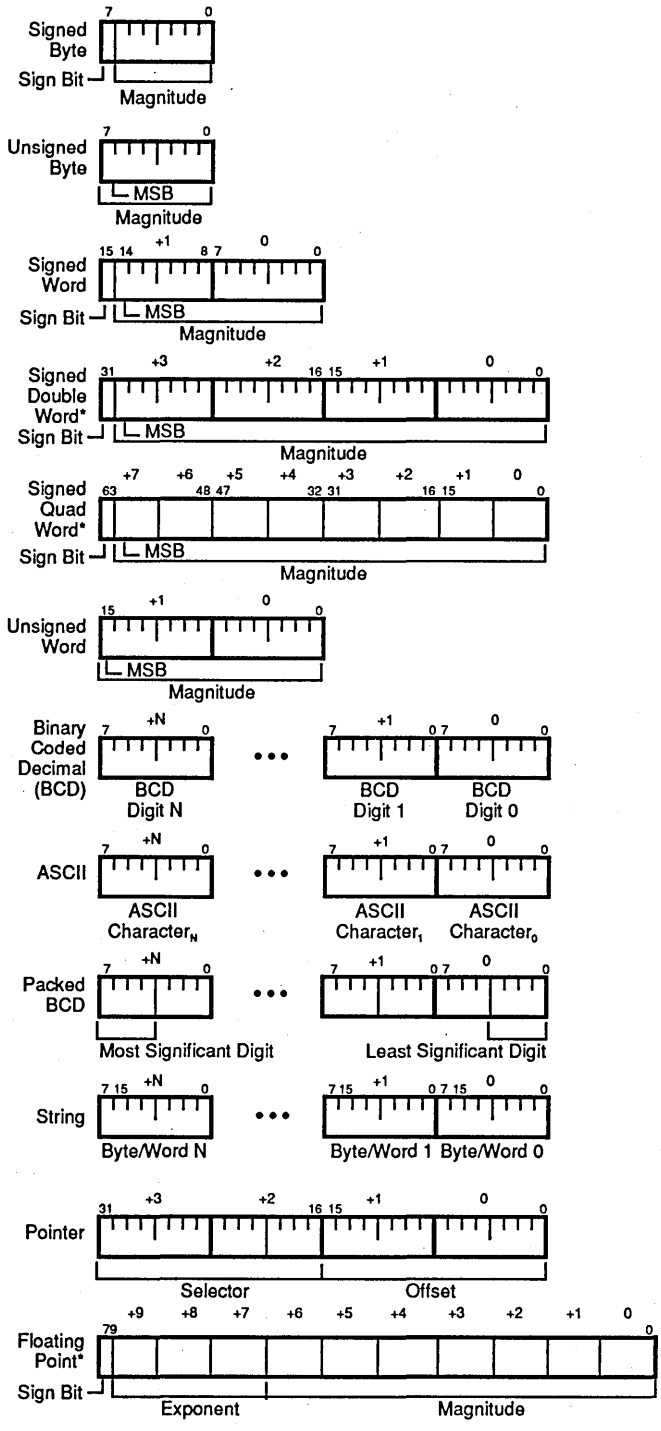


Figure 12. 80286 Supported Data Types

*Supported by iAPX 286/287 Numeric Data Processor Configuration

03552–11

Table 3. Interrupt Vector Assignments

Function	Interrupt Number	Related Instructions	Return Address Before Instruction Causing Exception?
Divide error exception	0	DIV, IDIV	Yes
Single step interrupt	1	All	
NMI interrupt	2	All	
Breakpoint interrupt	3	INT	
INTO detected overflow exception	4	INTO	No
BOUND range exceeded exception	5	BOUND	Yes
Invalid opcode exception	6	Any undefined opcode	Yes
Processor extension not available exception	7	ESC or WAIT	Yes
Reserved	8–15		
Processor extension error input	16	ESC or WAIT	
Reserved	17–31		
User-defined	32–255		

Table 4. Interrupt Processing Order

Order	Interrupt
1	INT instruction or exception
2	Single step
3	NMI
4	Processor extension segment overrun
5	INTR

has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single stepped.

Interrupt Priorities

When simultaneous interrupt requests occur, they are processed in a fixed order as shown in Table 4. Interrupt processing involves saving the flags, return address, and setting CS:IP to point at the first instruction of the interrupt handler. If other interrupts remain enabled, they are processed before the first instruction of the current interrupt handler is executed. The last interrupt processed is therefore the first one serviced.

Initialization and Processor Reset

Processor initialization or start up is accomplished by driving the RESET input pin HIGH. RESET forces the 80286 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as RESET is active. After RESET becomes inactive and an internal processing interval elapses, the 80286 begins execution in real address mode with the instruction at physical location FFFFF0(H). RESET also sets some registers to predefined values as shown in Table 5.

Machine Status Word Description

The machine status word (MSW) records when a task switch takes place and controls the operating mode of the 80286. It is a 16-bit register of which the lower four bits are used. One bit places the CPU into protected mode, while the other three bits, as shown in Table 6, control the processor extension interface. After RESET, this register contains FFF0(H) which places the 80286 in iAPX 86 real address mode.

Table 5. 80286 Initial Register State after RESET

Flag word	0002(H)
Machine Status Word	FFF0(H)
Instruction pointer	FFF0(H)
Code segment	F000(H)
Data segment	0000(H)
Extra segment	0000(H)
Stack segment	0000(H)

The LMSW and SMSW instructions can load and store the MSW in real address mode. The recommended use of TS, EM, and MP is shown in Table 7.

Table 6. MSW Bit Functions

Bit Position	Name	Function
0	PE	Protected mode Enable places the 80286 into protected mode and cannot be cleared except by RESET.
1	MP	Monitor Processor extension allows WAIT instructions to cause a processor extension not present exception (number 7).
2	EM	Emulate processor extension causes a processor extension not present exception (number 7) on ESC instructions to allow emulating a processor extension.
3	TS	Task Switched indicates the next instruction using a processor extension will cause exception 7, allowing software to test whether the current processor extension context belongs to the current task.

Table 7. Recommended MSW Encodings For Processor Extension Control

TS	MP	EM	Recommended Use	Instructions Causing Exception
0	0	0	iAPX 86 real address mode only. Initial encoding after RESET. 80286 operation is identical to iAPX 86, 88.	None
0	0	1	No processor extension is available. Software will emulate its function.	ESC
1	0	1	No processor extension is available. Software will emulate its function. The current processor extension context may belong to another task.	ESC
0	1	0	A processor extension exists.	None
1	1	0	A processor extension exists. The current processor extension context may belong to another task. The exception on WAIT allows software to test for an error pending from a previous processor extension operation.	ESC or WAIT

Halt

The HLT instruction stops program execution and prevents the CPU from using the local bus until restarted. Either NMI, INTR with IF = 1, or RESET will force the 80286 out of halt. If interrupted, the saved CS:IP will point to the next instruction after the HLT.

iAPX 286 Real Address Mode

The 80286 executes a fully upward-compatible superset of the 8086 instruction set in real address mode. In real address mode the 80286 is object code compatible with 8086 and 8088 software. The real address mode architecture (registers and addressing modes) is exactly as described in the 80286 Base Architecture section.

Memory Size

Physical memory is a contiguous array of up to 1,048,576 bytes (one megabyte) addressed by pins A_0 through A_{19} and $\overline{BHE}$. A_{20} through A_{23} are ignored.

Memory Addressing

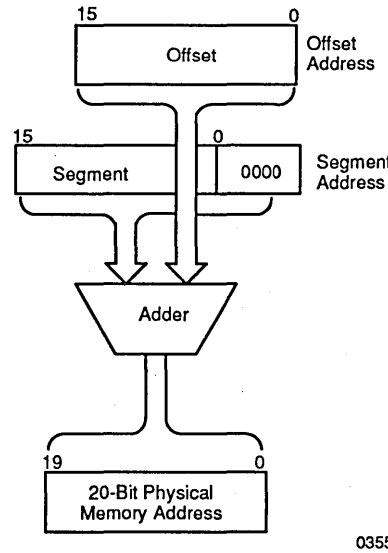
In real address mode the processor generates 20-bit physical addresses directly from a 20-bit segment base address and a 16-bit offset.

The selector portion of a pointer is interpreted as the upper 16 bits of a 20-bit segment address. The lower four bits of the 20-bit segment address are always zero. Segment addresses, therefore, begin on multiples of 16 bytes. See Figure 13 for a graphic representation of address formation.

All segments in real address mode are 64 kbytes in size and may be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (e.g., a word with its low order byte at offset FFFF(H) and its high order byte at offset 0000(H)). If, in real address mode, the information contained in a segment does not use the full 64 kbytes, the unused end of the segment may be overlaid by another segment to reduce physical memory requirements.

Reserved Memory Locations

The 80286 reserves two fixed areas of memory in real address mode (see Figure 1): system initialization area and interrupt table area. Locations from addresses FFFF0(H) through FFFFF(H) are reserved for system initialization. Initial execution begins at location FFFF0 (H). Locations 00000(H) through 003FF(H) are reserved for interrupt vectors.



03552-12

Figure 13. iAPX 86 Real Address Mode Address Calculation

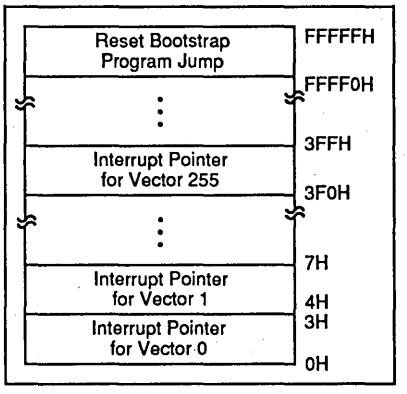


Figure 14. iAPX 86 Real Address Mode Initially Reserved Memory Locations

Table 8. Real Address Mode Addressing Interrupts

Function	Interrupt Number	Related Instructions	Return Address Before Instruction?
Interrupt table limit too small exception	8	INT vector is not within table limit	Yes
Processor extension segment overrun interrupt	9	ESC with memory operand extending beyond offset FFFF(H)	No
Segment overrun exception	13	Word memory reference with offset = FFFF(H) or an attempt to execute past the end of a segment	Yes

Interrupts

Table 8 shows the interrupt vectors reserved for exceptions and interrupts which indicate an addressing error. The exceptions leave the CPU in the state existing before attempting to execute the failing instruction (except for PUSH, POP, PUSHA, or POPA). Refer to the next section on protected mode initialization for a discussion on exception 8.

Protected Mode Initialization

To prepare the 80286 for protected mode, the LIDT instruction is used to load the 24-bit interrupt table base and 16-bit limit for the protected mode interrupt table. This instruction can also set a base and limit for the interrupt vector table in real address mode. After reset, the interrupt table base is initialized to 000000(H) and its size set to 03FF(H). These values are compatible with iAPX 86, 88 software. LIDT should only be executed in preparation for the protected mode.

Shutdown

Shutdown occurs when a severe error is detected that prevents further instruction processing by the CPU. Shutdown and halt are externally signaled via a halt bus operation. They can be distinguished by A_1 HIGH for halt and A_1 LOW for shutdown. In real address mode, shutdown can occur under two conditions:

- Exceptions 8 or 13 happen and the IDT limit does not include the interrupt vector.
- A CALL, INT, or POP instruction attempts to wrap around the stack segment when SP is not even.

An NMI input can bring the CPU out of shutdown if the IDT limit is at least 000F(H) and SP is greater than 0005(H); otherwise, shutdown can only be exited via the RESET input.

Protected Virtual Address Mode

The 80286 executes a fully upward-compatible superset of the 8086 instruction set in protected virtual address mode (protected mode). Protected mode also provides memory management and protection mechanisms and associated instructions.

The 80286 enters protected virtual address mode from real address mode by setting the PE (Protection Enable) bit of the machine status word with the Load Machine Status Word (LMSW) instruction. Protected mode offers extended physical and virtual memory address space,

memory protection mechanisms, and new operations to support operating systems and virtual memory.

All registers, instructions, and addressing modes described in the 80286 Base Architecture section remain the same. Programs for the iAPX 86, 88, 186, and real address mode 80286 can be run in protected mode; however, embedded constants for segment selectors are different.

Memory Size

The protected mode 80286 provides a 1 gigabyte virtual address space per task mapped into a 16-megabyte physical address space defined by the address pin A_{23}–A_0 and $\overline{BHE}$. The virtual address space may be larger than the physical address space since any use of an address that does not map to a physical memory location will cause a restartable exception.

Memory Addressing

As in real address mode, protected mode uses 32-bit pointers, consisting of 16-bit selector and offset components. The selector, however, specifies an index into a memory resident table rather than the upper 16 bits of a real memory address.

The 24-bit base address of the desired segment is obtained from the tables in memory. The 16-bit offset is added to the segment base address to form the physical address as shown in Figure 15. The tables are automatically referenced by the CPU whenever a segment register is loaded with a selector. All 80286 instructions which load a segment register will reference the memory-based tables without additional software. The memory-based tables contain 8-byte values called descriptors.

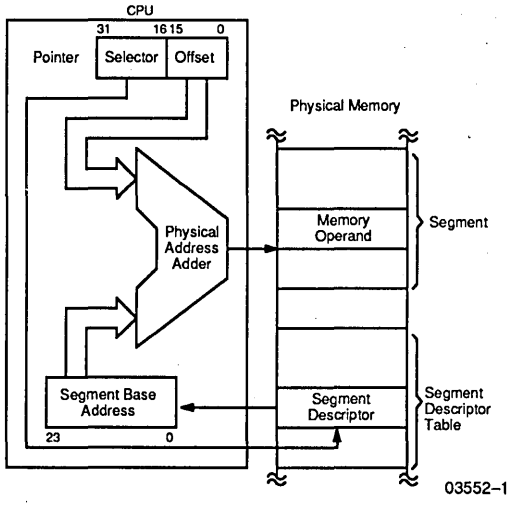


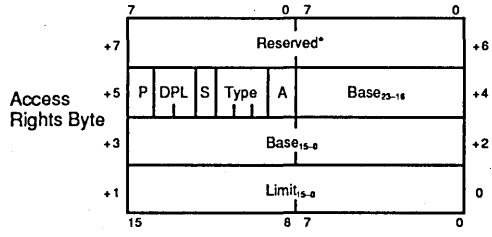
Figure 15. Protected Mode Memory Addressing

Descriptors

Descriptors define the use of memory. Special types of descriptors also define new functions for transfer of control and task switching. The 80286 has segment descriptors for code, stack and data segments, and system control descriptors for special system data segments and control transfer operations. Descriptor accesses are performed as locked bus operations to assure descriptor integrity in multi-processor systems.

Code and Data Segment Descriptors (S = 1)

Besides segment base addresses, code and data descriptors contain other segment attributes, including segment size (1 to 64 kbytes), access rights (read-only, read/write, execute-only, and execute/read), and presence in memory (for virtual memory systems) (see Figure 16). Any segment usage violating a segment attribute indicate by the segment descriptor will prevent the memory cycle and cause an exception or interrupt.



03552-15

*Must be set to 0 for compatibility with iAPX 386.

Code and data are stored in two types of segments: code segments and data segments. Both types are identified and defined by segment descriptors. Code segments are identified by the executable (E) bit set to 1 in the descriptor access rights byte. The access rights byte of both code and data segment descriptor types have three fields in common: present (P) bit, Descriptor Privilege Level (DPL), and accessed (A) bit. If P = 0, any attempted use of this segment will cause a not-present exception. DPL specifies the privilege level of the segment descriptor. DPL controls when the descriptor may be used by a task (refer to privilege discussion). The A bit shows whether the segment has been previously accessed for usage profiling, a necessity for virtual memory systems. The CPU will always set this bit when accessing the descriptor.

Data segments (S = 1, E = 0) may be either read-only or read-write as controlled by the W bit of the access rights byte. Read-only (W = 0) data segments may not be written into. Data segments may grow in two directions, as determined by the Expansion Direction (ED) bit: upwards (ED = 0) for data segments, and downwards (ED = 1) for a segment containing a stack. The limit field for a

Access Rights Byte Definition

	Bit Position	Name	Function		
	7	Present (P)	P = 1	Segment is mapped into physical memory.	
			P = 0	No mapping to physical memory exists; base and limit are not used. Segment privilege attribute used in privilege tests.	
	6–5	Descriptor Privilege Level (DPL)			
	4	Segment Descriptor (S)	S = 1	Code or Data segment descriptor	
			S = 0	Non-segment descriptor	
Type Field Definition	3	Executable (E)	E = 0	Data segment descriptor type is:	Data Segment
	2	Expansion Direction (ED)	ED = 0	Grow up segment, offsets must be ≤ limit.	
			ED = 1	Grow down segment, offsets must be > limit.	
	1	Writable (W)	W = 0	Data segment may not be written into.	
			W = 1	Data segment may be written into.	
	3	Executable (E)	E = 1	Code Segment Descriptor type is:	Code Segment
	2	Conforming (C)	C = 1	Code segment may only be executed when CPL ≥ DPL.	
	1	Readable (R)	R = 0	Code segment may not be read.	
			R = 1	Code segment may be read.	
	0	Accessed (A)	A = 0	Segment has not been accessed.	
			A = 1	Segment selector has been loaded into segment register or used by selector test instructions.	

Figure 16. Code and Data Segment Descriptors

data segment descriptor is interpreted differently depending on the ED bit (see Figure 16).

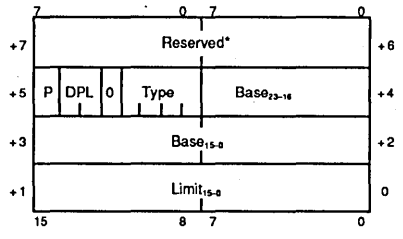
A code segment (S = 1, E = 1) may be execute-only or execute/read as determined by the Readable (R) bit. Code segments may never be written into and execute-only code segments (R = 0) may not be read. A code segment may also have an attribute called Conforming (C). A conforming code segment may be shared by programs that execute at different privilege levels. The DPL of a conforming code segment defines the range of privilege levels at which the segment may be executed (refer to privilege discussion).

System Segment Descriptors (S = 0, Type 1–3)

In addition to code and data segment descriptors, the protected mode 80286 defines system segment descriptors. These descriptors define special system data segments which contain a table of descriptors (Local Descriptor Table Descriptor) or segments which contain the execution state of a task (Task State Segment Descriptor).

Figure 17 gives the formats for the special system data segment descriptors. The descriptors contain a 24-bit base address of the segment and a 16-bit limit. The access byte defines the type of descriptor, its state and privilege level. The descriptor contents are valid and the segment is in physical memory if P = 1. If P = 0, the segment is not valid. The DPL field is only used in Task State Segment descriptors and indicates the privilege level at which the descriptor may be used (see Privilege). Since the Local Descriptor Table descriptor may only be used by a special privileged instruction, the DPL field is not used. Bit 4 of the access byte is 0 to indicate that it is a system control descriptor. The Type field specifies the descriptor type as indicated in Figure 17.

System Segment Descriptor



*Must be set to 0 for compatibility with iAPX 386.

03552-16

System Segment Descriptor Fields

Name	Value	Description
Type	1	Available Task State Segment
	2	Local Descriptor Table Descriptor
	3	Busy Task State Segment
P	0	Descriptor contents are not valid
	1	Descriptor contents are valid
DPL	0–3	Descriptor Privilege Level
Base	24-bit number	Base Address of special system data segment in real memory
Limit	16-bit number	Offset of last byte in segment

Figure 17. System Segment Format

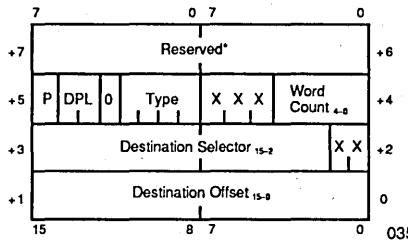
Gate Descriptors (S = 0, Type = 4–7)

Gates are used to control access to entry points within the target code segment. The gate descriptors are call gates, task gates, interrupt gates and trap gates. Gates provide a level of indirection between the source and destination of the control transfer. This indirection allows the CPU to automatically perform protection checks and control the entry point of the destination. Call gates are used to change privilege levels (see Privilege); task gates are used to perform a task switch; and interrupt and trap gates are used to specify interrupt service routines. The interrupt gate disables interrupts (resets IF) while the trap gates does not.

Figure 18 shows the format of the gate descriptors. The descriptor contains a destination pointer that points to the descriptor of the target segment and the entry point offset. The destination selector in an interrupt gate, trap gate, and call gate must refer to a code segment descriptor. These gate descriptors contain the entry point to prevent a program from constructing and using an illegal entry point. Task gates may only refer to a task state segment. Since task gates invoke a task switch, the destination offset is not used in the task gate.

Exception 13 is generated when the gate is used if a destination selector does not refer to the correct descriptor type. The Word Count field is used in the call gate descriptor to indicate the number of parameters (0–31 words) to be automatically copied from the caller's stack to the stack of the called routine when a control transfer changes privilege levels. The Word Count field is not used by any other gate descriptor.

Gate Descriptor



03552-17

*Must be set to 0 for compatibility with iAPX 386.

Gate Descriptor Fields

Name	Value	Description
Type	4	–Call Gate
	5	–Task Gate
	6	–Interrupt Gate
	7	–Trap Gate
P	0	–Descriptor Contents are not valid
	1	–Descriptor Contents are valid
DPL	0–3	Descriptor Privilege Level
Word Count	0–31	Number of words to copy from callers stack to called procedures stack. Only used with call gate.
Destination Selector	16-bit selector	Selector to the target code segment (Call, Interrupt or Trap Gate) Selector to the target task state segment (Task Gate)
Destination Offset	16-bit offset	Entry point within the target code segment

Figure 18. Gate Descriptor Format

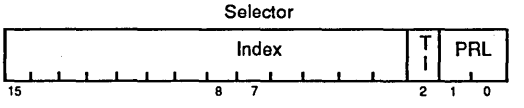
The access byte format is the same for all gate descriptors. P = 1 indicates that the gate contents are valid. P = 0 indicates the contents are not valid and causes exception 11 if referenced. DPL is the Descriptor Privilege Level and specifies when this descriptor may be used by a task (refer to privilege discussion). Bit 4 must equal 0 to indicate a system control descriptor. The Type field specifies the descriptor type as indicated in Figure 18.

Segment Descriptor Cache Registers

A segment descriptor cache register is assigned to each of the four segment registers (CS, SS, DS, ES). Segment descriptors are automatically loaded (cached) into a segment descriptor cache register (Figure 20) whenever the associated segment register is loaded with a selector. Only segment descriptors may be loaded into segment descriptor cache registers. Once loaded, all references to that segment of memory use the cached descriptor information instead of reaccessing memory. The descriptor cache registers are not visible to programs. No instructions exist to store their contents. They only change when a segment register is loaded.

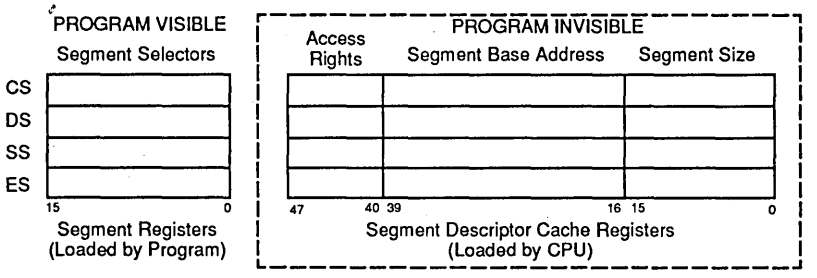
Selector Fields

A protected mode selector has three fields: descriptor entry index, local or global descriptor table indicator (TI), and selector privilege (RPL), as shown in Figure 19. These fields select one of two memory-based tables of descriptors, select the appropriate table entry, and allow high-speed testing of the selector's privilege attribute (refer to privilege discussion).



Bits	Name	Function
1–0	Requested Privilege Level (RPL)	Indicates Selector Privilege Level Desired
2	Table Indicator (TI)	TI = 0 Use Global Descriptor Table (GDT) TI = 1 Use Local Descriptor Table (LDT)
15–3	Index	Select Descriptor Entry in Table

Figure 19. Selector Fields



03552-18

Figure 20. Descriptor Cache Registers

Local and Global Descriptor Tables

Two tables of descriptors, called descriptor tables, contain all descriptors accessible by a task at any given time. A descriptor table is a linear array of up to 8192 descriptors. The upper 13 bits of the selector value are an index into a descriptor table. Each table has a 24-bit base register to locate the descriptor table in physical memory and a 16-bit limit register that confines descriptor access to the defined limits of the table as shown in Figure 21. A restartable exception (13) will occur if an attempt is made to reference a descriptor outside the table limits.

One table, called the Global Descriptor Table (GDT), contains descriptors available to all tasks. The other table, called the Local Descriptor Table (LDT), contains descriptors that can be private to a task. Each task may have its own private LDT. The GDT may contain all descriptor types except interrupt and trap descriptors. The LDT may contain only segment, task gate, and call gate descriptors. A segment cannot be accessed by a task if its segment descriptor does not exist in either descriptor table at the time of access.

The LGDT and LLDT instructions load the base and limit of the global and local descriptor tables. LGDT and LLDT are protected. They may only be executed by trusted programs operating at level 0. The LGDT instruction loads a six-byte field containing the 16-bit table limit and 24-bit base address of the Global Descriptor Table as shown in Figure 22. The LLDT instruction loads a selector which refers to a descriptor in the Local Descriptor Table. This descriptor contains the base address and limit for an LDT, as shown in Figure 17.

Interrupt Descriptor Table

The protected mode 80286 has a third descriptor table, called the Interrupt Descriptor Table (IDT) (see Figure 23), used to define up to 256 interrupts. It may contain only task gates, interrupt gates and trap gates. The IDT (Interrupt Descriptor Table) has a 24-bit base and 16-bit limit register in the CPU. The protected LIDT instruction loads these registers with a six-byte value of identical form to that of the LGDT instruction (see Figure 22 and Protected Mode Initialization).

References to IDT entries are made via INT instructions, external interrupt vectors, or exceptions. The IDT must be at least 256 bytes in size to allocate space for all reserved interrupts.

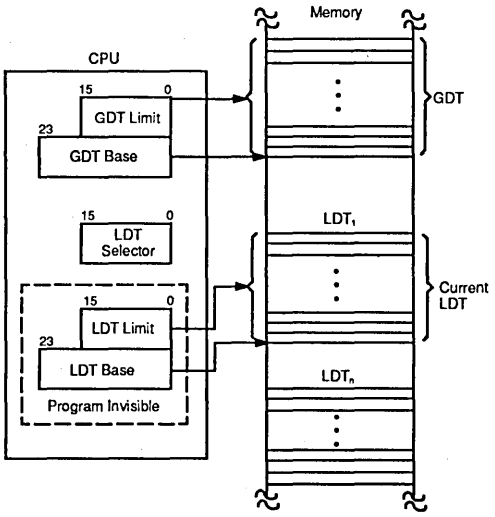
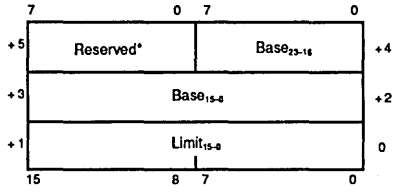


Figure 21. Local and Global Descriptor Table Definitions

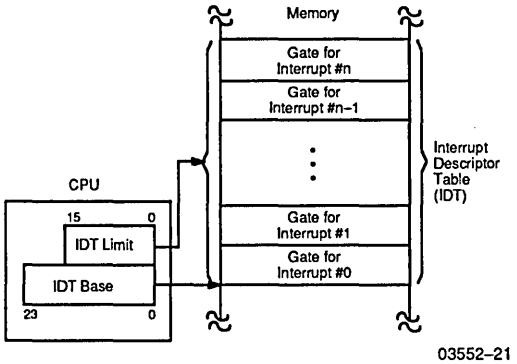
03552-19



03552–20

*Must be set to 0 for compatibility with iAPX 386.

Figure 22. Global Descriptor Table and Interrupt Descriptor Data Types



03552–21

Figure 23. Interrupt Descriptor Table Definition

Privilege

The 80286 has a four-level hierarchical privilege system which controls the use of privileged instructions and access to descriptors (and their associated segments) within a task. Four-level privilege, as shown in Figure 24, is an extension of the user/supervisor mode commonly found in minicomputers. The privilege levels are numbered 0 through 3. Level 0 is the most privileged level. Privilege levels provide protection *within* a task. (Tasks are isolated by providing private LDT's for each task.) Operating system routines, interrupt handlers, and other system software can be included and protected within the virtual address space of each task using the four levels of privilege. Tasks may also have a separate stack for each privilege level.

Tasks, descriptors, and selectors have a privilege level attribute that determines whether the descriptor may be used. Task privilege effects the use of instructions and descriptors. Descriptor and selector privilege only effect access to the descriptor.

Task Privilege

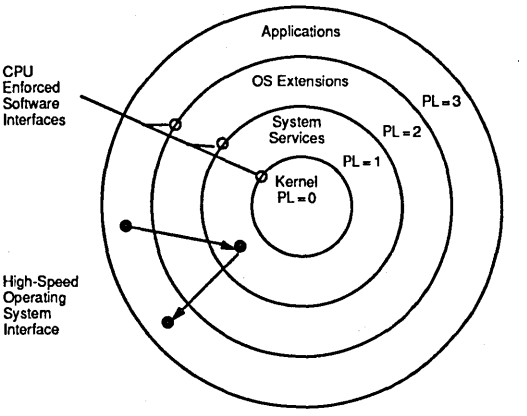
The task always executes at one of the four privilege levels. A task privilege level at any specific instant is called the Current Privilege Level (CPL) and is defined by the lower two bits of the CS register. CPL cannot change during execution in a single code segment. A task's CPL may only be changed by control transfers through gate descriptors to a new code segment (See Control Transfer). Tasks begin executing at the CPL value specified by the code segment when the task is initiated via a task switch operation. A task executing at Level 0 can access all data segments defined in the GDT and the task's LDT and is considered the most trusted level. A task executed at Level 3 has the most restricted access to data and is considered the least trusted level.

Descriptor Privilege

Descriptor privilege is specified by the Descriptor Privilege Level (DPL) field of the descriptor access byte. DPL specifies the least trusted privilege level (CPL) at which a task may access the descriptor. Descriptors with DPL = 0 are the most protected. Only tasks executing at privilege level 0 (CPL = 0) may access them. Descriptors with DPL = 3 are the least protected (i.e., have the least restricted access) since tasks can access them when CPL = 0, 1, 2, or 3. This rule applies to all descriptors, except LDT descriptors.

Selector Privilege

Selector privilege is specified by the Requested Privilege Level (RPL) field in the least significant two bits of a selector. Selector RPL may establish a less trusted privilege level than the current privilege level for the use of a selector. This level is called the task's effective privilege level (EPL). RPL can only reduce the scope of a task's access to data with this selector. A task's effective privilege is the numeric maximum of RPL and CPL. A selector with RPL = 0 imposes no additional restriction on its use while a selector with RPL = 3 can only refer to segments at privilege Level 3 regardless of the task's CPL. RPL is generally used to verify that pointer parameters passed to a more trusted procedure are not allowed to use data at a more privileged level than the caller (refer to pointer testing instructions).



03552–22

Figure 24. Hierarchical Privilege Levels

Descriptor Access and Privilege Validation

Determining the ability of a task to access a segment involves the type of segment to be accessed, the instruction used, the type of descriptor used and CPL, RPL, and DPL. The two basic types of segment accesses are control transfer (selectors loaded into CS) and data (selectors loaded into DS, ES, or SS).

Data Segment Access

Instructions that load selectors into DS and ES must refer to a data segment descriptor or readable code segment descriptor. The CPL of the task and the RPL of the selector must be the same as or more privileged (numerically equal to or lower than) than the descriptor DPL. In general, a task can only access data segments at the same or less privileged levels than the CPL or RPL (whichever is numerically higher) to prevent a program from accessing data it cannot be trusted to use.

An exception to the rule is a readable conforming code segment. This type of code segment can be read from any privilege level.

If the privilege checks fail (e.g., DPL is numerically less than the maximum of CPL and RPL) or an incorrect type of descriptor is referenced (e.g., gate descriptor or execute only code segment), exception 13 occurs. If the segment is not present, exception 11 is generated.

Instructions that load selectors into SS must refer to data segment descriptors for writable data segments. The descriptor privilege (DPL) and RPL must equal CPL. All other descriptor types or privilege level violation will cause exception 13. A not-present fault causes exception 12.

Control Transfer

Four types of control transfer can occur when a selector is loaded into CS by a control transfer operation (see Table 9). Each transfer type can only occur if the operation which loaded the selector references the correct descriptor type. Any violation of these descriptor usage rules (e.g., JMP through a call gate or RET to a Task State Segment) will cause exception 13.

The ability to reference a descriptor for control transfer is also subject to rules of privilege. A CALL or JUMP instruction may only reference a code segment descriptor with DPL equal to the task CPL or a conforming segment with DPL of equal or greater privilege than CPL. The RPL of the selector used to reference the code descriptor must have as much privilege as CPL.

RET and IRET instructions may only reference code segment descriptors with descriptor privilege equal to or less privileged than the task CPL. The selector loaded into CS is the return address from the stack. After the return, the selector RPL is the task's new CPL. If CPL changes, the old stack pointer is popped after the return address.

When a JMP or CALL references a Task State Segment descriptor, the descriptor DPL must be the same or less privileged than the task's CPL. Reference to a valid Task State Segment descriptor causes a task switch (see Task Switch Operation). Reference to a Task State Segment descriptor at a more privileged level than the task's CPL generates exception 13.

When an instruction or interrupt references a gate descriptor, the gate DPL must have the same or less privilege than the task CPL. If DPL is at a more privileged level than CPL, exception 13 occurs. If the destination

Table 9. Descriptor Types Used for Control Transfer

Control Transfer Types	Operation Types	Descriptor Referenced	Descriptor Table
Intersegment within the same privilege level	JMP, CALL, RET, IRET*	Code Segment	GDT/LDT
Intersegment to the same or higher privilege level	CALL	Call Gate	GDT/LDT
Interrupt within task may change CPL	Interrupt Instruction, Exception, External Interrupt	Trap or Interrupt Gate	IDT
Intersegment to a lower privilege level (changes task CPL)	RET, IRET*	Code Segment	GDT/LDT
Task Switch	CALL, JMP	Task State Segment	GDT
	CALL, JMP	Task Gate	GDT/LDT
	IRET**		
	Interrupt Instruction, Exception, External Interrupt	Task Gate	IDT

* NT (Nested Task bit of flag word) = 0
** NT (Nested Task bit of flag word) = 1

selector contained in the gate references a code segment descriptor, the code segment descriptor DPL must be the same or more privileged than the task CPL. If not, Exception 13 is issued. After the control transfer, the code segment descriptor DPL is the task's new CPL. If the destination selector in the gate references a task state segment, a task switch is automatically performed (see Task Switch Operation).

The privilege rules on control transfer require:

— JMP or CALL direct to a code segment (code segment descriptor) can only be to a conforming segment with DPL of equal or greater privilege than CPL or a non-conforming segment at the same privilege level.

— interrupts within the task or calls that may change privilege levels can only transfer control through a gate at the same or a less privileged level than CPL to a code segment at the same or more privileged level than CPL.

— return instructions that don't switch tasks can only return control to a code segment at the same or less privileged level.

— task switch can be performed by a call, a jump or an interrupt which references either a task gate or task state segment at the same or less privileged level.

Privilege Level Changes

Any control transfer that changes CPL within the task causes a change of stacks as part of the operation. Initial values of SS:SP for privilege levels 0, 1, and 2 are kept in the task state segment (refer to Task Switch Operation). During a JMP or CALL control transfer, the new stack pointer is loaded into the SS and SP registers and the previous stack pointer is pushed onto the new stack.

When returning to the original privilege level, its stack is restored as part of the RET or IRET instruction operation. For subroutine calls that pass parameters on the stack and cross privilege levels, a fixed number of words, as specified in the gate, are copied from the previous stack to the current stack. The intersegment RET instruction with a stack adjustment value will correctly restore the previous stack pointer upon return.

Protection

The 80286 includes mechanisms to protect critical instructions that affect the CPU execution state (e.g., HLT) and code or data segments from improper usage. These mechanisms are grouped under the term "protection" and have three forms:

- Restricted usage of segments (e.g., no write allowed to read-only data segments). The only segments available for use are defined by descriptors in the Local Descriptor Table (LDT) and Global Descriptor Table (GDT).
- Restricted access to segments via the rules of privilege and descriptor usage.
- Privileged instructions or operations that may only be executed at certain privilege levels as determined by the CPL and I/O Privilege Level (IOPL). The IOPL is defined by bits 14 and 13 of the flag word.

These checks are performed for all instructions and can be split into three categories: segment load checks (Table 10), operand reference checks (Table 11), and privileged instruction checks (Table 12). Any violation of the rules shown will result in an exception. A not-present exception related to the stack segment causes exception 12.

The IRET and POPF instructions do not perform some of their defined functions if CPL is not of sufficient privilege (numerically small enough). Precisely, these are:

- The IF bit is not changed if CPL > IOPL.
- The IOPL field of the flag word is not changed if CPL > 0.

No exceptions or other indication are given when these conditions occur.

Table 10. Segment Register Load Checks

Error Description	Exception Number
Descriptor table limit exceeded	13
Segment descriptor not present	11 or 12
Privilege rules violated	13
Invalid descriptor/segment type segment register load: — Read only data segment load to SS — Special control descriptor load to DS, ES, SS — Execute only segment load to DS, ES, SS — Data segment load to CS — Read/Execute code segment load to SS	13

Table 11. Operand Reference Checks

Error Description	Exception Number
Write into code segment	13
Read from execute-only code segment	13
Write to read-only data segment	13
Segment limit exceeded[1]	12 or 13

Note: Carry out in offset calculations is ignored.

Table 12. Privileged Instruction Checks

Error Description	Exception Number
CPL≠0 when executing the following instructions: LIDT, LLDT, LGDT, LTR, LMSW, CTS, HLT	13
CPL > IOPL when executing the following instructions: INS, IN, OUTS, OUT, STI, CLI, LOCK	13

Exceptions

The 80286 detects several types of exceptions and interrupts in protected mode (see Table 13). Most are restartable after the exceptional condition is removed. Interrupt handlers for most exceptions receive an error code, pushed on the stack after the return address, that identifies the selector involved (0 if none). The return address normally points to the failing instruction, including all leading prefixes. For a processor extension segment overrun exception, the return address will not point at the ESC instruction that caused the exception; however, the processor extension registers may contain the address of the failing instruction.

Special Operations

Task Switch Operation

The 80286 provides a built-in task switch operation which saves the entire 80286 execution state (registers, address space, and a link to the previous task), loads a new execution state, and commences execution in the new task. Like gates, the task switch operation is invoked by executing an inter-segment JMP or CALL instruction which refers to a Task State Segment (TSS) or task gate descriptor in the GDT or LDT. An INT n instruction, exception, or external interrupt may also invoke the task switch operation by selecting a task gate descriptor in the associated IDT descriptor entry.

The TSS descriptor points at a segment (see Figure 25) containing the entire 80286 execution state while a task gate descriptor contains a TSS selector. The limit field must be > 002B(H).

Each task must have a TSS associated with it. The current TSS is identified by a special register in the 80286 called the Task Register (TR). This register contains a selector referring to the task state segment descriptor that defines the current TSS. A hidden base and limit register associated with TR are loaded whenever TR is loaded with a new selector.

The IRET instruction is used to return control to the task that called the current task or was interrupted. Bit 14 in the flag register is called the Nested Task(NT) bit. It controls the function of the IRET instruction. If NT = 0, the IRET instruction performs the regular current task return; when NT = 1, IRET performs a task switch operation back to the previous task.

When a CALL or INT instruction initiates a task switch, the old and new TSS will be marked busy and the back link field of the new TSS set to the old TSS selector. The

Table 13. Protected Mode Exceptions

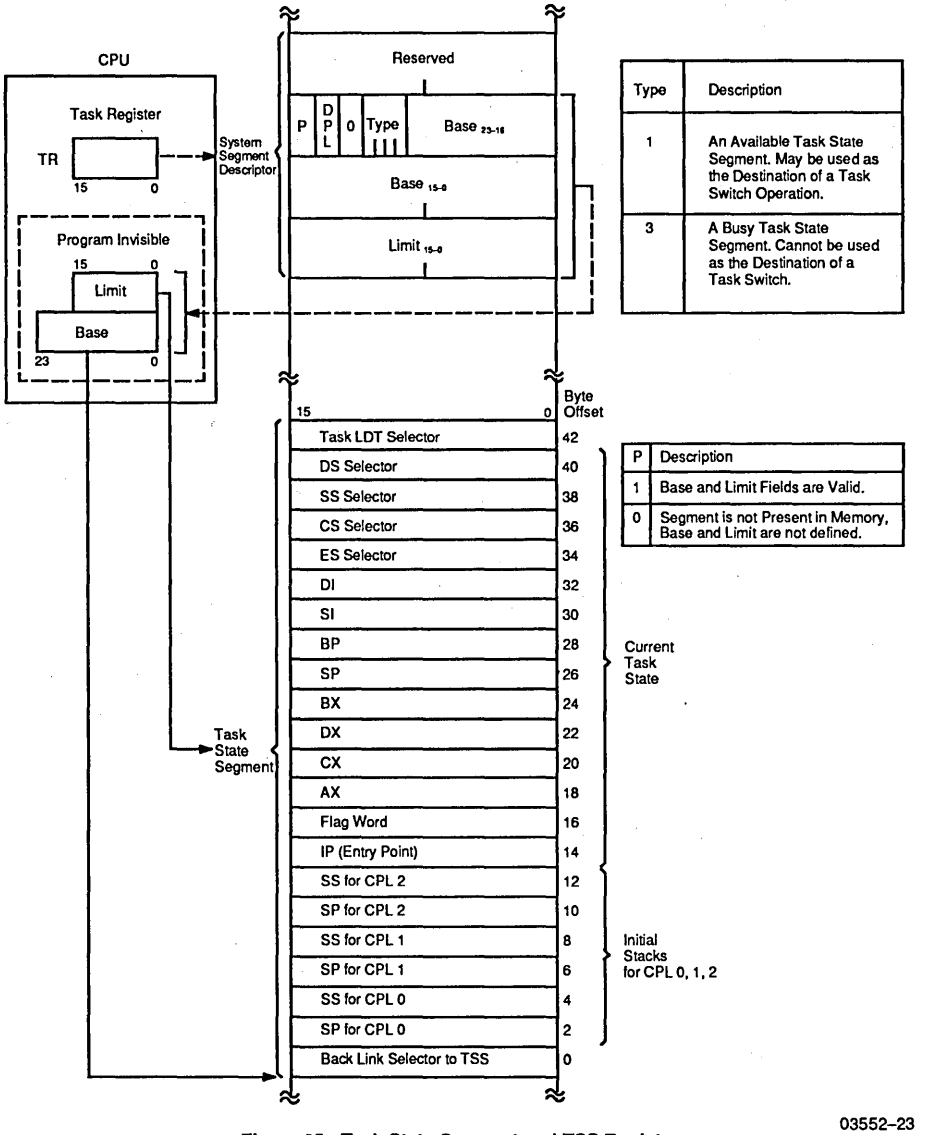
Interrupt Vector	Function	Return Address At Failing Instruction?	Always Restartable?	Error Code on Stack?
8	Double exception detected	Yes	No[2]	Yes
9	Processor extension segment overrun	No	No[2]	No
10	Invalid task state segment	Yes	Yes	Yes
11	Segment not present	Yes	Yes	Yes
12	Stack segment overrun or segment not present	Yes	Yes[1]	Yes
13	General protection	Yes	No[2]	Yes

Notes:

1. When a PUSHA or POPA instruction attempts to wrap around the stack segment, the machine state after the exception will not be restartable because stack segment wraparound is not permitted. This condition is identified by the value of the saved SP being either 0000(H), 0001(H), FFFE(H), or FFFF(H).
2. These exceptions indicate a violation to privilege rules or usage rules has occurred. Restart is generally not attempted under those conditions.
3. All these checks are performed for all instructions and can be split into three categories: Segment Load Checks (Table 10), Operand Reference Checks (Table 11), and Privileged Instruction Checks (Table 12). Any violation of the rules shown will result in an exception. A not-present exception causes exception 11 or 12 and is restartable.

NT bit of the new task is set by CALL or INT initiated task switches. An interrupt that does not cause a task switch will clear NT. NT may also be set or cleared by POPF or IRET instructions.

The task state segment is marked busy by changing the descriptor type field from Type 1 to Type 3. Use of a selector that references a busy task state segment causes Exception 13.



03552-23

Figure 25. Task State Segment and TSS Registers

Processor Extension Context Switching

The context of a processor extension is not changed by the task switch operation. A processor extension context need only be changed when a different task attempts to use the processor extension (which still contains the context of a previous task). The 80286 detects the first use of a processor extension after a task switch by causing the processor extension not present exception (7). The interrupt handler may then decide whether a context change is necessary.

Whenever the 80286 switches tasks, it sets the Task Switched (TS) bit of the MSW. TS indicates that a processor extension context may belong to a different task than the current one. The processor extension not present exception (7) will occur when attempting to execute an ESC or WAIT instruction if TS = 1 and a processor extension is present (MP = 1 in MSW).

Pointer Testing Instructions

The 80286 provides several instructions to speed pointer testing and consistency checks for maintaining system integrity (see Table 14). These instructions use the memory management hardware to verify that a selector value refers to an appropriate segment without risking an exception. A condition flag indicates whether use of the selector or segment will cause an exception.

Table 14. Pointer Test Instructions

Instruction	Operands	Function
ARPL	Selector, Register	Adjust Requested Privilege Level: adjusts the RPL of the selector to the numeric maximum of current selector RPL value and the RPL value in the register. Set zero flag if selector RPL was changed.
	Selector	
VERR	Selector	VERify for Read: sets the zero flag is the segment referred to by the selector can be read.
VERW	Register, Selector	VERify for Write: sets the zero flag if the segment referred to by the selector can be written.
LSL		Load Segment Limit: reads the segment limit into the register if privilege rules and descriptor type allow. Set zero flag if successful.
LAR	Register, Selector	Load Access Rights: reads the descriptor access rights byte into the register if privilege rules allow. Set zero flag if successful.

Double Fault and Shutdown

If two separate exceptions are detected during a single instruction execution, the 80286 performs the double fault exception (8). If an exception occurs during processing of the double fault exception, the 80286 will enter shutdown. During shutdown no further instructions or exceptions are processed. Either NMI (CPU remains in protected mode) or RESET (CPU exits protected mode) can force the 80286 out of shutdown. Shutdown is externally signalled via a HALT bus operation with A_1 HIGH.

Protected Mode Initialization

The 80286 initially executes in real address mode after RESET. To allow initialization code to be placed at the top of physical memory, A_{23-20} will be HIGH when the 80286 performs memory references relative to the CS register, until CS is changed. A_{23-20} will be zero for references to the DS, ES, or SS segments. Changing CS in real address mode will force A_{23-20} LOW whenever using CS thereafter. The initial CS:IP value of FFOO:FFFO provides 64K bytes of code space for initialization code without changing CS.

Before placing the 80286 into protected mode, several registers must be initialized. The GDT and IDT base registers must refer to a valid GDT and IDT. After executing the LMSW instruction to set PE, the 80286 must immediately execute an intrasegment JMP instruction to clear the instruction queue of instructions decoded in real address mode.

To force the 80286 CPU registers to match the initial protected mode state assumed by software, execute a JMP instruction with a selector referring to the initial TSS used in the system. This will load the task register, local descriptor table register, segment registers and initial general register state. The TR should point at a valid TSS since a task switch operation involves saving the current

System Interface

The 80286 system interface appears in two forms: a local bus and a system bus. The local bus consists of address, data, status, and control signals at the pins of the CPU. A system bus is any buffered version of the local bus. A system bus may also differ from the local bus in terms of coding of status and control lines and/or timing and loading of signals. The 80286 family includes several devices to generate standard system buses such as the IEEE 796 Standard MULTIBUS®.

Bus Interface Signals and Timing

The 80286 microsystem local bus interfaces the 80286 to local memory and I/O components. The interface has 24 address lines, 16 data lines, and 8 status and control signals.

The 80286 CPU, 82284 clock generator, 82C288 bus controller, 82289 bus arbiter, 8286/7 transceivers, and 8282/3 latches provide a buffered and decoded system bus interface. The 82284 generates the system clock and synchronizes $\overline{\text{READY}}$ and RESET. The 82C288 converts bus operation status encoded by the 80286 into command and bus control signals. These components

can provide the timing and electrical power drive levels required for most system bus interfaces including the MULTIBUS.

Physical Memory and I/O Interface

A maximum of 16 megabytes of physical memory can be addressed in protected mode. One megabyte can be addressed in real address mode. Memory is accessible as bytes or words. Words consist of any two consecutive bytes addressed with the least significant byte stored in the lowest address.

Byte transfers occur on either half of the 16-bit local data bus. Even bytes are accessed over D_{7-0} while odd bytes are transferred over D_{15-8}. Even-addressed words are transferred over D_{15-0} in one bus cycle, while odd-addressed words require *two* bus operations. The first transfers data on D_{15-8}, and the second transfers data on D_{7-0}. Both byte data transfers occur automatically, transparent to software.

Two bus signals, A_0 and $\overline{BHE}$, control transfers over the lower and upper halves of the data bus. Even address byte transfers are indicated by A_0 LOW and $\overline{BHE}$ HIGH. Odd address byte transfers are indicated by A_0 HIGH and $\overline{BHE}$ LOW. Both A_0 and $\overline{BHE}$ are LOW for even address word transfers.

The I/O address space contains 64K addresses in both modes. The I/O space is accessible as either bytes or words, as is memory. Byte-wide peripheral devices may be attached to either the upper or lower byte of the data bus. Byte-wide I/O devices attached to the upper data byte (D_{15-8}) are accessed with odd I/O addresses. Devices on the lower data byte are accessed with even I/O addresses. An interrupt controller such as the 8259A must be connected to the lower data byte (D_{7-0}) for proper return of the interrupt vector.

Bus Operation

The 80286 uses a double-frequency system clock (CLK input) to control bus timing. All signals on the local bus are measured relative to the system CLK input. The CPU divides the system clock by 2 to produce the internal processor clock, which determines bus state. Each processor clock is composed of two system clock cycles named phase 1 and phase 2. The 82284 clock generator output (PCLK) identifies the next phase of the processor clock. (See Figure 26.)

Six types of bus operations are supported: memory read, memory write, I/O read, I/O write, interrupt acknowledge, and halt/shutdown. Data can be transferred at a maximum rate of one word per two processor clock cycles.

The 80286 bus has three basic states: idle (T_i), send status (T_s), and perform command (T_c). The 80286 CPU also has a fourth local bus state called hold (T_h). T_h indicates that the 80286 has surrendered control of the local bus to another bus master in response to a HOLD request.

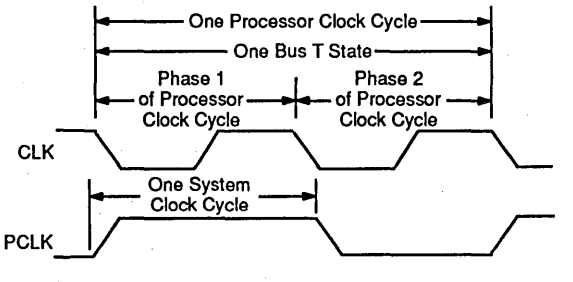
Each bus state is one processor clock long. Figure 27 shows the four 80286 local bus states and allowed transitions.

Bus States

The idle (T_i) state indicates that no data transfers are in progress or requested. The first active state, T_s, is signalled by either status line $\overline{S1}$ or $\overline{S0}$ going LOW also identifying phase 1 of the processor clock. During T_s, the command encoding, the address, and data (for a write operation) are available on the 80286 output pins. The 82C288 bus controller decodes the status signals and generates MULTIBUS-compatible read/write command and local transceiver control signals.

After T_s, the perform command (T_c) state is entered. Memory or I/O devices respond to the bus operation during T_c, either transferring read data to the CPU or accepting write data. T_c states may be repeated as often as necessary to assure sufficient time for the memory or I/O device to respond. The $\overline{READY}$ signal determines whether T_c is repeated. A repeated T_c state is called a wait state.

During hold (T_h), the 80286 will float all address, data, and status output pins, enabling another bus master to use the local bus. The 80286 HOLD input signal is used to place the 80286 into the T_h state. The 80286 HLDA output signal indicates that the CPU has entered T_h.



03552-24

Figure 26. System and Processor Clock Relationships

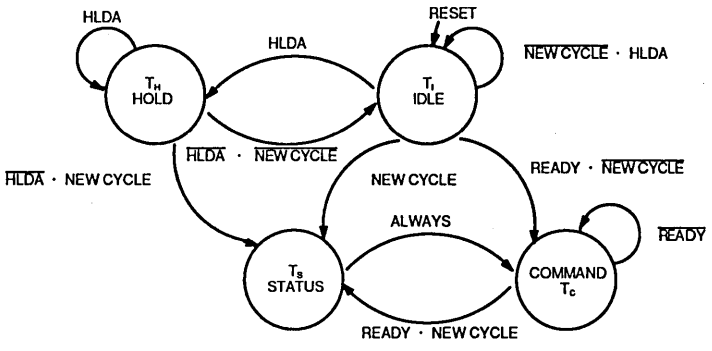


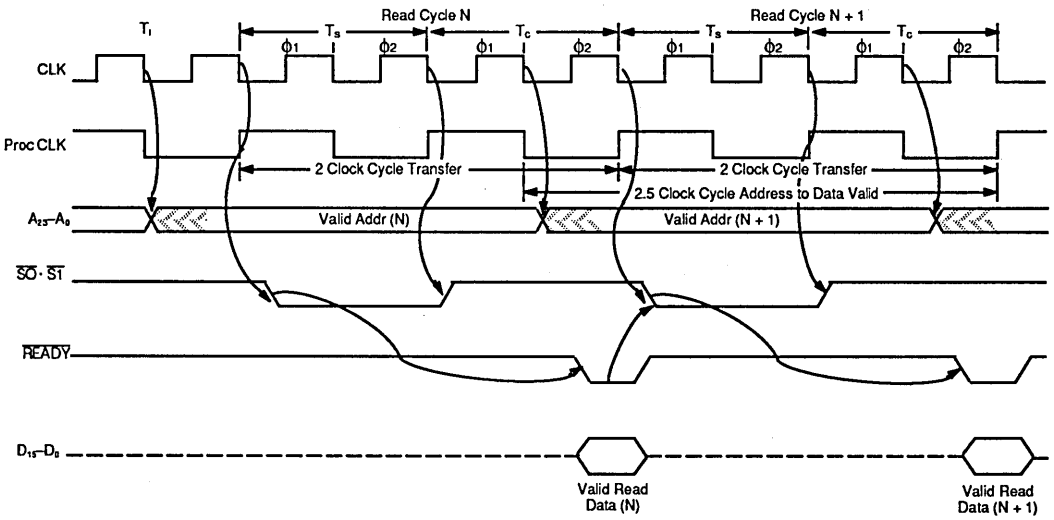
Figure 27. 80286 Bus States

Pipelined Addressing

The 80286 uses a local bus interface with pipelined timing to allow as much time as possible for data access. Pipelined timing allows bus operations to be performed in two processor cycles, while allowing each individual bus operation to last for three processor cycles.

The timing of the address outputs is pipelined such that the address of the next bus operation becomes available during the current bus operation. Or in other words, the first clock of the next bus operation is overlapped with the last clock of the current bus operation. Therefore, address decode and routing logic can operate in advance of the next bus operation. External address latches may hold the address stable for the entire bus operation and provide additional AC and DC buffering.

The 80286 does not maintain the address of the current bus operation during all Tc states. Instead, the address for the next bus operation may be emitted during phase 2 of any Tc. The address remains valid during phase 1 of the first Tc to guarantee hold time, relative to ALE, for the address latch inputs.



Pipelining: valid address (N+1) available in last phase of bus cycle (N).

03552-26

Figure 28. Basic Bus Cycle

Bus Control Signals

The 82C288 bus controller provides control signals: address latch enable (ALE), Read/Write commands, data transmit/receive (DT/$\overline{R}$)< and data enable (DEN) that control the address latches, data transceivers, write enable, and output enable for memory and I/O systems.

The Address Latch Enable (ALE) output determines when the address may be latched. ALE provides at least one system CLK period of address hold time from the end of the previous bus operation until the address for the next bus operation appears at the latch outputs. This adress hold time is required to support MULTIBUS and common memory systems.

The data bus transceivers are controlled by 82C288 outputs Data Enable (DEN) and Data Transmit/Receive (DT/$\overline{R}$). DEN enables the data transceivers while DT/$\overline{R}$ controls transceiver direction. DEN and DT/$\overline{R}$ are timed to prevent bus contention between the bus master, data bus transceivers, and system data bus transceivers.

Command Timing Controls

Two system timing customization options, command extension and command delay, are provided on the 80286 local bus.

Command extension allows additional time for external devices to respond to a command and is analogous to inserting wait states on the 8086. External logic can control the duration of any bus operation such that the operation is only as long as necessary. The $\overline{READY}$ input signal can extend any bus operation for as long as necessary.

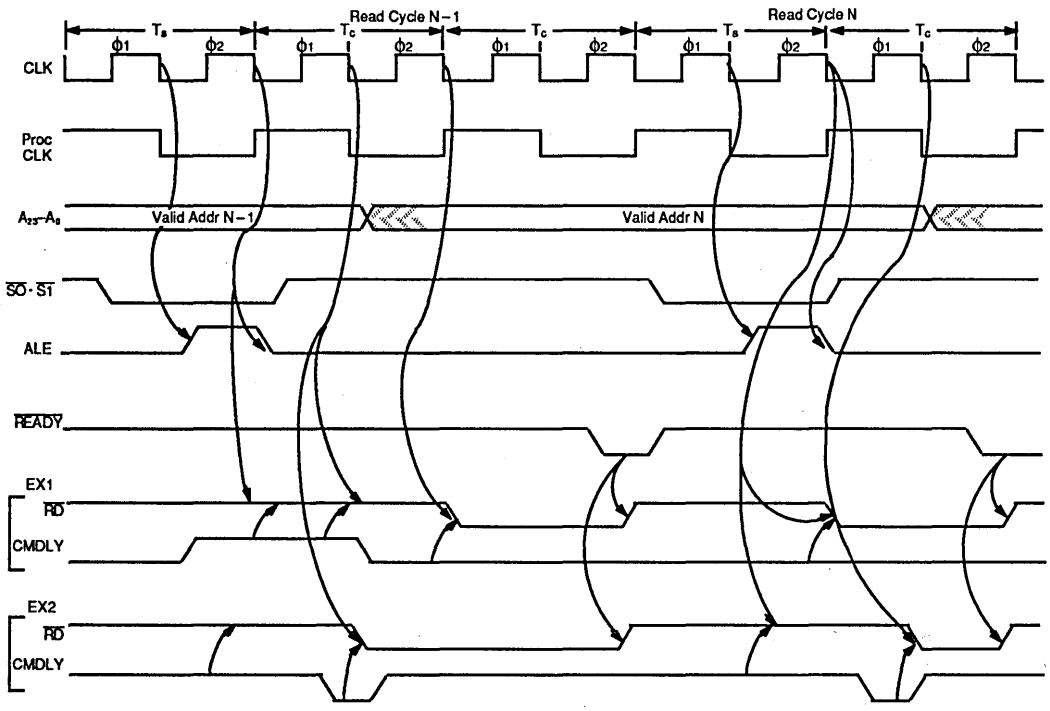
Command delay allows an increase of address or write data set-up time to system bus command active for any bus operation by delaying when the system bus command becomes active. Command delay is controlled by the 82C288 CMDLY input. After Ts, the bus controller samples CMDLY at each failing edge of CLK. If CMDLY is HIGH, the 82C288 will not activate the command signal. When CMDLY is LOW, the 82C288 will activate the command signal. After the command becomes active, the CMDLY input is not sampled.

When a command is delayed, the available response time from command active to return read data or accept write data is less. To customize system bus timing, an address decoder can determine which bus operations require delaying the command. The CMDLY input does not affect the timing of ALE, DEN, or DT/$\overline{R}$.

Figure 29 illustrates four uses of CMDLY. Example 1 shows delaying the read command two system CLKs for cycle N−1 and no delay for cycle N, and example 2 shows delaying the read command one system CLK for cycle N−1 and one system CLK delay for cycle N.

Bus Cycle Termination

At maximum transfer rates, the 80286 bus alternates between the status and command states. The bus status signals become inactive after Ts so that they may correctly signal the start of the next bus operation after the completion of the current cycle. No external indication of Tc exists on the 80286 local bus. The bus master and bus controller enter Tc directly after Ts and continue executing Tc cycles until terminated by $\overline{READY}$.



03552-27

Figure 29. CMDLY Controls and Leading Edge of the Command

READY Operation

The current bus master and 82C288 bus controller terminate each bus operation simultaneously to achieve maximum bus bandwidth. Both are informed in advance by READY active which identifies the last Tc cycle of the current bus operation. The bus master and bus controller must see the same sense of the READY signal, thereby requiring READY be synchronous to the system clock.

Synchronous Ready

The 82284 clock generator provides READY synchronization from both synchronous and asynchronous sources (See Figure 30). The synchronous ready input (SRDY) of the clock generator is sampled with the falling edge of CLK at the end of phase 1 of each Tc. The state of SRDY is then broadcast to the bus master and bus controller via the READY output line.

Asynchronous Ready

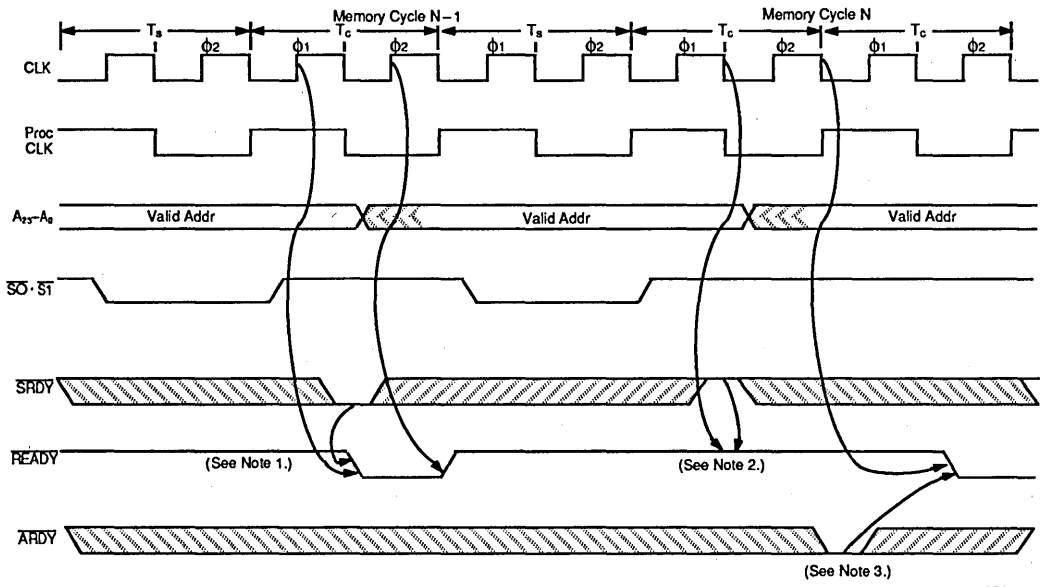
Many systems have devices or subsystems that are asynchronous to the system clock. As a result, their ready outputs cannot be guaranteed to meet the 82284 SRDY set-up and hold time requirements. The 82284 asynchronous ready input (ARDY) is designed to accept such signals. The ARDY input is sampled at the beginning of each Tc cycle by 82284 synchronization logic. This provides a system CLK cycle time to resolve its value before broadcasting it to the bus master and bus controller.

ARDY or ARDYEN must be HIGH at the end of Ts. ARDY cannot be used to terminate bus cycle with no wait status.

Each ready input of the 82284 has an enable pin (SRDYEN and ARDYEN) to select whether the current bus operation will be terminated by the synchronous or asynchronous ready. Either of the ready inputs may terminate a bus operation. These enable inputs are active low and have the same timing as their respective ready inputs. Address decode logic usually selects whether the current bus operation should be terminated by ARDY or SRDY.

Data Bus Control

Figures 31, 32, and 33 show how the DT/R, DEN, data bus, and address signals operate for different combinations of read, write, and idle bus operations. DT/R goes active (LOW) for a read operation. DT/R remains HIGH before, during, and between write operations.



Notes: 1. $\overline{\text{SRDYEN}}$ is active LOW.
2. If $\overline{\text{SRDYEN}}$ is HIGH, the state of $\overline{\text{SRDY}}$ will not affect $\overline{\text{READY}}$.
3. $\overline{\text{ARDYEN}}$ is active LOW.

Figure 30. Synchronous and Asynchronous Ready

The data bus is driven with write data during the second phase of Ts. The delay in write data timing allows the read data drivers, from a previous read cycle, sufficient time to enter three-state OFF before the 80286 CPU begins driving the local data bus for write operations. Write data will always remain valid for one system clock past the last Tc to provide sufficient hold time for MULTIBUS or other similar memory or I/O systems. During write-read or write-idle sequences, the data bus enters three-state OFF during the second phase of the processor cycle after the last Tc. In a write-write sequence the data bus does not enter three-state OFF between Tc and Ts.

Bus Usage

The 80286 local bus may be used for several functions: instruction data transfers, data transfers by other bus masters, instruction fetching, processor extension data transfers, interrupt acknowledge, and halt/shutdown. This section describes local bus activities which have special signals or requirements.

HOLD and HLDA

HOLD and HLDA allow another bus master to gain control of the local bus by placing the 80286 bus into the Th state. The sequence of events required to pass control between the 80286 and another local bus master are shown in Figure 34.

In this example, the 80286 is initially in the Th, state as signaled by HLDA being active. Upon leaving Th, as signaled by HLDA going inactive, a write operation is started. During the write operation another local bus master requests the local bus from the 80286 as shown by the HOLD signal. After completing the write operation, the 80286 performs one Ti bus cycle, to guarantee write data hold time, then enters Th as signaled by HLDA going active.

The CMDLY signal and $\overline{\text{ARDY}}$ ready are used to start and stop the write bus command, respectively. Note that $\overline{\text{SRDY}}$ must be inactive or disabled by $\overline{\text{SRDYEN}}$ to guarantee $\overline{\text{ARDY}}$ will terminate the cycle.

HOLD must not be active during the time from the leading edge of RESET until 34 CLKs following the trailing edge of RESET unless the 80286 is in the Halt condition. To ensure that the 80286 remains in the Halt condition until the processor Reset operation is complete, no interrupts should occur after the execution of HLT until 34 CLKs after the trailing edge of the RESET pulse.

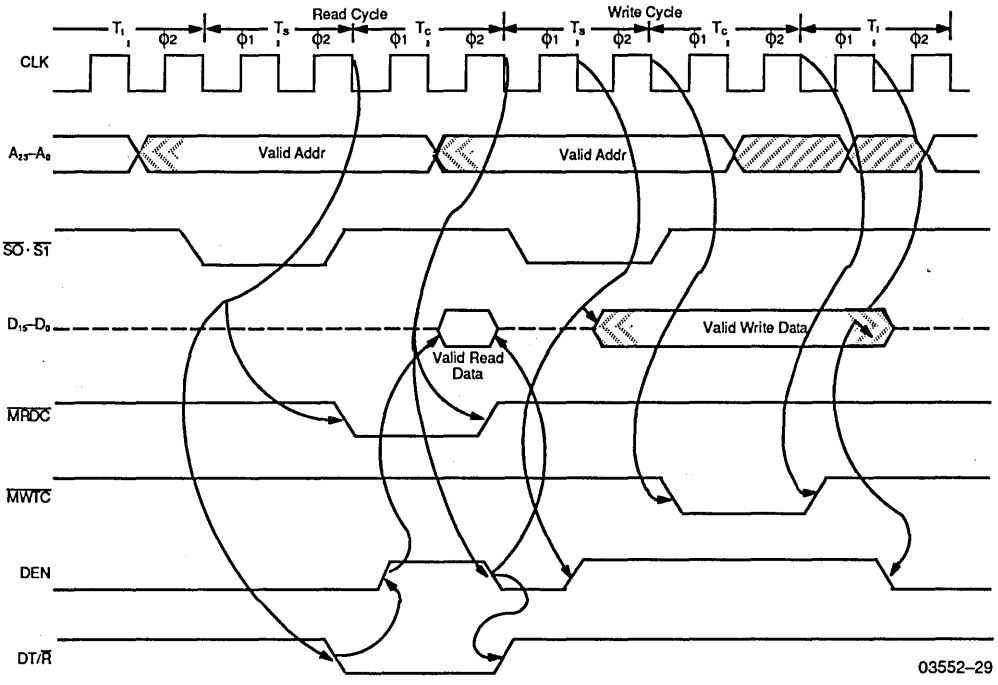


Figure 31. Back-to-Back Read-Write Cycles

Lock

The CPU asserts an active lock signal during Interrupt-Acknowledge cycles, the XCHG instruction, and during some descriptor accesses. Lock is also asserted when the LOCK prefix is used. The LOCK prefix may be used with the following ASM-286 assembly instructions; MOVS, INS, and OUTS. For bus cycles other than Interrupt-Acknowledge cycles, Lock will be active for the first and subsequent cycles of a series of cycles to be locked. Lock will not be shown active during the last cycle to be locked. For the next-to-last cycle, Lock will become inactive at the end of the first Tc regardless of the number of wait-states inserted. For Interrupt-Acknowledge cycles, Lock will be active for each cycle, and will become inactive at the end of the first Tc for each cycle regardless of the number of wait-states inserted.

Instruction Fetching

The 80286 Bus Unit (BU) will fetch instructions ahead of the current instruction being executed. This activity is called prefetching. It occurs when the local bus would otherwise be idle and obeys the following rules:

A prefetch bus operation starts when at least two bytes of the 6-byte prefetch queue are empty.

The prefetcher normally performs word prefetches independent of the byte alignment of the code segment base in physical memory.

The prefetcher will perform only a byte code fetch operation for control transfers to an instruction beginning on a numerically odd physical address.

Prefetching stops whenever a control transfer or HLT instruction is decoded by the IU and placed into the instruction queue.

In real address mode, the prefetcher may fetch up to 5 bytes beyond the last control transfer or HLT instruction in a code segment.

In protected mode, the prefetcher will never cause a segment overrun exception. The prefetcher stops at the last physical memory word of the code segment. Exception 13 will occur if the program attempts to execute beyond the last full instruction in the code segment.

If the last byte of a code segment appears on an even physical memory address, the prefetcher will read the next physical byte of memory (perform a word code fetch). The value of this byte is ignored and any attempt to execute it causes exception 13.

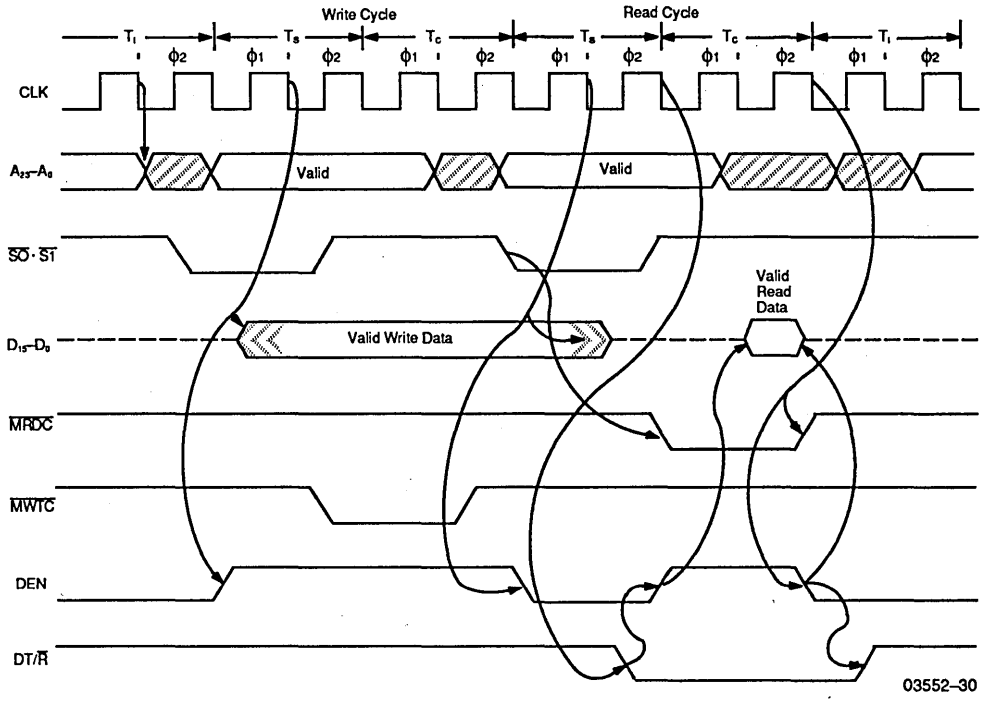


Figure 32. Back-to-Back Write-Read Cycles

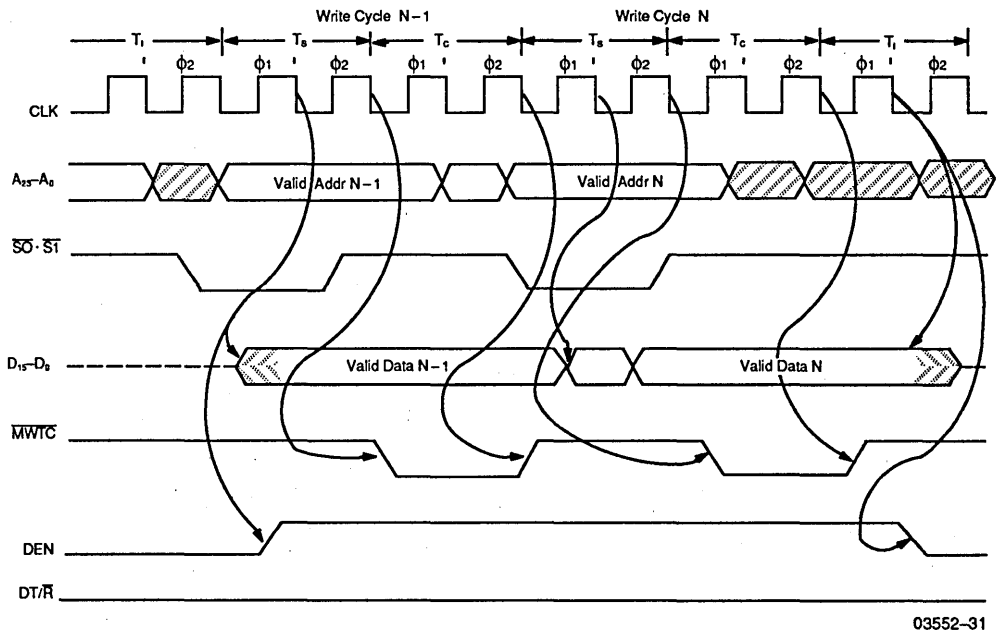
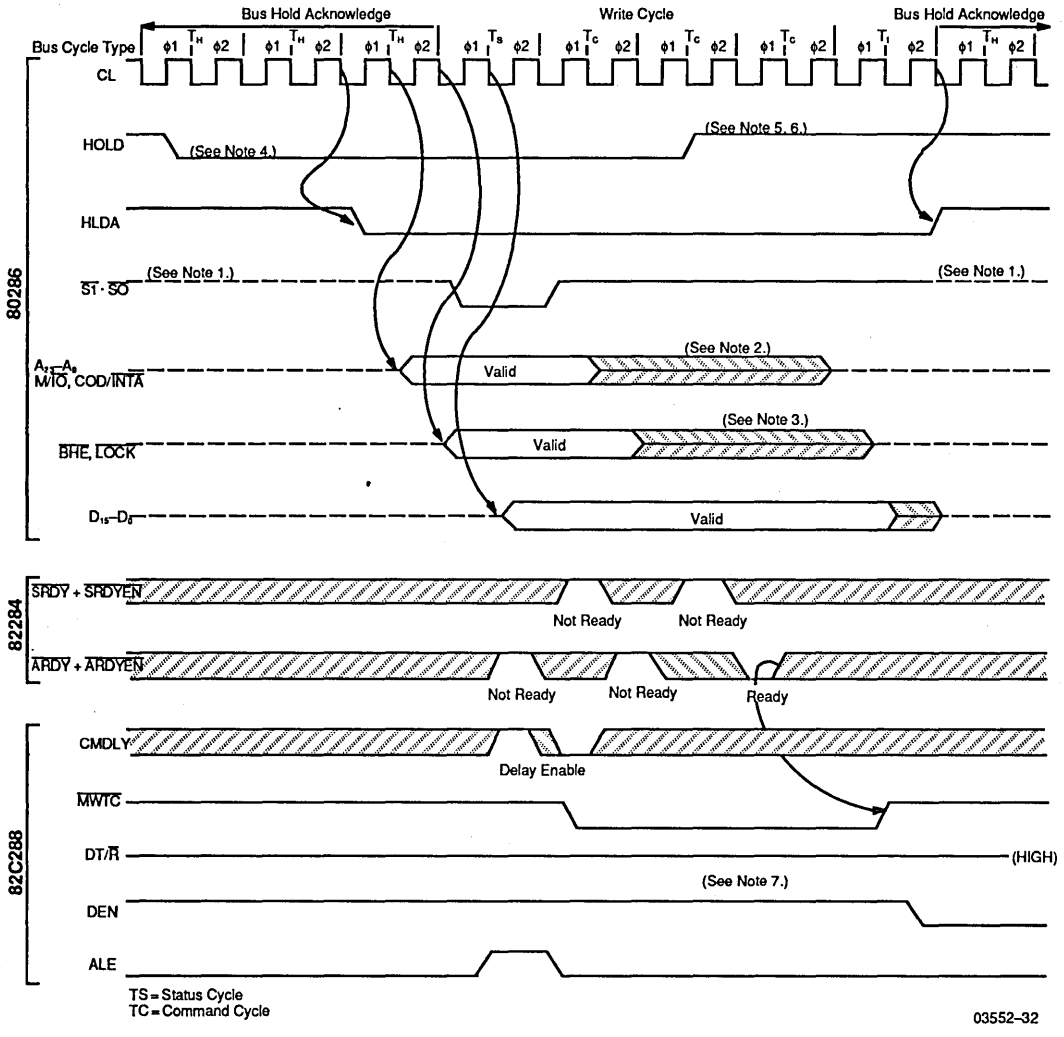


Figure 33. Back-to-Back Write-Write Cycles



Notes: 1. Status lines are not driven by 80286, yet remain high due to pull-up resistors in 82C288 and 82289 during HOLD state.

2. Address, M/$\overline{IO}$ and COD/$\overline{INTA}$ may start may start floating during any TC, depending on when internal 80286 bus arbiter decides to release bus to external HOLD. The float starts in $\phi 2$ of TC.
3. $\overline{BHE}$ and $\overline{LOCK}$ may start floating after the end of any TC, depending on when internal 80286 bus arbiter decides to release bus to external HOLD.
4. The minimum HOLD $\downarrow$ to HLDA $\downarrow$ time is shown. Maximum is one T_H longer.
5. The earliest HOLD $\uparrow$ time is shown which will always allow a subsequent memory cycle if pending.
6. The minimum HOLD $\uparrow$ to HLDA $\uparrow$ time is shown. Maximum is a function of the instruction, type of bus cycle and other machine status (i.e., Interrupts, Waits, Lock, etc.).
7. Asynchronous ready allows termination of the cycle. Synchronous ready does not signal ready in this example. Synchronous ready state is ignored after ready is signaled via the asynchronous input.

Figure 34. MULTIBUS Write Terminated by Asynchronous Ready with Bus Hold

Processor Extension Transfers

The processor extension interface uses I/O port addresses 00F8(H), and 00FA(H), and 00FC(H) which are part of the I/O port address range and is a reserved area. An ESC instruction with EM = 0 and TS = 0 will perform I/O bus operations to one or more of these I/O port addresses independent of the value of IOPL and CPL.

ESC instructions with memory references enable the CPU to accept PEREQ inputs for processor extension operand transfers. The CPU will determine the operand starting address and read/write status of the instruction. For each operand transfer, two or three bus operations, one word transfer with I/O port address 00FA(H), and one or two bus operations with memory are performed. Three bus operations are required for each word operand aligned on an odd byte address.

Interrupt Acknowledge Sequence

Figure 35 illustrates an interrupt acknowledge sequence performed by the 80286 in response to an INTR input. An interrupt acknowledge sequence consists of two INTA bus operations. The first allows a master 8259A Programmable Interrupt Controller (PIC) to determine which, if any, of its slaves should return the interrupt vector. An eight-bit vector is read by the 80286 during the second INTA bus operation to select an interrupt handler routine from the interrupt table.

The Master Cascade Enable (MCE) signal of the 82C288 is used to enable the cascade address drivers, during INTA bus operations (see Figure 35), onto the local address bus for distribution to slave interrupt controllers via the system address bus. The 80286 emits the $\overline{LOCK}$ signal (active LOW) during T_S of the first INTA bus operation. A local bus "hold" request will not be honored until the end of the second INTA bus operation.

Three idle processor clocks are provided by the 80286 between INTA bus operations to allow for the minimum INTA to INTA time and CAS (cascade address) out delay of the 8259A. The second INTA bus operation must always have at least one extra T_C state added via logic controlling $\overline{READY}$. A_{23}–A_0 are in three-state OFF until after the first T_C state of the second INTA bus operation. This prevents bus contention between the cascade address drivers and CPU address drivers. The extra T_C state allows time for the 80286 to resume driving the address lines for subsequent bus operations.

Local Bus Usage Priorities

The 80286 local bus is shared among several internal units and external HOLD requests. In case of simultaneous requests, their relative priorities are:

(Highest) Any transfers which assert $\overline{LOCK}$ either explicitly (via the LOCK instruction prefix) or implicitly (i.e., segment descriptor access, interrupt acknowledge sequence, or an XCHG with memory).

The second of the two-byte bus operations required for an odd aligned word operand.

Local bus request via HOLD input.

Processor extension data operand transfer via PEREQ input.

Data transfer performed by EU as part of an instruction.

(Lowest) An instruction prefetch request from BU. The EU will inhibit prefetching two processor clocks in advance of any data transfers to minimize waiting by EU for a prefetch to finish.

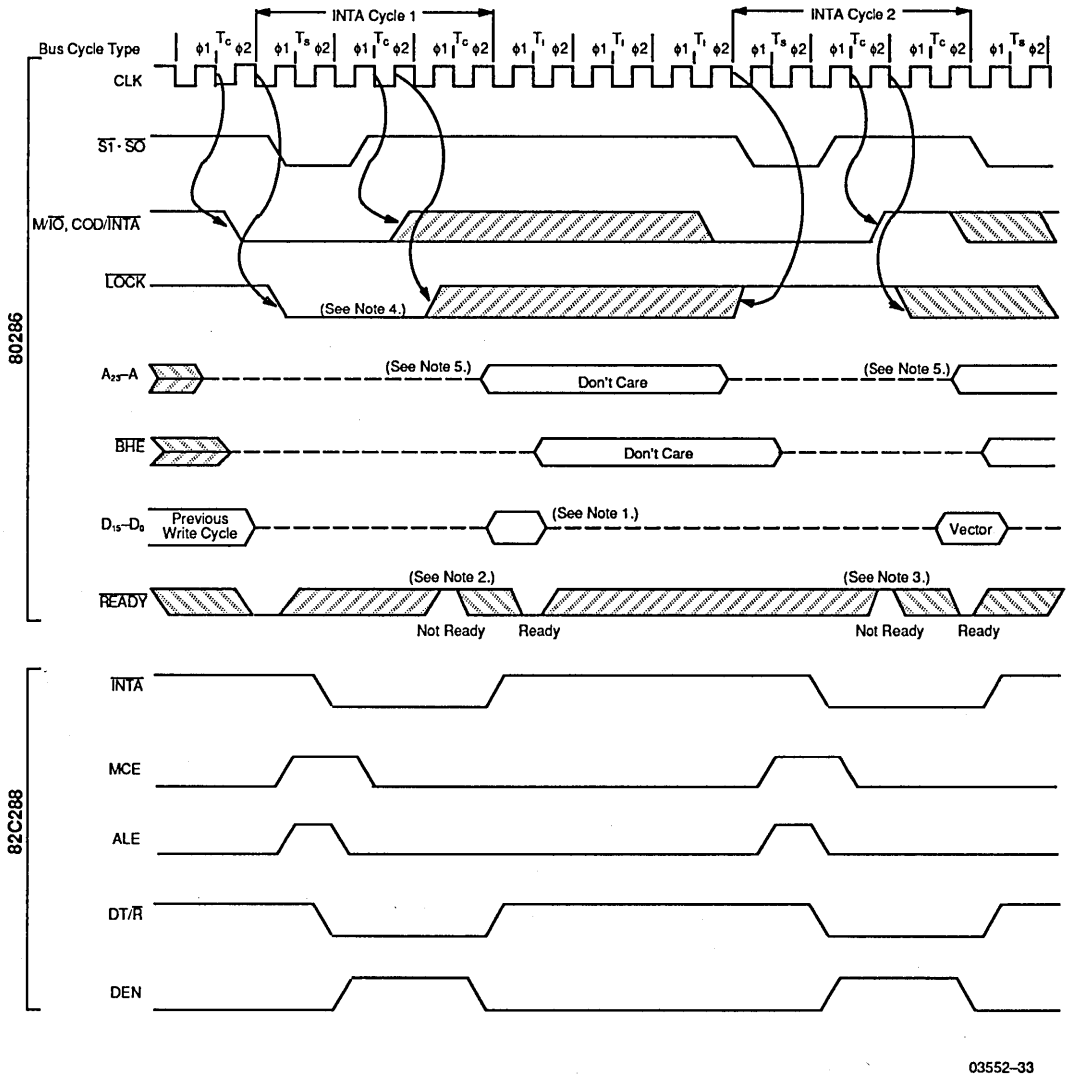
Halt or Shutdown Cycles

The 80286 externally indicates halt or shutdown conditions as a bus operation. These conditions occur due to a HLT instruction or multiple protection exceptions while attempting to execute one instruction. A halt or shutdown bus operation is signalled when $\overline{S1}$, $\overline{S0}$ and COD/$\overline{INTA}$ are LOW and M/$\overline{IO}$ is HIGH. A_1 HIGH indicates halt, and A_1 LOW indicates shutdown. The 82C288 bus controller does not issue ALE, nor is $\overline{READY}$ required to terminate a halt or shutdown bus operation.

During halt or shutdown, the 80286 may service PEREQ or HOLD requests. A processor extension segment overrun exception during shutdown will inhibit further service of PEREQ. Either NMI or RESET will force the 80286 out of either halt or shutdown. An INTR, if interrupts are enabled, or a processor extension segment overrun exception will also force the 80286 out of halt.

System Configurations

The versatile bus structure of the 80286 microsystem, with a full complement of support chips, allows flexible configuration of a wide range of systems. The basic configuration, shown in Figure 36, is similar to an iAPX 86 maximum mode system. It includes the CPU plus an 8259A interrupt controller, 82284 clock generator, and the 82C288 Bus Controller. The iAPX 86 latches (29843 and 29845) and transceivers (29833 and 29863) may be used in an 80286 microsystem.



03552-33

Notes:

1. Data is ignored.
2. First INTA cycle should have at least one wait state inserted to meet 8259A minimum INTA pulse width.
3. Second INTA cycle must have at least one wait state inserted since the CPU will not drive A_{23}–A_0, $\overline{BHE}$, and $\overline{LOCK}$ until after the first T_c state.
   The CPU imposed one/clock delay prevents bus contention between cascade address buffer being disabled by MCE ↓ and address outputs.
   Without the wait state, the 80286 address will not be valid for a memory cycle started immediately after the second INTA cycle. The 8259A also requires one wait state for minimum INTA pulse width.
4. $\overline{LOCK}$ is active for the first INTA cycle to prevent the 82289 from releasing the bus between INTA cycles in a multi-master system.
5. A_{23}–A_0 exits three-state OFF during $\phi 2$ of the second T_c in the INTA cycle.

Figure 35. Interrupt Acknowledge Sequence

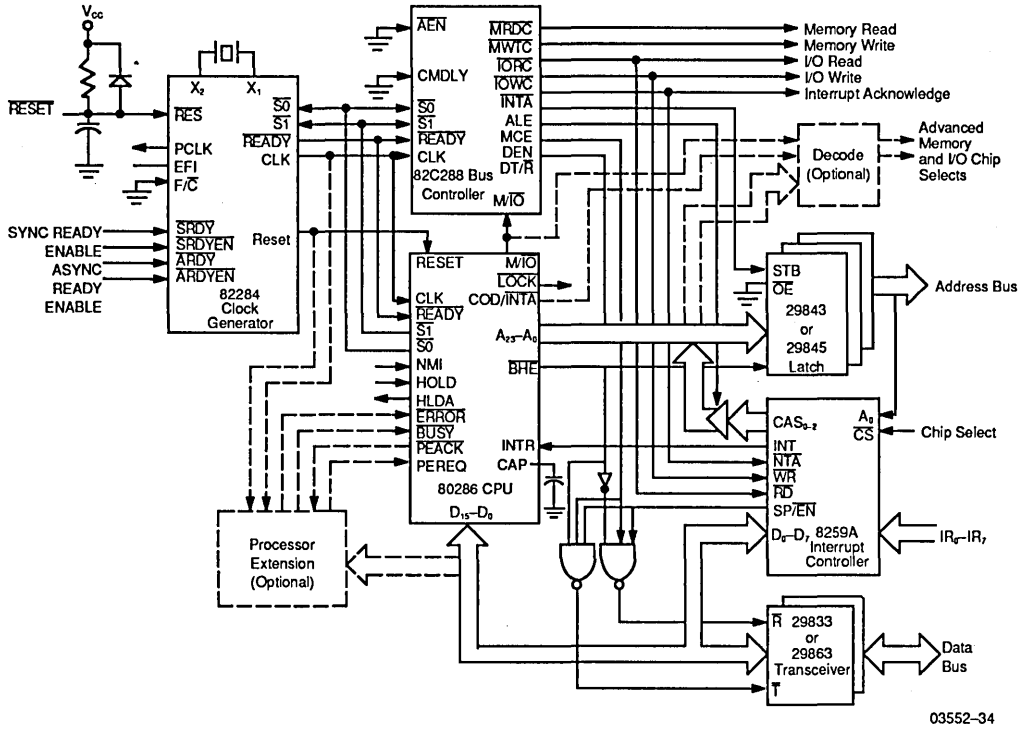


Figure 36. Basic 80286 System Configuration

As indicated by the dashed lines in Figure 36, the ability to add processor extensions is an integral feature of 80286 microsystems. The processor extension interface allows external hardware to perform special functions and transfer data concurrent with CPU execution of other instructions. Full system integrity is maintained because the 80286 supervises all data transfers and instruction execution for the processor extension.

The 80286 with the 80287 numeric processor extension (NPX) uses this interface. The iAPX 286/287 has all the instructions and data types of an iAPX 86/87 or iAPX 88/87. The 80287 NPX can perform numeric calculations and data transfers concurrently with CPU program execution. Numerics code and data have the same integrity as all other information protected by the 80286 protection mechanism.

The 80286 can overlap chip select decoding and address propagation during the data transfer for the previous bus operation. This information is latched into the 29843/45's by ALE during the middle of a Ts cycle. The latched chip select and address information remains stable during the bus operation while the next cycle's address is being decoded and propagated into the system. Decode logic can be implemented with a high-speed bipolar PROM.

The optional decode logic shown in Figure 36 takes advantage of the overlap between address and data of the 80286 bus cycle to generate advanced memory and I/O-select signals. This minimizes system performance degradation caused by address propagation and decode delays. In addition to selecting memory and I/O, the advanced selects may be used with configurations supporting local and system buses to enable the appropriate bus interface for each bus cycle. The COD/$\overline{\text{INTA}}$ and M/$\overline{\text{IO}}$ signals are applied to the decode logic to distinguish between interrupt, I/O, code and data bus cycles.

By adding the 82289 bus arbiter chip, the 80286 provides a MULTIBUS system bus interface as shown in Figure 37. The ALE output of the 82C288 for the MULTIBUS bus is connected to its CMDLY input to delay the start of commands one system CLK as required to meet MULTIBUS address and write data set-up times. This arrangement will add at least one extra Tc state to each bus operation which uses the MULTIBUS.

A second 82C288 bus controller and additional latches and transceivers could be added to the local bus of Figure 37. This configuration allows the 80286 to support an on-board bus for local memory and peripherals and the MULTIBUS for system bus interfacing.

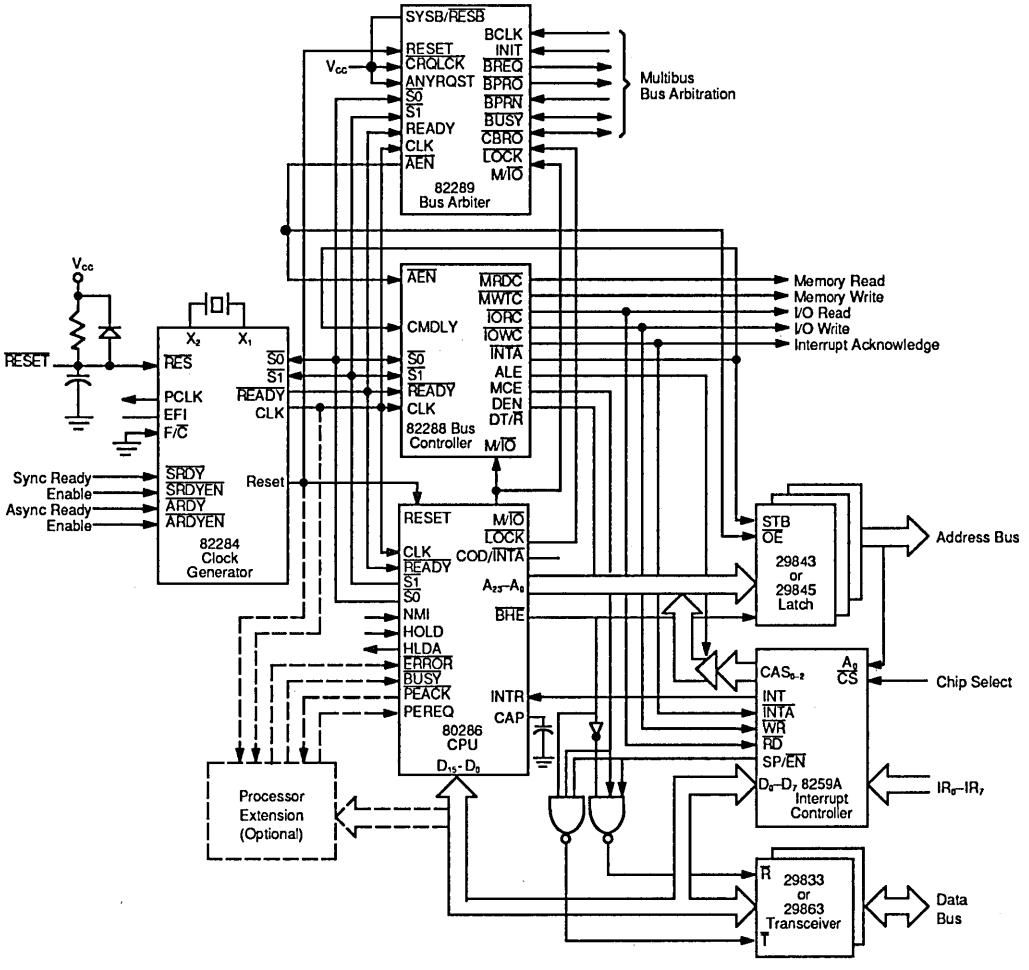


Figure 37. MULTIBUS System Bus Interface

03552-35

Figure 38 shows the interface of the 80286 with the Am2968 Dynamic Memory Controller. The interface is a timing controller which consists of some control logic and a delay line. The timing controller runs asynchronously to the CPU. It arbitrates between memory requests and refresh requests by generating the proper signals to the dynamic memory controller and memory. The design described is a simple, cost-effective solution to interfacing the 80286 with the Am2968. A further description about DRAM selection based on processor speed may be found in the Am2968 Application Note.

Two-operand instructions (e.g., MOV and ADD) are usually three to six bytes long. Memory-to-memory operations are provided by a special class of string instructions requiring one to three bytes.

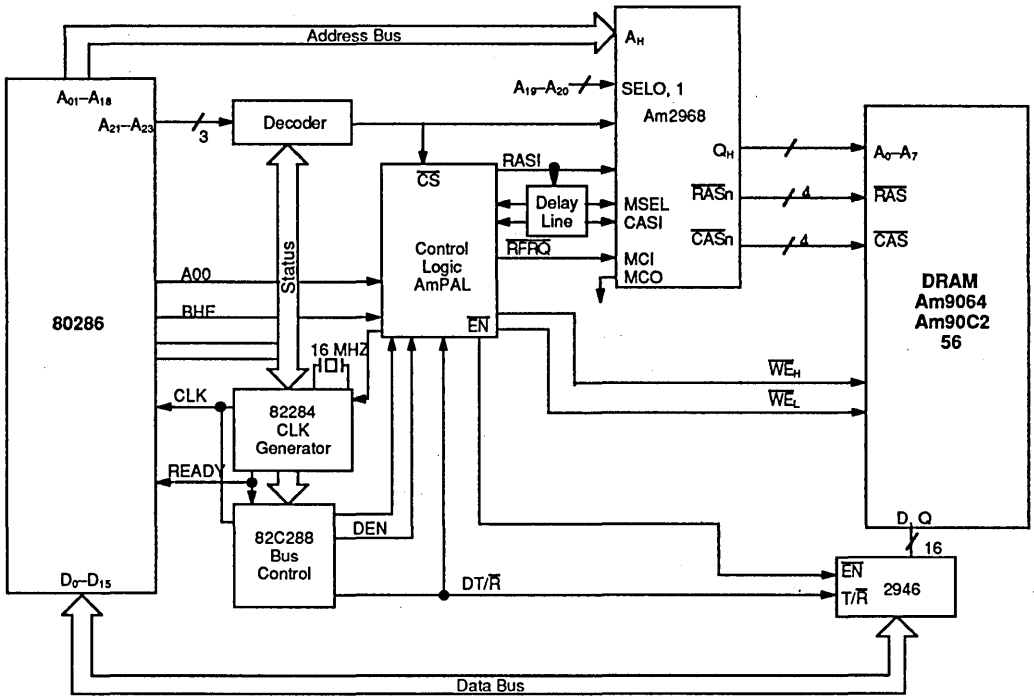
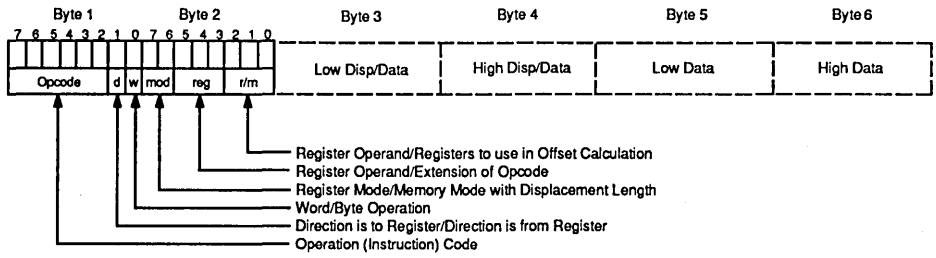


Figure 38. 80286 Interface with the Am2968 Dynamic Memory Controller

03552–36

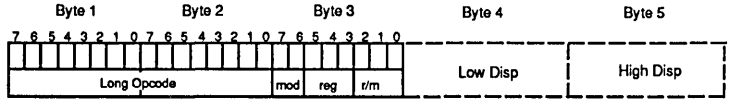
Table 15. 80286 Systems Recommended Pull-up Resistor Values

80286 Pin and Name	Pull-up Value	Purpose
4–$\overline{S1}$	20KΩ±10%	Pull $\overline{S0}$, $\overline{S1}$, and $\overline{PEACK}$ inactive during 80286 hold periods.
5–$\overline{S0}$		
6–$\overline{PEACK}$		
53–$\overline{ERROR}$	20KΩ±10%	Pull $\overline{ERROR}$ and $\overline{BUSY}$ inactive when 80287 not present (or temporarily removed from socket).
54–$\overline{BUSY}$		
63–$\overline{READY}$	910Ω±5%	Pull $\overline{READY}$ inactive within required minimum time (C_L = 150 pF, $I_R \leq$ 7mA).



03552-37

A. Short Opcode Format Example



B. Long Opcode Format Example

03552-38

Figure 39. 80286 Instruction Format Examples

1

80286 INSTRUCTION SET SUMMARY

Instruction Timing Notes

The instruction clock counts listed below establish the maximum execution rate of the 80286. With no delays in bus cycles, the actual clock count of an 80286 program will average 5% more than the calculated clock count, due to instruction sequences which execute faster than they can be fetched from memory.

To calculate elapsed times for instruction sequences, multiply the sum of all instruction clock counts, as listed in the table below, by the processor clock period. An 8-MHz processor clock has a clock period of 125 nanoseconds and requires an 80286 system clock (CLK input) of 16 MHz.

Instruction Clock Count Assumptions

1. The instruction has been prefetched, decoded, and is ready for execution. Control transfer instruction clock counts include all time required to fetch, decode, and prepare the next instruction for execution.
2. Bus cycles do not require wait states.
3. There are no processor extension data transfer or local bus HOLD requests.
4. No exceptions occur during instruction execution.

Instruction Set Summary Notes

Addressing displacements selected by the MOD field are not shown. If necessary they appear after the instruction fields shown.

Above/below refers to unsigned value

Greater refers to positive signed value

Less refers to less positive (more negative) signed values

if d = 1 then to register; if d = 0 then from register
if w = 1 then word instruction; if w = 0 then byte instruction
if s = 0 then 16-bit immediate data to form the operand
if s = 0 then an immediate data byte is sign-extended to form the 16-bit operand
x = don't care
z = used for string primitives for comparison with ZF FLAG

If two clock counts are given, the smaller refers to a register operand and the larger refers to a memory operand.

* = add one clock if offset calculation requires summing 3 elements
n = number of times repeated
m = number of bytes of code in next instruction

Level (L)—Lexical nesting level of the procedure

The following comments describe possible exceptions, side effects, and allowed usage for instructions in both operating modes of the 80286.

Real Address Mode Only

1. This is a protected mode instruction. Attempted execution in real address mode will result in an undefined opcode exception (6).
2. A segment overrun exception (13) will occur if a word operand reference at offset FFFF(H) is attempted.
3. This instruction may be executed in real address mode to initialize the CPU for protected mode.
4. The IOPL and NT fields will remain 0.
5. Processor extension segment overrun interrupt (9) will occur if the operand exceeds the segment limit.

Either Mode

6. An exception may occur, depending on the value of the operand.
7. $\overline{\text{LOCK}}$ is automatically asserted regardless of the presence or absence of the LOCK instruction prefix.
8. $\overline{\text{LOCK}}$ does not remain active between all operand transfers.

Protected Virtual Address Mode Only

9. A general protection exception (13) will occur if the memory operand cannot be used due to either a segment limit or access rights violation. If a stack segment limit is violated, a stack segment overrun exception (12) occurs.
10. For segment load operations, the CPL, RPL, and DPL must agree with privilege rules to avoid an exception. The segment must be present to avoid a not-present exception (11). If the SS register is the destination, and a segment-not-present violation occurs, a stack exception (12) occurs.
11. All segment descriptor accesses in the GDT or LDT made by this instruction will automatically assert $\overline{\text{LOCK}}$ to maintain descriptor integrity in multiprocessor systems.
12. JMP, CALL, INT, RET, IRET instructions referring to another code segment will cause a general protection exception (13) if any privilege rule is violated.
13. A general protection exception (13) occurs if CPL $\neq$ 0.
14. A general protection exception (13) occurs if CPL > IOPL.
15. The IF field of the flag word is not updated if CPL > IOPL. The IOPL field is updated only if CPL = 0.
16. Any violation of privilege rules as applied to the selector operand do not cause a protection exception; rather, the instruction does not return a result and the zero flag is cleared.
17. If the starting address of the memory operand violates a segment limit, or an invalid access is at-

tempted, a general protection exception (13) will occur before the ESC instruction is executed. A stack segment overrun exception (12) will occur if the stack limit is violated by the operand's starting address. If a segment limit is violated during an attempted data transfer, then a processor extension segment overrun exception (9) occurs.

18. The destination of an INT, JMP, CALL, RET, or IRET instruction must be in the defined limit of a code segment or a general protection exception (13) will occur.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65 to + 150° C
Voltage on Any Pin with
Respect to Ground ..... –1.0 to + 7.0 V
Power Dissipation ..... 3.15 Watts

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (Tc) ..... 0 to +85° C
Supply Voltage (Vcc) ..... 5 V ±5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS (V_{CC} = 5 V ±5%, T_{CASE} = 0 to +85° C)

Parameter	Description	Test Conditions	Min.	Max.	Unit
V_{IL}	Input LOW Voltage		–.5	8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + .5	V
V_{ILC}	CLK Input LOW Voltage		–.5	.6	V
V_{IHC}	CLK Input HIGH Voltage		3.8	V_{CC} + .5	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA		.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = –400 µA	2.4		V
I_{LI}	Input Leakage Current	0 V ≤ V_{IN} ≤ V_{CC}		±10	µA
I_{LO}	Output Leakage Current	0.45 V ≤ V_{OUT} ≤ V_{CC}		±10	µA
I_{CC}	Supply Current (turn on, 0°C)	Note 1		600	mA
C_{CLK}	CLK Input Capacitance	Fc = 1 MHz		20	pF
C_{IN}	Other Input Capacitance	Fc = 1 MHz		10	pF
C_O	Input /Output Capacitance	Fc = 1 MHz		20	pF
I_{LO}	Output Leakage Current	0 V ≤ V_{OUT} ≤ 0.45 V		±1	mA
I_{IL}	Input Sustaining Current on $\overline{BUSY}$ and $\overline{ERROR}$ pins	V_{IN} = 0 V	30	500	µA
I_{LCR}	Input CLK Leakage Current	0.45 ≤ V_{IN} ≤ V_{CC}		±10	µA
I_{LCR}	Input CLK Leakage Current	0 V ≤ V_{IN} ≤ 0.45V		+1	mA

Note: Low temperature is worst case.

SWITCHING CHARACTERISTICS

V_{CC} = +5 V ± 5%, T_{CASE} = 0° to +85° C
AC Timings are referenced to 0.8 V and 2.0 V points of signals as illustrated in datasheet waveforms, unless otherwise noted.

Parameters	Description	Test Conditions	8 MHz		10 MHz		Unit
			Min.	Max.	Min.	Max.	
1	System Clock (CLK) Period		62	125	50	125	ns
2	System Clock (CLK) LOW Time	at 1.0 V	15	100	12	109	ns
3	System Clock (CLK) HIGH Time	at 3.6 V	25	110	16	113	ns
17	System Clock (CLK) Rise Time	1.0 V to 3.6 V		10		8	ns
18	System Clock (CLK) Fall Time	3.6 V to 1.0 V		10		8	ns
4	Asynchronous Inputs Setup Time	Note 1	20		20		ns
5	Asynchronous Inputs Hold Time	Note 1	20		20		ns
6	RESET Setup Time		28		23		ns
7	RESET Hold Time		5		5		ns
8	Read Data Setup Time		10		8		ns
9	Read Data Hold Time		8		8		ns
10	$\overline{\text{READY}}$ Setup Time		38		26		ns
11	$\overline{\text{READY}}$ Hold Time		25		25		ns
12	Status/$\overline{\text{PEACK}}$ Valid Delay	Note 2, Note 3	1	40	–	–	ns
12a	Status/$\overline{\text{PEACK}}$ Active Delay	Note 2, Note 3	–	–	1	22	ns
12b	Status/$\overline{\text{PEACK}}$ Inactive Delay	Note 2, Note 3	–	–	1	30	ns
13	Address Valid Delay	Note 2, Note 3	1	60	1	35	ns
14	Write Data Valid Delay	Note 2, Note 3	0	50	0	30	ns
15	Address/Status/Data Float Delay	Note 2, Note 4	0	50	0	47	ns
16	HLDA Valid Delay	Note 2, Note 3	0	50	0	47	ns
19	Address Valid to Status Valid Setup Time	Note 3, Note 5, Note 6	38		27		ns

Notes: 1. Asynchronous inputs are INTR, NMI, HOLD PEREQ, $\overline{\text{ERROR}}$, and $\overline{\text{BUSY}}$. This specification is given only for testing purposes, to assure recognition at a specific CLK edge.

2. Delay from 1.0 V on the CLK to 0.8 V or 2.0 V or float on the output as appropriate for valid or floating condition.
3. Output load: C_L = 100 pF.
4. Float condition occurs when output current is less than I_{LO} in magnitude.
5. Delay measured from address either reaching 0.8 V or 2.0 V (valid) to status going active reaching 2.0 V or status going inactive reaching 0.8 V.
6. For load capacitance of 10 pF on STATUS/$\overline{\text{PEACK}}$ lines, subtract typically 7 ns for 8 MHz spec, and maximum 7 ns for 10 MHz spec.

SWITCHING CHARACTERISTICS (continued)

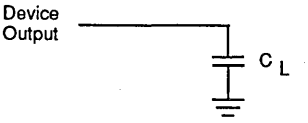
V_{CC} = +5 V ± 5%, T_{CASE} = 0° to +85° C

AC Timings are referenced to 0.8 V and 2.0 V points of signals as illustrated in datasheet waveforms, unless otherwise noted.

Parameters	Description	Test Conditions	12.5 MHz Min.	12.5 MHz Max.	16 MHz Min.	16 MHz Max.	Unit
1	System Clock (CLK) Period		40	125	31	125	ns
2	System Clock (CLK) LOW Time	at 1.0 V	11	112	10	113	ns
3	System Clock (CLK) HIGH Time	at 3.6 V	13	114	12	115	ns
17	System Clock (CLK) Rise Time	1.0 V to 3.6 V		8	8		ns
18	System Clock (CLK) Fall Time	3.6 V to 1.0 V		8	8		ns
4	Asynchronous Inputs Setup Time	Note 1	15		11		ns
5	Asynchronous Inputs Hold Time	Note 1	15		11		ns
6	RESET Setup Time		18		14		ns
7	RESET Hold Time		5		3		ns
8	Read Data Setup Time		5		5		ns
9	Read Data Hold Time		6		5		ns
10	$\overline{\text{READY}}$ Setup Time		22		15		ns
11	$\overline{\text{READY}}$ Hold Time		20				ns
12	Status/$\overline{\text{PEACK}}$ Valid Delay	Note 2, Note 3	–	–	1	18	ns
12a	Status/$\overline{\text{PEACK}}$ Active Delay	Note 2, Note 3	3	18	1	18	ns
12b	Status/$\overline{\text{PEACK}}$ Inactive Delay	Note 2, Note 3	3	20	1	20	ns
13	Address Valid Delay	Note 2, Note 3	1	32	1	29	ns
14	Write Data Valid Delay	Note 2, Note 3	0	30	0	22	ns
15	Address/Status/Data Float Delay	Note 2, Note 4	0	32	0	29	ns
16	HLDA Valid Delay	Note 2, Note 3	0	25	0	25	ns
19	Address Valid to Status Valid Setup Time	Note 3, Note 5, Note 6	22		22		ns

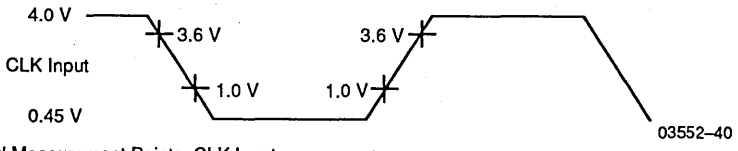
Notes: 1. Asynchronous inputs are INTR, NMI, HOLD PEREQ, $\overline{\text{ERROR}}$, and $\overline{\text{BUSY}}$. This specification is given only for testing purposes, to assure recognition at a specific CLK edge.

2. Delay from 1.0 V on the CLK to 0.8 V or 2.0 V or float on the output as appropriate for valid or floating condition.
3. Output load: C_L = 100 pF.
4. Float condition occurs when output current is less than I_{LO} in magnitude.
5. Delay measured from address either reaching 0.8 V or 2.0 V (valid) to status going active reaching 2.0 V or status going inactive reaching 0.8 V.
6. For load capacitance of 10 pF on STATUS/$\overline{\text{PEACK}}$ lines, subtract typically 7 ns for 8 MHz spec, and maximum 7 ns for 10 MHz spec.



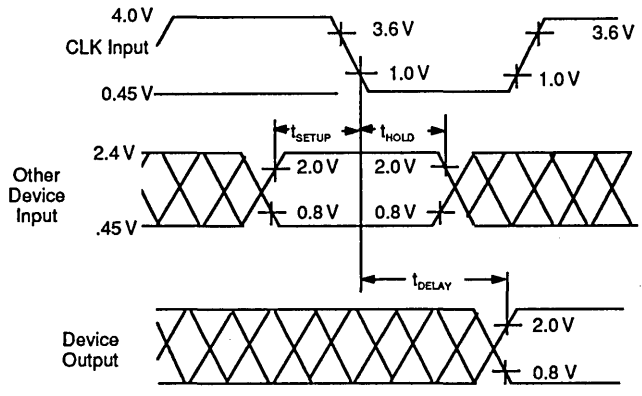
03552–39

Note: 7. AC Test Loading on Outputs



03552–40

Note: 8. AC Drive and Measurement Points–CLK Input

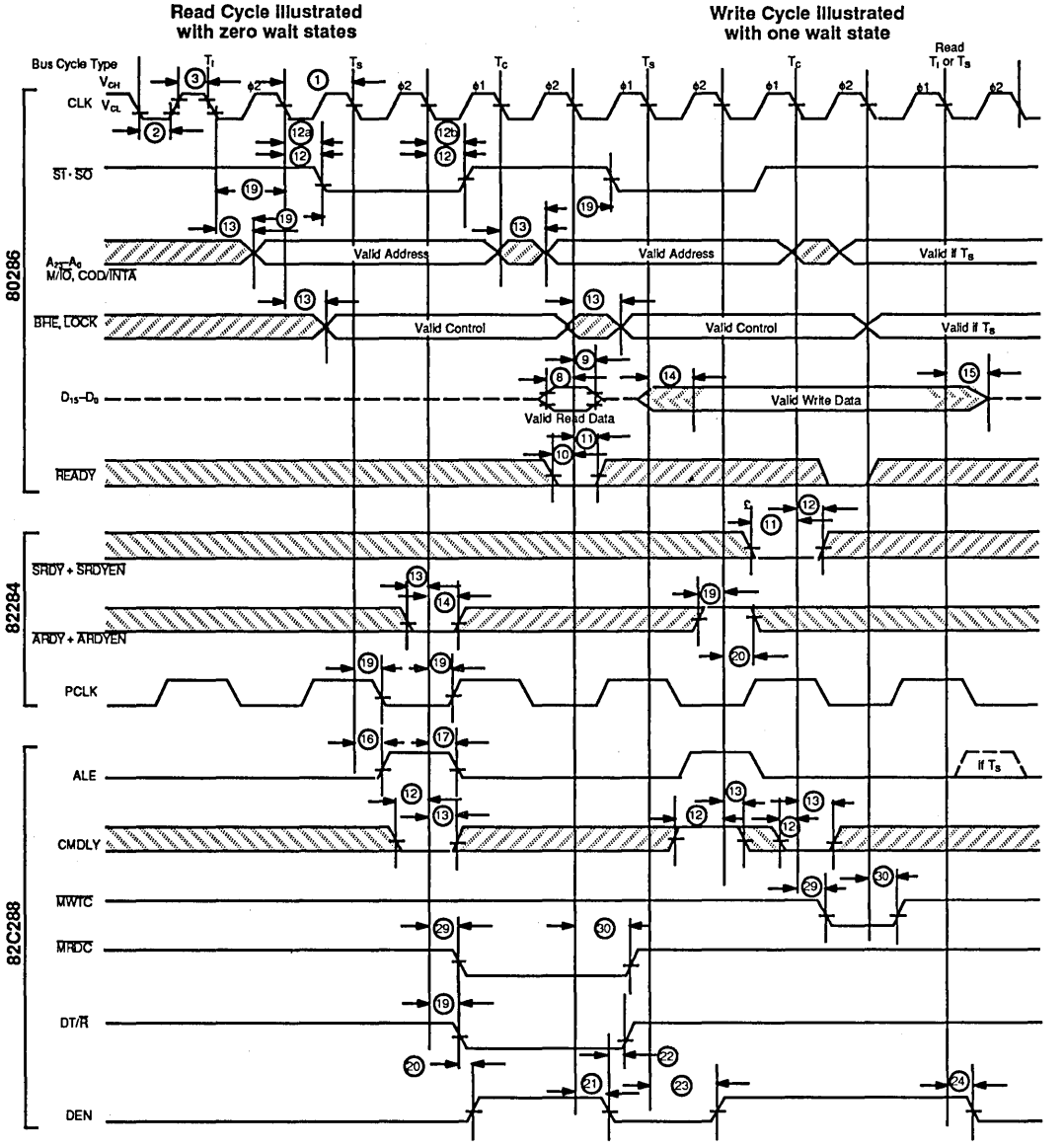


03552–41

Note: AC Setup, Hold and Delay Time Measurement—General

SWITCHING WAVEFORMS

Major Cycle Timing

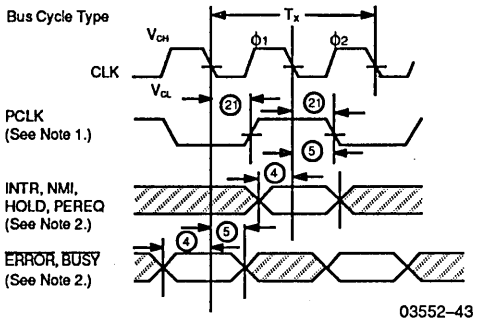


Note: The modified timing is due to the $\overline{\text{CMDLY}}$ signal being active.

03552–42

SWITCHING WAVEFORMS (continued)

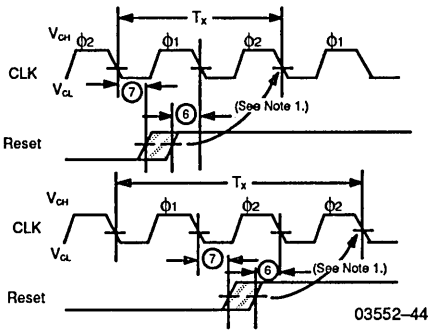
80286 Asynchronous Input Signal Timing



Notes: 1. PCLK indicates which processor cycle phase will occur on the next CLK. PCLK may not indicate the correct phase until the first bus cycle is performed.

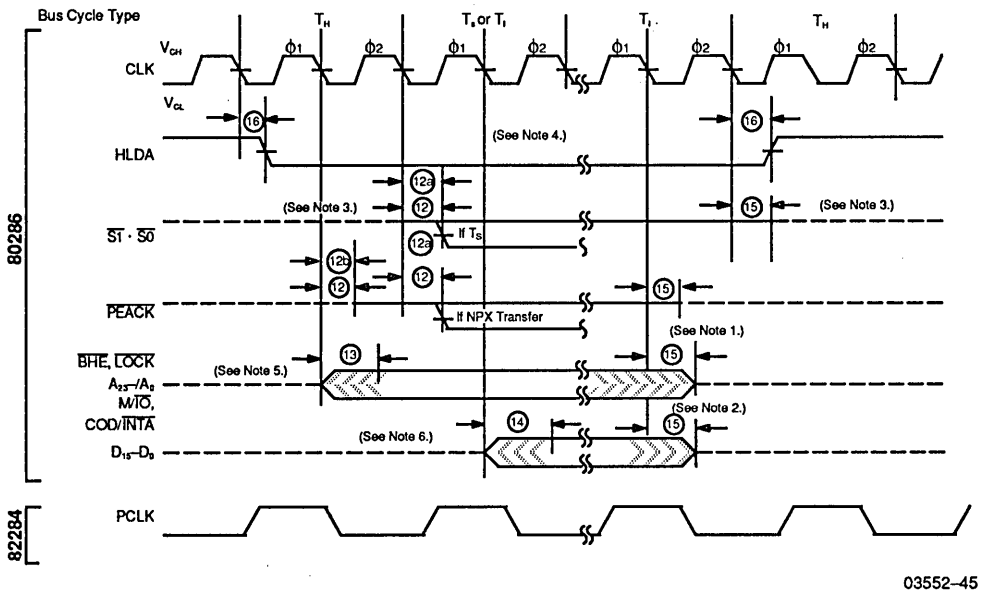
2. These inputs are asynchronous. The setup and hold times shown assure recognition for testing purposes.

80286 Reset Input Timing and Subsequent Processor Cycle Phase



Note: When RESET meets the set-up time shown, the next CLK will start or repeat $\phi1$ of a processor cycle.

Exiting and Entering Hold



Notes: 1. These signals may not be driven by the 80286 during the time shown. The worst case in terms of latest float time is shown.

2. The data bus will be driven as shown if the last cycle before T_i in the diagram was a write T_c.

3. The 80286 floats its status pins during T_H. External 20 kΩ resistors keep these signals high (see Table 15).

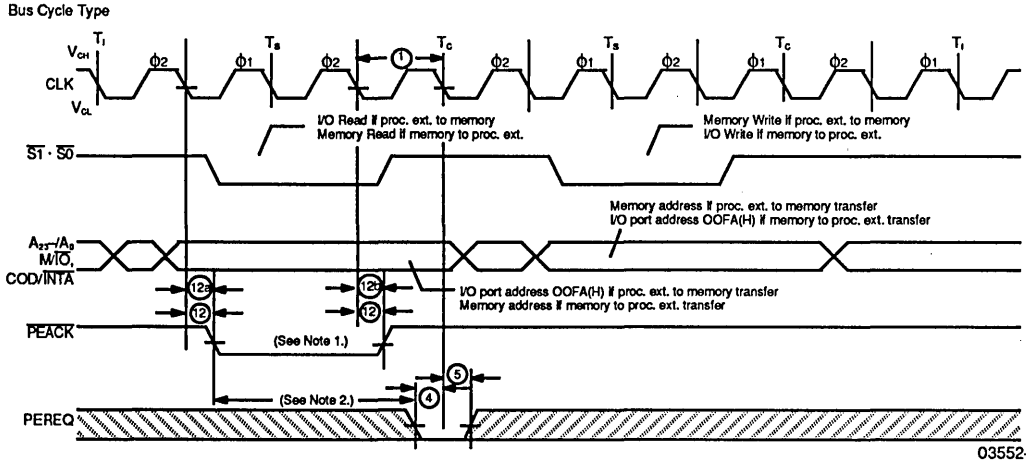
4. For HOLD request set-up to HLDA, refer to Figure 34.

5. $\overline{BHE}$ and $\overline{LOCK}$ are driven at this time but will not become valid until T_s.

6. The data bus will remain in three-state OFF if a read cycle is performed.

SWITCHING WAVEFORMS (continued)

80286 PEREQ/$\overline{\text{PEACK}}$ Timing Required PEREQ Timing for One Transfer Only



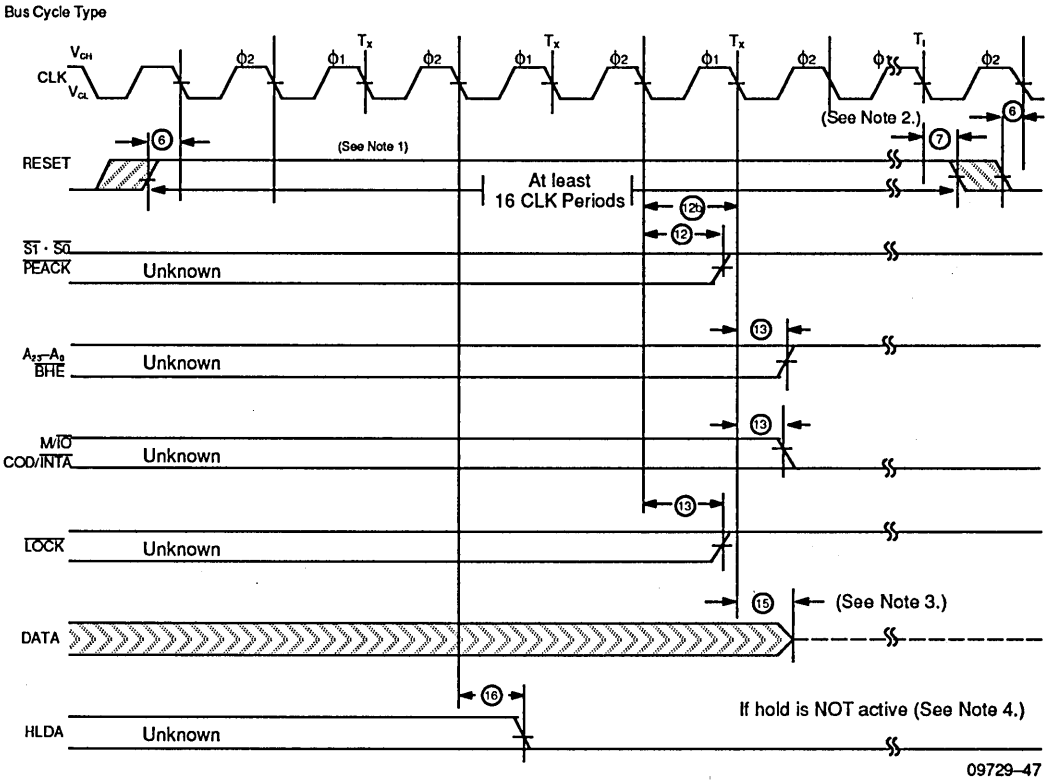
03552-46

Assuming word-aligned memory operand; if odd-aligned, 80286 transfers to/from memory byte-at-a-time with two memory cycles.

Notes: 1. $\overline{\text{PEACK}}$ always goes active during the first bus operation of a processor extension data operand transfer sequence. The first bus operation will be either a memory read at operand address or I/O read at port address OOFA(H).

2. To prevent a second processor extension data operand transfer, the worst case maximum time (shown above) is: 3 x 1 – 11 max – 4 min. The actual, configuration dependent, maximum time is: 3 x 1 – 11 max – 4 min + A x 2 x 1. A is the number of extra T_c states added to either the first or second bus operation of the processor extension data operand transfer sequence.

Initial 80286 Pin State During Reset



Notes: 1. Set-up time for RESET ↑ may be violated with the consideration that ϕ1 of the processor clock may begin one system CLK period later.

2. Set-up and hold times for RESET ↓ must be met for proper operation, but RESET ↓ may occur during ϕ1 or ϕ2.
3. The data bus is only guaranteed to be in three-state OFF at the time shown.
4. HOLD is acknowledged during RESET, causing HLDA to go active and the appropriate pins to float. If HOLD remains active while RESET goes inactive, the 80286 remains in HOLD state and will not perform any bus accesses until HOLD is deactivated.

80286 INSTRUCTION SET SUMMARY

Function	Format				Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
DATA TRANSFER								
MOV = Move:								
Register to Register/Memory	1000100w	mod reg r/m			2,3*	2,3*	2	9
Register/Memory to Register	1000101w	mod reg r/m			2,5*	2,5*	2	9
Immediate to register/ Memory	1100011w	mod000r/m	data	data if w = 1	2,3*	2,3*	2	9
Immediate to register	1011wreg	data	data if w = 1		2	2		
Memory to accumulator	1010000w	addr-low	addr-high		5	5	2	9
Accumulator to memory	1010001w	addr-low	addr-high		3	3	2	9
Register/memory to segment register	10001110	mod0reg r/m			2,5*	17,19*	2	9,10,11
Segment register to register/memory	10001100	mod0reg r/m			2,3*	2,3*	2	9
PUSH = Push:								
Memory	11111111	mod110r/m			5*	5*	2	9
Register	01010reg				3	3	2	9
Segment register	000reg110				3	3	2	9
Immediate	0110100	data	data if s = 0		3	3	2	9
PUSHA = Push All	01100000				17	17	2	9
POP = Pop								
Memory	10001111	mod000r/m			5*	5*	2	9
Register	01011reg				5	5	2	9
Segment register	000reg111	(reg ≠ 01)			5	20	2	9,10,11
POPA = Pop All	01100001				19	19	2	9
XCHG = Exchange:								
Register/memory with register	1000011w	modreg r/m			3,5*	3,5*	2,7	7,9
Register with accumulator	10010reg				3	3		
IN = Input from:								
Fixed port	1110010w	port			5	5		14
Variable port	1110110w				5	5		14
OUT = Output to:								
Fixed port	1110011w	port			3	3		14
Variable port	1110111w				3	3		14
XLAT = Translate byte to AL	11010111				5	5		9
LEA = Load EA to register	10001101	mod reg r/m			3*	3*		
LDS = Load pointer to DS	11000101	mod reg r/m	(mod ≠ 11)		7*	21*	2	9,10,11
LES = Load pointer to ES	11000100	mod reg r/m	(mod ≠ 11)		7*	21*	2	9,10,11
LAHF = Load AH with flags	10011111				2	2		
SAHF = Store AH into flags	10011110				2	2		
PUSHF = Push flags	10011100				3	3	2	9
POPF = Pop flags	10011101				5	5	2,4	9,15

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

See footnotes at end of this document.

Function	Format				Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
ARITHMETIC								
ADD=Add:								
Reg/memory with register to either	000000dw	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	100000sw	mod 0 0 r/m	data	data if s:w=01	3,7*	3,7*	2	9
Immediate to accumulator	0000010w	data	data if w=1		3	3		
ADC=Add with carry:								
Reg/memory with register to either	000100dw	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	100000sw	mod 0 1 0 r/m	data	data if s:w=01	3,7*	3,7*	2	9
Immediate to accumulator	0001010w	data	data if w=1		3	3		
INC=Increment:								
Register/memory	1111111w	mod 0 0 0 r/m			2,7*	2,7*	2	9
Register	01000reg				2	2		
SUB=Subtract:								
Reg/memory and register to either	001010dw	mod reg r/m			2,7*	2,7*	2	9
Immediate from register/memory	100000sw	mod 1 0 1 r/m	data	data if s:w=1	3,7*	3,7*	2	9
Immediate from accumulator	0001110w	data	data if w=1		3	3		
SBB=Subtract with borrow:								
Reg/memory and register to either	000110dw	mod reg r/m			2,7*	2,7*	2	9
Immediate from register/memory	100000sw	mod 0 1 1 r/m	data	data if s:w=01	3,7*	3,7*	2	9
Immediate from accumulator	0010110w	data	data if w=1		3	3		
DEC=Decrement:								
Register/memory	1111111w	mod 0 0 1 r/m			2,7*	2,7*	2	9
Register	01001reg				2	2		
CMP=Compare:								
Register/memory with register	0011101w	mod reg r/m			2,6*	2,6*	2	9
Register with register/memory	0011100w	mod reg r/m			2,7*	2,7*	2	9
Immediate with register/memory	100000sw	mod 1 1 1 r/m	data	data if s:w=01	3,6*	3,6*	2	9
Immediate with accumulator	0011110w	data	data if w=1		3	3		
NEG =Change sign	1111011w	mod 0 1 1 r/m			2	7*	2	7
AAA=ASCII adjust for add	00110111				3	3		
DAA=Decimal adjust for add	00100111				3	3		
AAS=ASCII adjust for subtract	00111111				3	3		
DAS=Decimal adjust for subtract	00101111				3	3		
MUL=Multiply (unsigned)	1111011w	mod 1 0 0 r/m						
Register-Byte					13	13		
Register-Word					21	21		
Memory-Byte					16*	16*	2	9
Memory-Word					24*	24*	2	9
IMUL=Integer multiply (signed)	1111011w	mod 1 0 1 r/m						
Register-Byte					13	13		
Register-Word					21	21		
Memory-Byte					16*	16*	2	9
Memory-Word					24*	24*	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.
See footnotes at end of this document.

80286 INSTRUCTION SET SUMMARY (continued)

Function	Format	Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
ARITHMETIC (Continued)					
IMUL = Integer immediate multiply: (signed)	0 1 1 0 1 0 a 1 | mod reg r/m | data | data if a = 0	21,24*	21,24*	2	9
DIV = Divide (unsigned):	1 1 1 1 0 1 1 w | mod 1 1 0 r/m				
Register-Byte		14	14		
Register-Word		22	22		
Memory-Byte		17*	17*	2,6	6,9
Memory-Word		25*	25*	2,6	6,9
IDIV = Integer divide (signed)	1 1 1 1 0 1 1 w | mod 1 1 1 r/m				
Register-Byte		17	17		
Register-Word		25	25		
Memory-Byte		20*	20*	2	9
Memory-Word		28*	28*	2	9
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0 | 0 0 0 0 1 0 1 0	16	16		
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1 | 0 0 0 0 1 0 1 0	14	14		
CBW = Convert byte to word	1 0 0 1 1 0 0 0	2	2		
CWD = Convert word to double word	1 0 0 1 1 0 0 1	2	2		
LOGIC					
Shift/Rotate Instructions:					
Register/Memory by 1	1 1 0 1 0 0 0 w | mod TTT r/m	2,7*	2,7*	2	9
Register/Memory by CL	1 1 0 1 0 0 1 w | mod TTT r/m	5+n, 8+n*	5+n, 8+n*	2	9
Register Memory by Count	1 1 0 0 0 0 0 w | mod TTT r/m	5+n, 8+n*	5+n, 8+n*	2	9
	TTT — **Instruction**: 0 0 0 ROL; 0 0 1 ROR; 0 1 0 RCL; 0 1 1 RCR; 1 0 0 SHL/SAL; 1 0 1 SHR; 1 1 1 SAR				
AND = And:					
Reg/memory and register to either	0 0 1 0 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 1 0 0 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 1 0 0 1 0 w | data | data if w = 1	3	3		
TEST = And function to flags, no result:					
Register/memory and register	1 0 0 0 0 1 0 w | mod reg r/m	2,6*	2,6*	2	9
Immediate data and register/memory	1 1 1 1 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1	3,6*	3,6*	2	9
Immediate data and accumulator	1 0 1 0 1 0 0 w | data | data if w = 1	3	3		
OR = Or:					
Reg/memory and register to either	0 0 0 0 1 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 0 0 1 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 0 0 1 1 0 w | data | data if w = 1	3	3		
XOR = Exclusive or:					
Reg/memory and register to either	0 0 1 1 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 1 1 0 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 1 1 0 1 0 w | data | data if w = 1	3	3		
NOT = Invert register/memory	1 1 1 1 0 1 1 w | mod 0 1 0 r/m	2,7*	2,7*	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

See footnotes at end of this document.

Function	Format	Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
STRING MANIPULATION:					
MOVS = Move byte/word	1 0 1 0 0 1 0 w	5	5	2	9
CMPS = Compare byte/word	1 0 1 0 0 1 1 w	8	8	2	9
SCAS = Scan byte/word	1 0 1 0 1 1 1 w	7	7	2	9
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w	5	5	2	9
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w	3	3	2	9
INS = Input byte/wd from DX port	0 1 1 0 1 1 0 w	5	5	2	9,14
OUTS = Output byte/wd to DX port	0 1 1 0 1 1 1 w	5	5	2	9,14
Repeated by count in CX					
MOVS = Move string	1 1 1 1 0 0 1 0 | 1 0 1 0 0 1 0 w	5+4n	5+4n	2	9
CMPS = Compare string	1 1 1 1 0 0 1 z | 1 0 1 0 0 1 1 w	5+9n	5+9n	2	9
SCAS = Scan string	1 1 1 1 0 0 1 z | 1 0 1 0 1 1 1 w	5+8n	5+8n	2	9
LODS = Load string	1 1 1 1 0 1 0 | 1 0 1 0 1 1 0 w	5+4n	5+4n	2	9
STOS = Store string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 0 1 w	4+3n	4+3n	2	9
INS = Input string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 0 w	5+4n	5+4n	2	9,14
OUTS = Output string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 1 w	5+4n	5+4n	2	9,14
CONTROL TRANSFER					
CALL = Call:					
Direct within segment	1 1 1 0 1 0 0 0 | disp-low | disp-high	7+m	7+m	2	8
Register memory indirect within segment	1 1 1 1 1 1 1 1 | mod 0 1 0 r/m	7+m,11+m	7+m,11+m	2	8,9
Direct intersegment	1 0 0 1 1 0 1 0 | segment offset | segment selector	13+m	26+m	2	8,11,12
Protected Mode Only (Direct Intersegment):					
Via call gate to same privilege level		41+m		8,11,12	
Via call gate to different privilege level, no parameters		82+m		8,11,12	
Via call gate to different privilege level, x parameters		86+4x+m		8,11,12	
Via TSS		177+m		8,11,12	
Via task gate		182+m		8,11,12	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 0 1 1 r/m | (mod ≠ 11)	16+m	29+m*	2	8,9,11,12
Protected Mode Only (Indirect Intersegment):					
Via call gate to same privilege level			44+m*		8,9,11,12
Via call gate to different privilege level, no parameters			83+m*		8,9,11,12
Via call gate to different privilege level, x parameters			90+4x+m*		8,9,11,12
Via TSS			180+m*		8,9,11,12
Via task gate			185+m*		8,9,11,12
JMP = Unconditional Jump					
Short/long	1 1 1 0 1 0 1 1 | disp-low	7+m	7+m		8
Direct within segment	1 1 1 0 1 0 0 1 | disp-low | disp-high	7+m	7+m		8
Register/mem indirect within segment	1 1 1 1 1 1 1 1 | mod 1 0 0 r/m	7+m,11+m*	7+m,11+m*	2	8,9
Direct intersegment	1 1 1 0 1 0 1 0 | segment offset | segment selector	11+m	23+m		8,11,12
Protected Mode Only (Indirect Intersegment):					
Via call gate to same privilege level			38+m		8,11,12
Via TSS			175+m		8,11,12
Via task gate			180+m		8,11,12
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 1 0 1 r/m | (mod ≠ 11)	15+m*	26+m*	2	8,9,11,12

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

See footnotes at end of this document.

Function	Format			Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
CONTROL TRANSFER (Continued):							
Protected Mode Only (Indirect Intersegment)							
Via call gate to same privilege level					41+m*		8,9,11,12
Via TSS					178+m*		8,9,11,12
Via task gate					183+m*		8,9,11,12
RET = Return from CALL:							
Within segment	11000011			11+m	11+m	2	8,9
Within seg adding immed to SP	11000010	data-low	data-high	11+m	11+m	2	8,9
Intersegment	11001011			15+m	25+m	2	8,9,11,12
Intersegment adding immediate to SP	11001010	data-low	data-high	15+m		2	8,9,11,12
Protected Mode Only (RET):							
To different privilege level					55+m		
JE/JZ = Jump on equal zero	01110100	disp		7+m or 3	7+m or 3		8
JL/JNGE = Jump on less not greater or equal	01111100	disp		7+m or 3	7+m or 3		8
JLE/JNG = Jump on less or equal not greater	01111110	disp		7+m or 3	7+m or 3		8
JB/JNAE = Jump on below not above or equal	01110010	disp		7+m or 3	7+m or 3		8
JBE/JNA = Jump on below or equal not above	01110110	disp		7+m or 3	7+m or 3		8
JP/JPE = Jump on parity/parity even	01111010	disp		7+m or 3	7+m or 3		8
JO = Jump on overflow	01110000	disp		7+m or 3	7+m or 3		8
JS = Jump on sign	01111000	disp		7+m or 3	7+m or 3		8
JNE/JNZ = Jump on not equal not zero	01110101	disp		7+m or 3	7+m or 3		8
JNL/JGE = Jump on not less greater or equal	01111101	disp		7+m or 3	7+m or 3		8
JNLE/JG = Jump on not less or equal greater	01111111	disp		7+m or 3	7+m or 3		8
JNB/JAE = Jump on not below above or equal	01110011	disp		7+m or 3	7+m or 3		8
JNBE/JA = Jump on not below or equal above	01110111	disp		7+m or 3	7+m or 3		8
JNP/JPO = Jump on not par/par odd	01111011	disp		7+m or 3	7+m or 3		8
JNO = Jump on not overflow	01110001	disp		7+m or 3	7+m or 3		8
JNS = Jump on not sign	01111001	disp		7+m or 3	7+m or 3		8
LOOP = Loop CX Times	11100010	disp		8+m or 4	8+m or 4		8
LOOPZ/LOOPE = Loop while zero equal	11100001	disp		8+m or 4	8+m or 4		8
LOOPNZ/LOOPNE = Loop while not zero equal	11100000	disp		8+m or 4	8+m or 4		8
JCXZ = Jump on CX zero	11100011	disp		8+m or 4	8+m or 4		8
ENTER = Enter Procedure	11001000	data-low	data-high	L			
L = 0				11	11	2	9
L = 1				15	15	2	9
L > 1				16–4(L–1)	16–4(L–1)	2	9
LEAVE = Leave Procedure	11001001			5	5	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

See footnotes at end of this document.

Function	Format	Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
CONTROL TRANSFER (Continued):					
INT = Interrupt:					
Type specified	11001101 type	23+m		2	
Type 3	11001100	23+m		2	
INTO = Interrupt on overflow	11001110	24–m or 3 (3 if no) (Interrupt)	24 – or 3 (3 if no) (Interrupt)	2	
Protected Mode Only:					
Via interrupt or trap gate to same privilege level			40+m		8,11,12
Via interrupt or trap gate to fit different privilege level			78+m		8,11,12
Via Task Gate			167+m		8,11,12
IRET = Interrupt return	11001111	17+m	31+m	2,4	8,9,11, 12,15
Protected Mode Only:					
To different privilege level				55+m	8,9,11, 12,15
To different task (NT = 1)				169+m	8,9,11,12
BOUND = Detect value out of range	01100010 mod reg r/m	13	13 (Use INT clock count if exception 6)	2,6	
PROCESSOR CONTROL					
CLC = Clear carry	11111000	2	2		
CMC = Complement carry	11110101	2	2		
STC = Set carry	11111001	2	2		
CLD = Clear direction	11111100	2	2		
STD = Set direction	11111101	2	2		
CLI = Clear interrupt	11111010	3	3		14
STI = Set interrupt	11111011	2	2		14
HLT = Halt	11110100	2	2		13
WAIT = Wait	10011011	3	3		
LOCK = Bus lock prefix	11110000	0	0		14
CTS = Clear task switched flag	00001111 00000110	2	2	3	13
ESC = Processor Extension Escape	10011TTT mod LLL r/m (TTT LL are opcode to processor extension)	9–20*	9–20*	5	17
SEG = Segment override prefix	001 reg 110	0	0		

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

See footnotes at end of this document.

80286 INSTRUCTION SET SUMMARY (continued)

Function	Format	Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
PROTECTION CONTROL:					
LGDT = Load global descriptor table register	00001111 00000001 mod010 r/m	11*	11*	2,3	9,13
SGDT = Store global descriptor table register	00001111 00000001 mod000 r/m	11*	11*	2,3	9,13
LIDT = Load interrupt descriptor table register	00001111 00000001 mod011 r/m	12*	12*	2,3	9,13
SIDT = Store interrupt descriptor table register	00001111 00000001 mod001 r/m	12*	12*	2,3	9
LLDT = Load local descriptor table register from table memory	00001111 00000000 mod010 r/m		17,19*	1	9,11,13
SLDT = Store local descriptor table register to register/memory	00001111 00000000 mod000 r/m		2,3*	1	9
LTR = Load task register from register/memory	00001111 00000000 mod011 r/m		17,19*	1	9,11,13
STR = Store task register to register memory	00001111 00000000 mod001 r/m		2,3*	1	9,11,13
LMSW = Load machine status word from register/memory	00001111 00000001 mod110 r/m	3,6*	3,6*	2,3	9,13
SMSW = Store machine status word	00001111 00000001 mod100 r/m	2,3*	2,3*	2,3	9
LAR = Load access rights from register/memory	00001111 00000010 mod reg r/m		14,16*	1	9,16
LSL = Load segment limit from register/memory	00001111 00000011 mod reg r/m		14,16*	1	9,16
ARPL = Adjust requested privilege level from register/memory	01100011 mod reg r/m		10,11*	2	9
VERR = Verify read access: register/memory	00001111 00000000 mod100 r/m		14,16*	1	9,16
VERR = Verify write access:	00001111 00000000 mod101 r/m		14,16*	1	9,16

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.
See footnotes at end of this document.

Footnotes

The effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field

if mod = 00 then DISP = 0*, disp-low and disp-high are absent

if mod = 01 then DISP = disp-low sign-extended to 16 bits, disp-high is absent

if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP

if r/m = 001 then EA = (BX) + (DI) + DISP

if r/m = 010 then EA = (BP) + (SI) + DISP

if r/m = 011 then EA = (BP) + (DI) + DISP

if r/m = 100 then EA = (SI) + DISP

if r/m = 101 then EA = (DI) + DISP

if r/m = 110 then EA = (BP) + DISP*

if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

REG is assigned according to the following:

REG	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

16-Bit (w = 1)		8-Bit (w = 0)	
000	AX	000	AL
001	CX	001	CL
010	DX	010	DL
011	BX	011	BL
100	SP	100	AH
101	BP	101	CH
110	SI	110	DH
111	DI	111	BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed by the DI register) are computed using the ES segment, which may not be overridden.

Preliminary

80C286

High-Speed CMOS 80286 Microprocessor

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Ultra high-performance processor**
  - —Over 20X performance of 8086
  - —Over 1.5X performance of 386SX-16*
  - —Comparable performance to 386 at same clock speed*
- **Wide range of clock rates**
  - —25 MHz (80C286-25)
  - —20 MHz (80C286-20)
  - —16 MHz (80C286-16)
  - —12.5 MHZ (80C286-12)
- **100% functionally and pin compatible with NMOS 286**
- **Static CMOS design for low power operation**
  - —Standby mode Icc = 5 mA maximum
  - —Operating mode Icc
    - 220 mA max at 12.5 MHz
    - 260 mA at max 16 MHz
    - 310 mA at max 20 MHz
    - 360 mA at max 25 MHz
- **68-lead LCC and 68-lead PLCC packages**

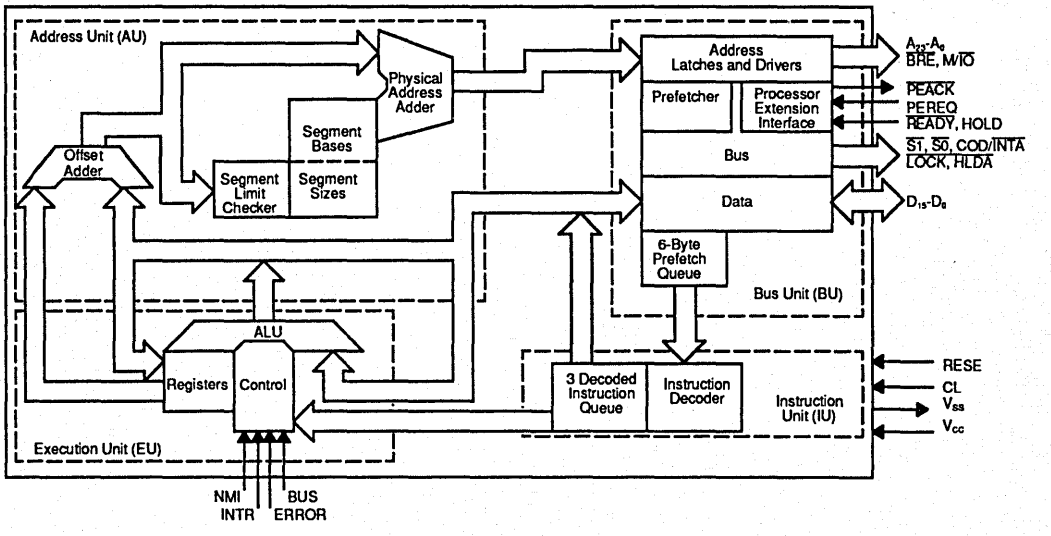
*When running 16-bit code (i.e., DOS or OS/2)

GENERAL DESCRIPTION

AMD's 80C286 is a high-speed implementation of the industry standard 80286 microprocessor. It is 100% functionally compatible with the NMOS version and is a plug compatible replacement. AMD's high-speed CMOS process allows clock speeds much higher than those attainable with NMOS. This CMOS 80286 operates at clock speeds up to 25 MHz.

This CMOS design is a static implementation which allows the processor to be clocked down to DC and still retain full register status. This is useful for designs where power consumption is a consideration as the 80C286 uses only 5 mA of supply current when in standby mode. The 80C286 also retains full functionality from it's maximum clock frequency through very low frequencies down to DC. Since power consumption is proportional to clock speed, the 80C286 may be clocked at a slower rate to draw less current.

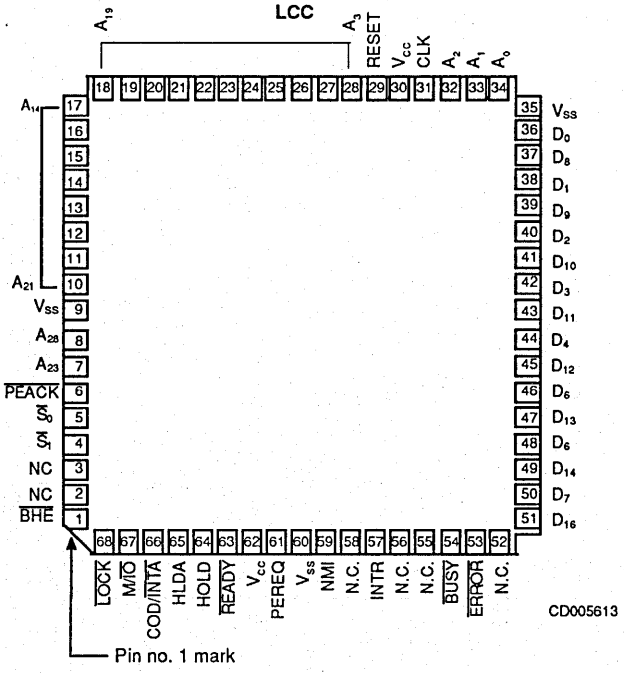
BLOCK DIAGRAM



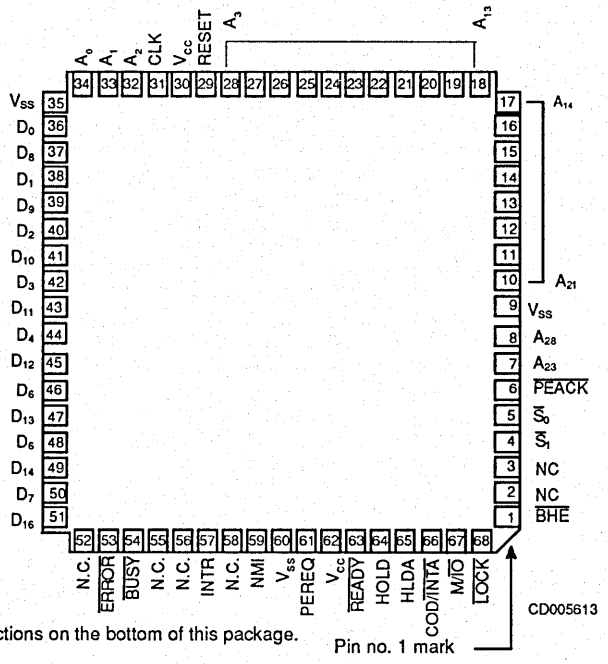
Publication # 09729 Rev. A Amendment /0
Issue Date: May 1989

CONNECTION DIAGRAMS

Component Pad Views—as viewed from underside of component on the PC board.



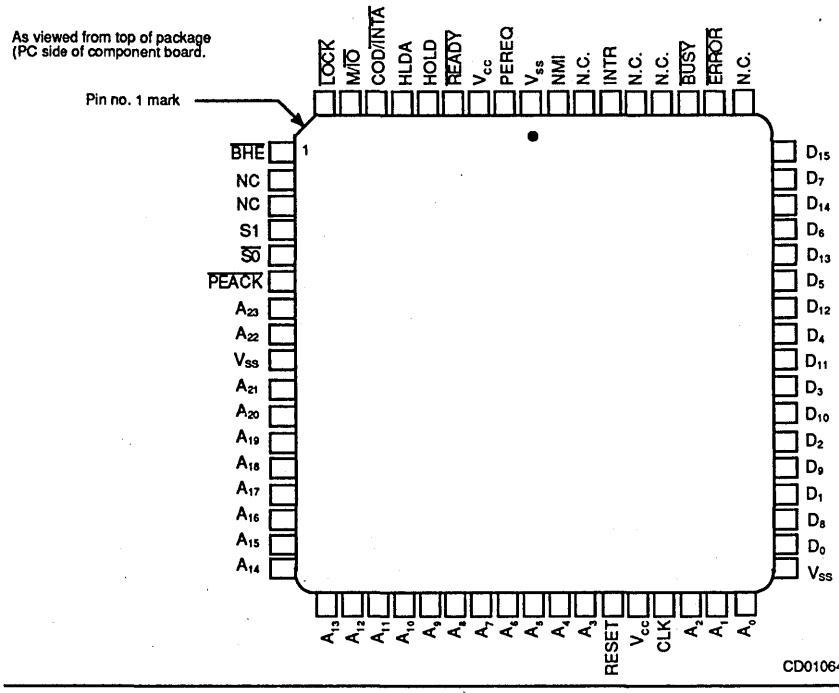
PC Board Views—as viewed from the component side of the PC board.



There are no electrical connections on the bottom of this package.

CONNECTION DIAGRAMS (continued)

PLCC



PIN DESIGNATIONS
(sorted by pin number)

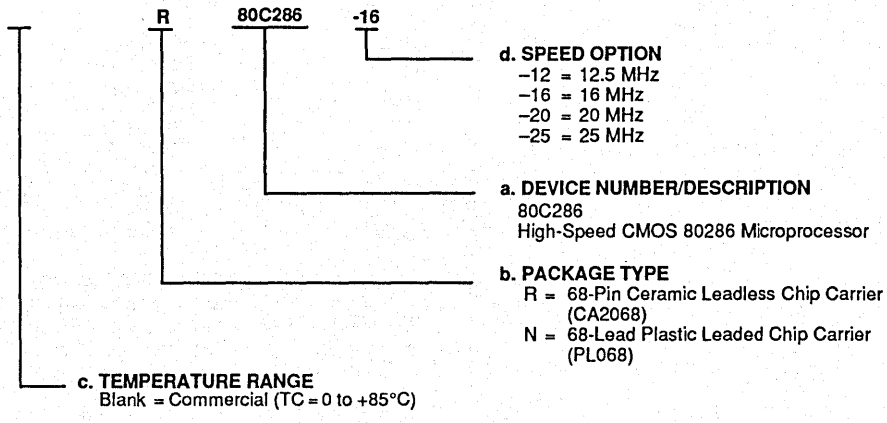
Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	$\overline{BHE}$	24	A_7	47	D_{13}
2	NC	25	A_6	48	D_6
3	NC	26	A_5	49	D_{14}
4	$\overline{S1}$	27	A_4	50	D_7
5	$\overline{S0}$	28	A_3	51	D_{15}
6	$\overline{PEACK}$	29	RESET	52	NC
7	A_{23}	30	Vcc	53	$\overline{ERROR}$
8	A_{22}	31	CLK	54	$\overline{BUSY}$
9	V_{SS}	32	A_2	55	NC
10	A_{21}	33	A_1	56	NC
11	A_{20}	34	A_0	57	INTR
12	A_{19}	35	V_{SS}	58	NC
13	A_{18}	36	D_0	59	NMI
14	A_{17}	37	D_8	60	V_{SS}
15	A_{16}	38	D_1	61	PEREQ
16	A_{15}	39	D_9	62	Vcc
17	A_{14}	40	D_2	63	$\overline{READY}$
18	A_{13}	41	D_{10}	64	HOLD
19	A_{12}	42	D_3	65	$\overline{HLDA}$
20	A_{11}	43	D_{11}	66	$\overline{COD/INTA}$
21	A_{10}	44	D_4	67	$\overline{M/IO}$
22	A_9	45	D_{12}	68	$\overline{LOCK}$
23	A_8	46	D_5		

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b .Package Type
c. Temperature Range
d. Speed Option
e. Optional Processing



Valid Combinations	
80C286-25 80C286-20 80C286-16 80C286-12	R, N

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

CLK

System Clock (Input; Active HIGH)

System Clock provides the fundamental timing for 80C286 systems. It is divided by two inside the 80C286 to generate the processor clock. The internal divide-by-two circuitry can be synchronized to an external clock generator by LOW-to-HIGH transition on the RESET input.

D_0-D_{15}

Data Bus (Input/Output; Active HIGH)

Data Bus inputs data during memory, I/O, and interrupt acknowledge read cycles; outputs data during memory and I/O write cycles. The data bus is active HIGH and floats to three-state OFF during bus hold acknowledge.

A23-A0

Address Bus (Output; Active HIGH)

Address Bus outputs physical memory and I/O port addresses. A_0 is LOW when data is to be transferred on pins D_{7-0}. A_{23}-A_{16} are LOW during I/O transfers. The address bus is active HIGH and floats to three-state OFF during bus hold acknowledge.

$\overline{BHE}$

Bus High Enable (Output; Active LOW)

Bus High Enable indicates transfer of data on the upper byte of the data bus D_{15-8}. Eight-bit oriented devices assigned to the upper byte of the data bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is active LOW and floats to three-state OFF during bus hold acknowledge.

$\overline{BHE}$ and A_0 Encodings

$\overline{BHE}$ Value	A_0 Value	Function
0	0	Word transfer
0	1	Byte transfer on upper half of data bus (D_{15-8})
1	0	Byte transfer on lower half of data bus (D_{7-0})
1	1	Reserved

$\overline{S1}$, $\overline{S0}$

Bus Cycle Status (Output; Active LOW)

Bus Cycle Status indicates initiation of a bus cycle and, along with M/$\overline{IO}$ and COD/$\overline{INTA}$, defines the type of bus cycle. The bus is in a Ts state whenever one or both are LOW. $\overline{S1}$ and $\overline{S0}$ are active LOW and float to three-state OFF during bus hold acknowledge.

80C286 Bus Cycle Status Definition

COD/$\overline{INTA}$	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Bus cycle initiated
0 (LOW)	0	0	0	Interrupt acknowledge
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	None; not a status cycle
0	1	0	0	IF $A_1 = 1$ then halt; else shutdown
0	1	0	1	Memory data read
0	1	1	0	Memory data write
0	1	1	1	None; not a status cycle
1 (HIGH)	0	0	0	Reserved
1	0	0	1	I/O Read
1	0	1	0	I/O Write
1	0	1	0	None; not a status cycle
1	1	0	0	Reserved
1	1	0	1	Memory instruction read
1	1	1	0	Reserved
1	1	1	1	None; not a status cycle

M/$\overline{IO}$

Memory/IO Select (Output)

Memory/IO Select distinguishes memory access from I/O access. If HIGH during Ts, a memory cycle or a halt/shutdown cycle is in progress. If LOW, an I/O cycle or an interrupt acknowledge cycle is in progress M/$\overline{IO}$ floats to three-state OFF during bus hold acknowledge.

COD/$\overline{INTA}$

Code/Interrupt Acknowledge (Output)

Code/Interrupt Acknowledge distinguishes instruction fetch cycles from memory data read cycles. Also distinguishes interrupt acknowledge cycles from I/O cycles. COD/$\overline{INTA}$ floats to three-state OFF during bus hold acknowledge.

$\overline{LOCK}$

Bus Lock (Output; Active LOW)

Bus Lock indicates that other system bus masters are not to gain control of the system bus following the current bus cycle. The $\overline{LOCK}$ signal may be activated explicitly by the "LOCK" instruction prefix or automatically by 80C286 hardware during memory XCHG instructions, interrupt acknowledge, or descriptor table access. $\overline{LOCK}$ is active LOW and floats to three-state OFF during hold acknowledge.

PIN DESCRIPTION (continued)

READY

Bus Ready (Input; Active LOW)

Bus Ready terminates a bus cycle. Bus cycles are extended without limit until terminated by $\overline{\text{READY}}$ LOW. $\overline{\text{READY}}$ is an active LOW synchronous input requiring set-up and hold times relative to the system clock be met for correct operation. $\overline{\text{READY}}$ is ignored during bus hold acknowledge.

HOLD, HLDA

Bus Hold Request and Hold Acknowledge (Input/Output; Active HIGH)

Bus Hold Request and Hold Acknowledge control ownership of the 80C286 local bus. The HOLD input allows another local bus master to request control of the local bus. When control is granted, the 80C286 will float its bus drivers to three-state OFF and then active HLDA, thus entering the bus hold acknowledge condition. The local bus will remain granted to the requesting master until HOLD becomes inactive which results in the 80C286 deactivating HLDA and regaining control of the local buys. This terminates the bus hold acknowledge condition. HOLD may be asynchronous to the system clock. These signals are active HIGH.

INTR

Interrupt Request (Input; Active HIGH)

Interrupt Request requests the 80C286 to suspend its current program execution and service a pending external request. Interrupt requests are masked whenever the interrupt enable bit in the flag word is cleared. When the 80C286 responds to an interrupt request, it performs two interrupt acknowledge bus cycles to read an 8-bit interrupt vector that identifies the source of the interrupt. To assure program interruption, INTR must remain active until the first interrupt acknowledge cycle is completed. INTR is sampled at the beginning of each processor cycle and must be active HIGH at least two processor cycles before the current instruction ends in order to interrupt before the next instruction. INTR is level sensitive, active HIGH, and may be asynchronous to the system clock.

NMI

Non-maskable Interrupt Request (Input; Active HIGH)

Non-maskable Interrupt Request interrupts the 80C286 with an internally supplied vector value of 2. No interrupt acknowledge cycles are performed. The interrupt enable bit in the 80C286 flag word does not affect this input. The NMI input is active HIGH, may be asynchronous to the system clock, and is edge triggered after internal synchronization. For proper recognition, the input must have been previously LOW for at least four system clock cycles and remain HIGH for at least four system clock cycles.

PEREQ, $\overline{\text{PEACK}}$

Processor Extension Operand Request and Acknowledge (Input/Output)

Processor Extension Operand Request and Acknowledge extends the memory management and protection capabilities of the 80C286 to processor extensions. The PEREQ input requests the 80C286 to perform a data operand transfer for a processor extension. The $\overline{\text{PEACK}}$ output signals the processor extension when the requested operand is being transferred. PEREQ is active HIGH and may be asynchronous to the system clock. $\overline{\text{PEACK}}$ is active LOW.

$\overline{\text{BUSY}}$, $\overline{\text{ERROR}}$

Processor Extension Busy and Error

Processor Extension Busy and Error indicate the operating condition of a processor extension to the 80C286. An active $\overline{\text{BUSY}}$ input stops 80C286 program execution on WAIT and some ESC instructions until $\overline{\text{BUSY}}$ becomes inactive (HIGH). The 80C286 may be interrupted while waiting for $\overline{\text{BUSY}}$ to become inactive. An active $\overline{\text{ERROR}}$ input causes the 80C286 to perform a processor extension interrupt when executing WAIT or some ESC instructions. These inputs are active LOW and may be asynchronous to the system clock.

RESET

System Reset (Input; Active HIGH)

System Reset clears the internal logic of the 80C286 and is active HIGH. The 80C286 may be reinitialized at any time with a LOW-to-HIGH transition on RESET which remains active for more than 16 system clock cycles. During RESET active, the output pins of the 80C286 enter the state shown below:

80C286 Pin State during Reset

Pin Value	Pin Names
1 (HIGH)	$\overline{S0}$, $\overline{S1}$, $\overline{\text{PEACK}}$, A_{23}-A_0, $\overline{\text{BHE}}$, $\overline{\text{LOCK}}$
0 (LOW)	M/$\overline{IO}$, COD/$\overline{INTA}$, HLDA
three-state OFF	D_{15}-D_0

Operation of the 80C286 begins after a HIGH-to-LOW transition on RESET. The HIGH-to-LOW transition of RESET must be synchronous to the system clock. Approximately 50 system clock cycles are required by the 80C286 for internal initializations before the first bus cycle to fetch code from the power-on execution address is performed.

PIN DESCRIPTION (continued)

A LOW-to-HIGH transition of RESET synchronous to the system clock, will begin a new processor cycle at the next HIGH-to-LOW transition of the system clock. The LOW-to-HIGH transition of RESET may be asynchronous to the system clock; however, in this case it cannot be predetermined which phase of the processor clock will occur during the next system period. Synchronous LOW-to-HIGH transitions of RESET are only required for systems where the processor clock must be phase synchronous to another clock.

V_{SS}
System Ground (Input; Active HIGH)
System Ground: 0 volts.

V_{CC}
System Power (Input; Active HIGH)
System Power: +5 volt power supply.

FUNCTIONAL DESCRIPTION

Introduction

The 80C286 is a fully static advanced, high-performance microprocessor with specially optimized capabilities for multiple user and multitasking systems. Depending on the application, the 80C286's performance is up to 20 times faster than the standard 5-MHz 8086, while providing complete upward software compatibility with AMD's iAPX 86, 88, and 186 family of CPUs.

Static Operation

AMD's 80C286 is composed of complete static circuitry. Unlike the dynamic circuit design, the 80C286's internal registers, counters, and latches are static and do not require refresh which eliminates the minimum operating frequency restriction that is typically placed on microprocessors.

AMD's 80C286 can operate from DC to the specified upper frequency limit. The clock to the processor may be stopped at any point (either phase one or phase two of the processor clock cycle) and held there indefinitely. Additionally, a significant decrease in power requirement occurs if the clock is stopped in phase two of the processor clock cycle. Details on clock relationships can be found in the Bus Operation section.

Note that the ability to stop the clock to processor is useful for system debug or power critical applications such as battery-powered laptop personal computers. The 80C286 can be single-stepped using only the CPU clock, and this state can be maintained as long as necessary. Single step clock operation allows for simple interface circuitry to provide critical information during system debug.

Static design allows very low frequency operation (down to DC). In a power critical situation, this can provide low power operation since 80C286 power dissipation is directly related to operating frequency. As the system frequency is reduced, so is the operating power until, ultimately, with the clock stopped in phase two of the processor clock cycle, the 80C286 power requirement is the standby current (5 mA maximum).

The 80C286 operates in two modes: iAPX 86 real address mode and protected virtual address mode. Both modes execute a superset of the iAPX 86 and 88 instruction set.

In iAPX 86 real address mode programs use real addresses with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80C286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16 megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. Both modes provide the same base instruction set, registers, and addressing modes.

The following pages describe first, the base 80C286 architecture common to both modes; second, iAPX 86 real address mode; and third, protected mode.

80C286 Base Architecture

The iAPX 86, 88, 286, and C286 CPU family all contain the same basic set of registers, instructions, and addressing modes. The 80C286 processor is upward compatible with the 8086, 8088, and 80186 CPUs.

Register Set

The 80C286 base architecture has fifteen registers as shown in Figure 1. These registers are grouped into the following four categories:

General Registers: Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used either in their entirety as 16-bit words or split into pairs of separate 8-bit registers.

Segment Registers: Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers: Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode determines the specific registers used for operand address calculations.

Status and Control Registers: Three 16-bit special purpose registers record or control certain aspect of the 80C286 processor state. These include the Instruction Pointer, which contains the offset address of the next sequential instruction to be executed.

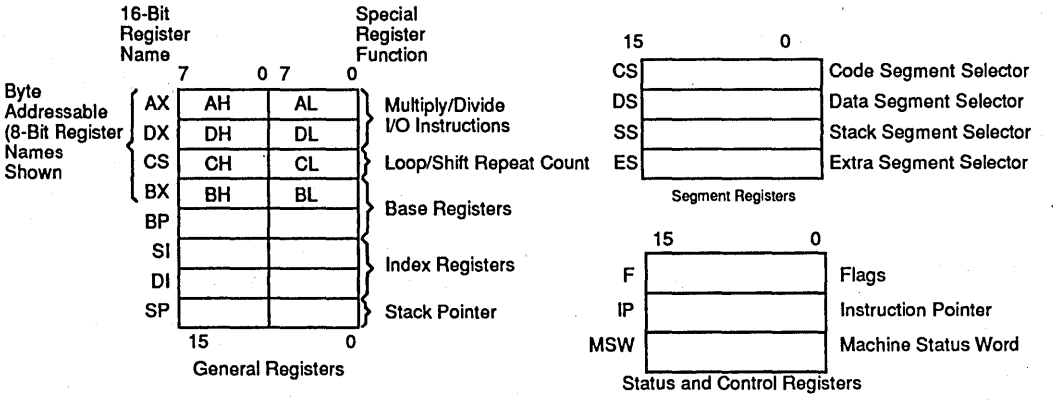


Figure 1. Register Set

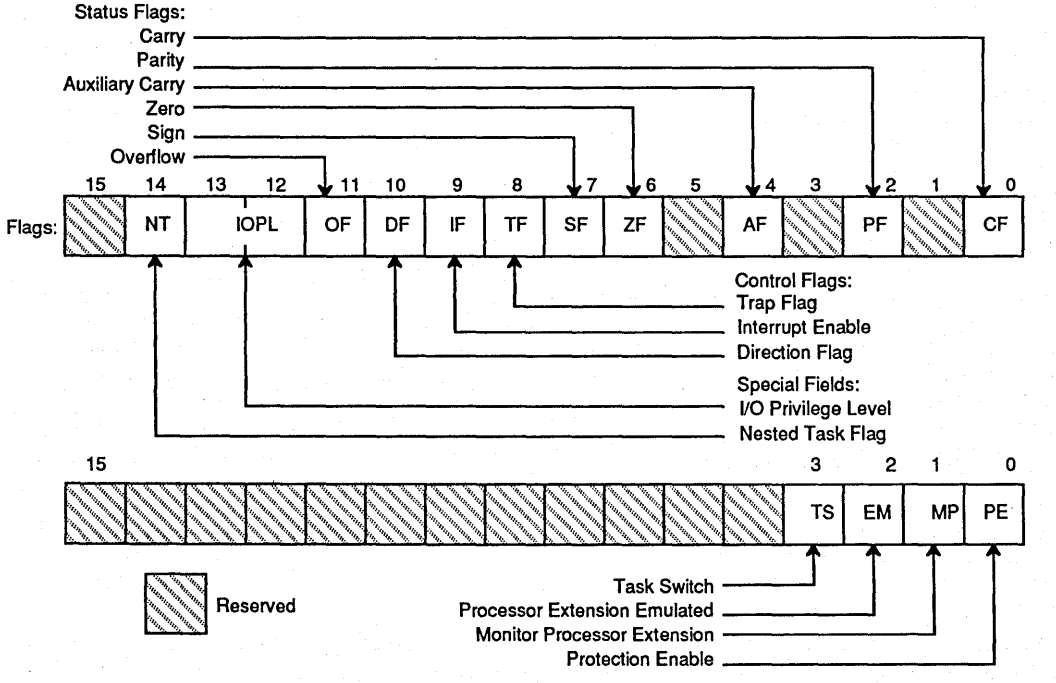


Figure 2. Status and Control Register Bit Functions

Flags Word Description

The Flags word (Flags) records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80C286 within a given operating mode (bits 8 and 9). Flags is a 16-bit register. The function of the flag bits is given in Table 1.

Table 1. Flags Word Bit Functions

Bit Position	Name	Function
0	CF	Carry Flag—Set on high-order bit carry or borrow; cleared otherwise
2	PF	Parity Flag—Set if low-order 8 bits of result contain an even number of 1 bits; cleared otherwise
4	AF	Set on carry-from or borrow-to the low-order four bits of AL; cleared otherwise
6	ZF	Zero Flag—Set if result is zero; cleared otherwise
7	SF	Sign Flag—Set equal to high-order bit of result (0 if positive, 1 if negative)
11	OF	Overflow Flag—Set if result is a too-large large positive number or a too-small negative number (excluding sign-bit) to fit in destination operand; cleared otherwise
8	TF	Single Step Flag—Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt
9	IF	Interrupt-Enable Flag—When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location
10	DF	Direction Flag—Causes string instructions to auto decrement the appropriate index registers when set. Clearing DF causes auto increment.

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, program transfer, high-level instructions, and processor control. These categories are summarized in Figures 3–9.

An 80C286 instruction can reference zero, one, or two operands where an operand resides in a register, in the instruction itself or in memory. Zero-operand instructions (e.g., NOP and HLT) are usually one byte long. One-operand instructions (e.g., INC and DEC) are usually two bytes long, but some are encoded in only one byte. One-operand instructions may reference a register or memory location. Two-operand instructions permit the following six types of instruction operations:

Register to Register
Memory to Register
Immediate to Register
Memory to Memory
Register to Memory
Immediate to Memory

Two-operand instructions (e.g., MOV and ADD) are usually three to six bytes long. Memory to memory operations are provided by a special class of string instructions requiring one to three bytes. For detailed instruction formats and encodings, refer to the instruction set summary at the end of this document.

Memory is organized as sets of variable length segments. Each segment is a linear contiguous sequence of up to 64K(^{2}16) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit segment selector and a 16-bit offset. The segment selector indicates the desired segment in memory. The offset component indicates the desired byte address within the segment.

Memory Organization

All instructions that address operands in memory must specify the segment and the offset. For speed and compact instruction encoding, segment selectors are usually stored in the high speed segment registers. An instruction need specify only the desired segment register and an offset to address a memory operand.

	General Purpose
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
	Input/Output
IN	Input byte or word
OUT	Output byte or word
	Address Object
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
	Flag Transfer
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

Figure 3. Data Transfer Instructions

	Addition
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
	Subtraction
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
	Multiplication
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
	Division
CBW	Convert byte to word
CWD	Convert word to doubleword

Figure 4. Arithmetic Instructions

MOVS	Move byte or word string
INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/REPNZ	Repeat while not equal/not zero

Figure 5. String Instructions

	Logicals
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
	Shifts
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
	Rotates
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

Figure 6. Shift/Rotate/Logical Instructions

Conditional Transfers		**Unconditional Transfers**	
JA/JNBE	Jump if above/not below nor equal	CALL	Call procedure
JAE/JNB	Jump if above or equal/not below	RET	Return from procedure
JB/JNAE	Jump if below/not above nor equal	JMP	Jump
JBE/JNA	Jump if below or equal/not above		
JC	Jump if carry	**Iteration Controls**	
JE/JZ	Jump if equal/zero		
JG/JNLE	Jump if greater/not less nor equal	LOOP	Loop
JGE/JNL	Jump if greater or equal/not less	LOOPE/LOOPZ	Loop if equal/zero
JL/JNGE	Jump if less/not greater nor equal	LOOPNE/LOOPNZ	Loop if not equal/not zero
JLE/JNG	Jump if less or equal/not greater	JCXZ	Jump if register CX = 0
JNC	Jump if not carry		
JNE/JNZ	Jump if not equal/not zero	**Interrupts**	
JNO	Jump if not overflow		
JNP/JPO	Jump if not parity/parity odd	INT	Interrupt
JNS	Jump if not sign	INTO	Interrupt if overflow
JO	Jump if overflow	IRET	Interrupt return
JP/JPE	Jump if parity/parity even		
JS	Jump if sign		

Figure 7. Program Transfer Instructions

	Flag Operations
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
	External Synchronization
HLT	Halt until interrupt or reset
WAIT	Wait for BUSY not active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
	No Operation
NOP	No operation
	Execution Environment Control
LMSW	Load machine status word
SMSW	Store machine status word

Figure 8. Processor Control Instructions

ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 9. High-Level Instructions

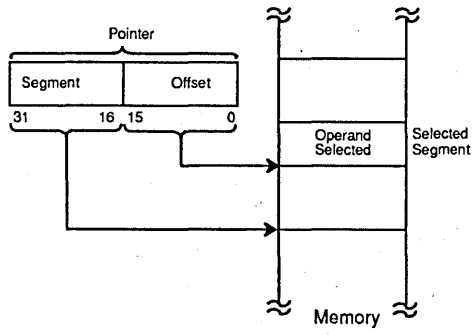


Figure 10. Two-Component Address

All instructions that address operands in memory must specify the segment and the offset. For speed and instruction encoding, segment selectors are usually stored in the high speed segment registers. An instruction need specify only the desired segment register and an offset to address a memory operand.

Most instructions need not explicitly specify which segment register is used. The correct segment register is automatically chosen according to the rules of Table 2. These rules follow the way programs are written (see Figure 11) as independent modules that require areas for code and data, a stack, and access to external data areas.

Memory Reference Needed	Segment Register Used	Selection Rule Selection Rule
Instructions	Code (CS)	Automatic with instruction prefetch
Stack	Stack (SS)	All stack pushes and pops. Any memory reference which uses BP as a base register.
Local Data	Data (DS)	All data references except when relative to stack or string destination.
External (Global) Data	Extra (ES)	Alternate data segment and destination of string operation.

Table 2. Segment Register Selection Rules

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs. To access operands that do not reside in one of the four immediately available segments, either a full 32-bit pointer can be used or a new segment selector must be loaded.

Addressing Modes

The 80C286 provides a total of eight addressing modes for instructions to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

Register Operand Mode: The operand is located in one of the 8- or 16-bit general registers.

Immediate Operand Mode: The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: segment selector and offset. The segment selector is supplied by a segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override

prefix. The offset is calculated by summing any combination of the following three address elements:

the displacement (an 8- or 16-bit immediate value contained in the instruction)

the base (contents of either the BX or BP base registers)

the index (contents of either the SI or DI index registers)

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, here described.

Direct Mode: The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.

Register Indirect Mode: The operand's offset is in one of the registers SI, DI, BX, or BP.

Based Mode: The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).

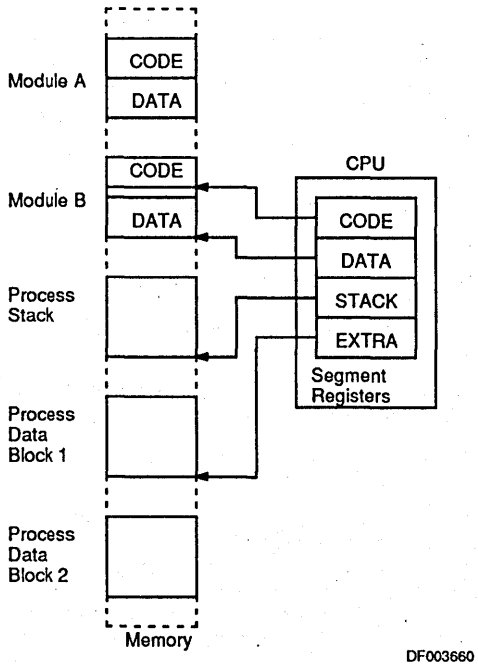


Figure 11. Segmented Memory Helps Structure Software

Indexed Mode: The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).

Based Indexed Mode: The operand's offset is the sum of the contents of a base register and an index register.

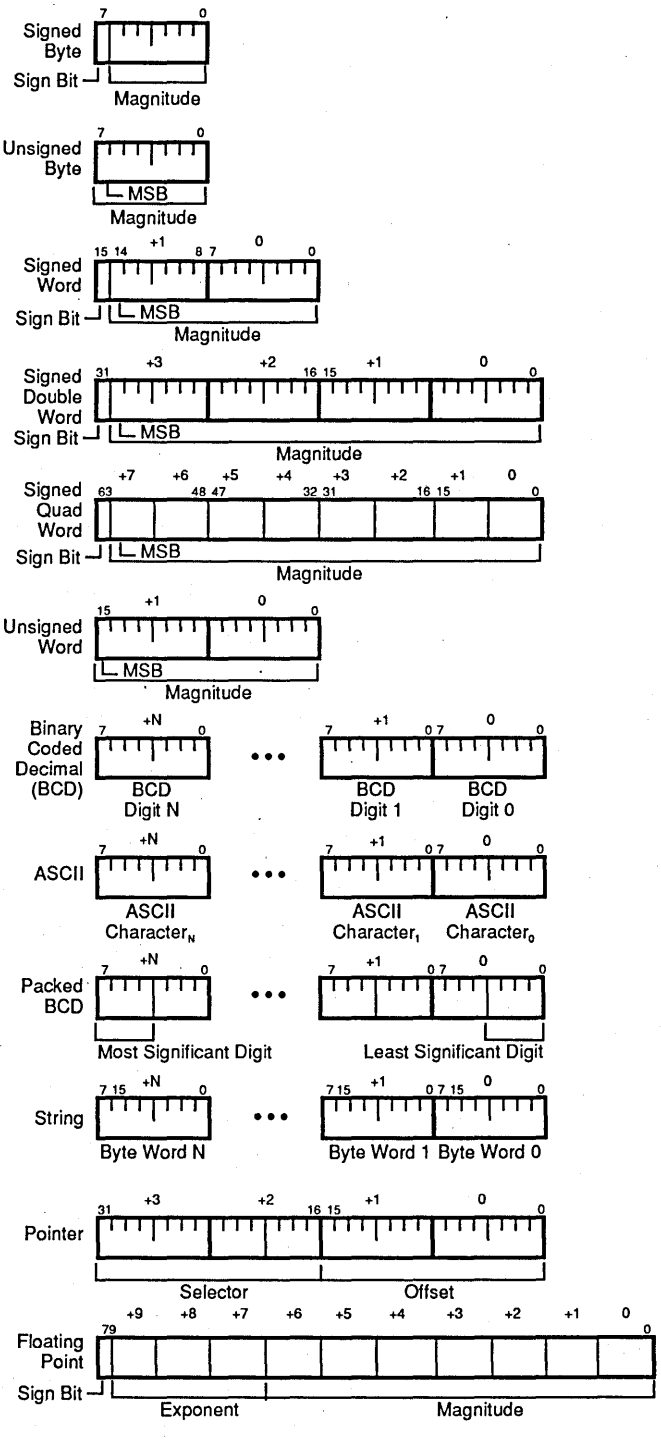
Based Indexed Mode with Displacement: The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

The 80C286 directly supports the following data types:

Integer:	A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a two's complement representation. signed 32- and 64-bit integers are supported using the 80C287 Numeric Data Processor.
Ordinal:	An unsigned binary numeric value contained in an 8-bit byte or 16-bit word.
Pointer:	A 32-bit quantity, composed of a segment selector component and an offset component. Each component is a 16-bit word.
String:	A contiguous sequence of bytes or words. A string may contain from 1 byte to 64K bytes.
ASCII:	A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
BCD:	A byte (unpacked) representation of the decimal digits 0-9.
Packed BCD:	A byte (packed) representation of two decimal digits 0-9 storing one digit in each nibble of the byte.
Floating Point:	A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using the 80C287 Numeric Processor configuration.)

Figure 12 graphically represents the data types supported by the 80C286.



*Supported by iAPX 80C286/80C287 Numeric Data Processor Configuration

DF003670

Figure 12. 80C286 Supported Data Types

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. I/O instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register. Eight-bit port addresses are zero extended such that A15-A8 are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.

Table 3. Interrupt Vector Assignments

Function	Interrupt Number	Related Instructions	Return Address Before Instruction Causing Exception?
Divide error exception	0	DIV, IDIV	Yes
Single step interrupt	1	All	
NMI interrupt	2	All	
Breakpoint interrupt	3	INT	
INTO detected overflow exception	4	INTO	No
BOUND range exceeded exception	5	BOUND	Yes
Invalid opcode exception	6	Any undefined opcode	Yes
Processor extension not available exception	7	ESC or WAIT	Yes
Reserved	8–15		
Processor extension error input	16	ESC or WAIT	
Reserved	17–31		
User defined	32–255		

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (flags) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable. Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. The return address from an exception will always point at the instruction causing the exception and include any leading instruction prefixes.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0-31, some of which are used for instruction exceptions, are reserved. For each interrupt, an 8-bit vector must be supplied to the 80C286 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware-initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Maskable Interrupt (INTR)

The 80C286 provides a maskable hardware interrupt request pin, INTR. Software enables this input by setting the interrupt flag bit (IF) in the flag word. All 224 user-defined interrupt sources can share this input, yet they can retain separate interrupt handlers. An 8-bit vector read by the CPU during the interrupt acknowledge sequence (discussed in System Interface section) identifies the source of the interrupt.

Further maskable interrupts are disabled while servicing an interrupt by resetting the IF but as part of the response to an interrupt or exception. The saved flag word will reflect the enable status of the processor prior to the interrupt. Until the flag word is restored to the flag register, the interrupt flag will be zero unless specifically set. The interrupt return instruction includes restoring the flag word, thereby restoring the original status of IF.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt input (NMI) is also provided. NMI has higher priority than INTR. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed.

While executing the NMI servicing procedure, the 80C286 will not service further NMI requests, INTR requests, or the processor extension segment overrun interrupt until an interrupt return (IRET) instruction is executed or the CPU is reset. If NMI occurs while currently servicing an NMI, its presence will be saved for servicing after executing the first IRET instruction. IF is cleared at the beginning of an NMI interrupt to inhibit INTR interrupts.

Single Step Interrupt

The 80C286 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single step interrupt and is controlled by the single step flag bit (TF) in the flag word. Once this bit is set, an internal single step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single stepped.

Interrupt Priorities

When simultaneous interrupt requests occur, they are processed in a fixed order as shown in Table 4. Interrupt processing involves saving the flags, return address, and setting CS:IP to point at the first instruction of the interrupt handler. If other interrupts remain enabled, they are processed before the first instruction of the current interrupt handler is executed. The last interrupt processed is therefore the first one serviced.

Table 4. Interrupt Processing Order

Order	Interrupt
1	INT instruction or exception
2	Single step
3	NMI
4	Processor extension segment overrun
5	INTR

Initialization and Processor Reset

Processor initialization or start up is accomplished by driving the RESET input pin HIGH. RESET forces the 80C286 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as RESET is active. After RESET becomes inactive and an internal processing interval elapses, the 80C286 begins execution in real address mode with the instruction at physical location FFFFF0(H). RESET also sets some registers to predefined values as shown in Table 5.

Table 5. 80C286 Initial Register State after RESET

Flag word	0002(H)
Machine Status Word	FFF0(H)
Instruction pointer	FFF0(H)
Code segment	F000(H)
Data segment	0000(H)
Extra segment	0000(H)
Stack segment	0000(H)

Machine Status Word Description

The machine status word (MSW) records when a task switch takes place and controls the operating mode of the 80C286. It is a 16-bit register of which the lower four bits are used. One bit places the CPU into protected mode, while the other three bits, as shown in Table 6, control the processor extension interface. After RESET, this register contains FFF0(H) which places the 80C286 in iAPX 86 real address mode.

Table 6. MSW Bit Functions

Bit Position	Name	Function
0	PE	Protected mode Enable places the 80C286 into protected mode and cannot be cleared except by RESET.
1	MP	Monitor Processor extension allows WAIT instructions to cause a processor extension not present exception (number 7).
2	EM	Emulate processor extension causes a processor extension not present exception (number 7) on ESC instructions to allow emulating a processor extension.
3	TS	Task Switched indicates the next instruction using a processor extension will cause exception 7, allowing software to test whether the current processor extension context belongs to the current task.

The LMSW and SMSW instructions can load and store the MSW in real address mode. The recommended use of TS, EM, and MP is shown in Table 7.

Table 7. Recommended MSW Encodings For Processor Extension Control

TS	MP	EM	Recommended Use	Instructions Causing Exception
0	0	0	iAPX 86 real address mode only. Initial encoding after RESET. 80C286 operation is identical to iAPX 86, 88.	None
0	0	1	No processor extension is available. Software will emulate its function.	ESC
1	0	1	No processor extension is available. Software will emulate its function. The current processor extension context may belong to another task.	ESC
0	1	0	A processor extension exists.	None
1	1	0	A processor extension exists. The current processor extension context may belong to another task. The exception (number 7) on WAIT allows software to test for an error pending from a previous processor extension operation.	ESC or WAIT

Halt

The HLT instruction stops program execution and prevents the CPU from using the local bus until restarted. Either NMI, INTR with IF = 1, or RESET will force the 80C286 out of halt. If interrupted, the saved CS:IP will point to the next instruction after the HLT.

iAPX 80C286 Real Address Mode

The 80C286 executes a fully upward-compatible superset of the 8086 instruction set in real address mode. In real address mode the 80C286 is object code compatible with 8086 and 8088 software. The real address mode architecture (registers and addressing modes) is exactly as described in the 80C286 Base Architecture section.

Memory Size

Physical memory is a contiguous array of up to 1,048,576 bytes (one megabyte) addressed by pins A0 through A19 and $\overline{BHE}$. A20 through A23 are ignored.

Memory Addressing

In real address mode the processor generates 20-bit physical addresses directly from a 20-bit segment base address and a 16-bit offset.

The selector portion of a pointer is interpreted as the upper 16 bits of a 20-bit segment address. The lower four bits of the 20-bit segment address are always zero. Segment addresses, therefore, begin on multiples of 16 bytes. See Figure 13 for a graphic representation of address formation.

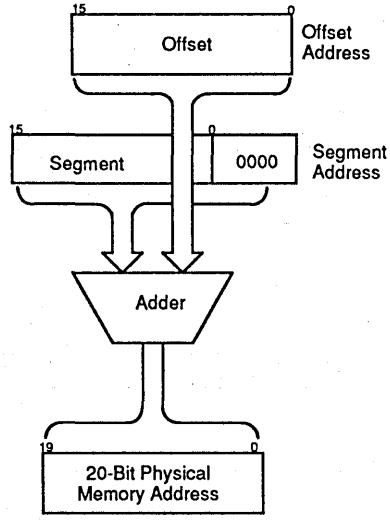


Figure 13. 80C286 Real Address Mode Address Calculation

All segments in real address mode are 64K bytes in size and may be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (e.g., a word with its low order byte at offset FFFF(H) and its high order byte at offset 0000(H). If, in real address mode, the information contained in a segment does not use the full 64K bytes, the unused end of the segment may be overlayed by another segment to reduce physical memory requirements.

Reserved Memory Locations

The 80C286 reserves two fixed areas of memory in real address mode (see Figure 14): system initialization area and interrupt table area. Locations from addresses FFFF0(H) through FFFFF(H) are reserved for system initialization. Initial execution begins at location FFFF0(H). Locations 00000(H) through 003FF(H) are reserved for interrupt vectors.

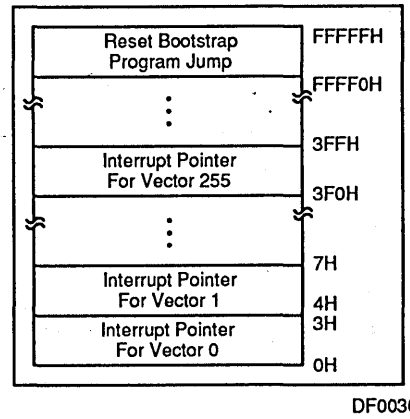


Figure 14. 80C286 Real Address Mode Initially Reserved Memory Locations

Table 8. Real Address Mode Addressing Interrupts

Function	Interrupt Number	Related Instructions	Return Address Before Instruction?
Interrupt table limit too small exception	8	INT vector not within table limit	Yes
Processor extension segment overrun interrupt	9	ESC with memory operand extending beyond offset FFFF(H)	No
Segment overrun exception	13	Word memory reference with offset = FFFF(H) or an attempt to execute past the end of a segment	Yes

Interrupts

Table 8 shows the interrupt vectors reserved for exceptions and interrupts which indicate an addressing error. The exceptions leave the CPU in the state existing before attempting to execute the failing instruction (except for PUSH, POP, PUSHA, or POPA). Refer to the next section on protected mode initialization for a discussion on exception 8.

Protected Mode Initialization

To prepare the 80C286 for protected mode, the LIDT instruction is used to load the 24-bit interrupt table base and 16-bit limit for the protected mode interrupt table. This instruction can also set a base and limit for the interrupt vector table in real address mode. After reset, the interrupt table base is initialized to 000000(H) and its size set to 03FF(H). These values are compatible with iAPX 86, 88 software. LIDT should only be executed in preparation for the protected mode.

Shutdown

Shutdown occurs when a severe error is detected that prevents further instruction processing by the CPU. Shutdown and halt are externally signaled via a halt bus operation. They can be distinguished by A1 HIGH for halt and A1 LOW for shutdown. In real address mode, shutdown can occur under two conditions:

- Exceptions 8 or 13 happen and the IDT limit does not include the interrupt vector.
- A CALL, INT, or POP instruction attempts to wrap around the stack segment when SP is not even.

An NMI input can bring the CPU out of shutdown if the IDT limit is at least 000F(H) and SP is greater than 0005(H); otherwise, shutdown can only be exited via the RESET input.

Protected Virtual Address Mode

The 80C286 executes a fully upward-compatible superset of the 8086 instruction set in protected virtual address mode (protected mode). Protected mode also provides memory management and protection mechanisms and associated instructions.

The 80C286 enters protected virtual address mode from real address mode by setting the PE (Protection Enable) bit of the machine status word with the Load Machine Status Word (LMSW) instruction. Protected mode offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

All registers, instructions, and addressing modes described in the 80C286 Base Architecture section remain the same. Programs for the iAPX 86, 88, 186, and real address model 80C286 can be run in protected mode; however, embedded constants for segment selectors are different.

Memory Size

The protected mode 80C286 provides a 1 gigabyte virtual address space per task mapped into a 16-megabyte physical address space defined by the address pin A23–A0 and $\overline{BHE}$. The virtual address space may be larger than the physical address space since any use of an address that does not map to a physical memory location will cause a restartable exception.

Memory Addressing

As in real address mode, protected mode uses 32-bit pointers, consisting of 16-bit selector and offset components. The selector, however, specifies an index into a memory-resident table rather than the upper 16-bits of a real memory address.

The 24-bit base address of the desired segment is obtained from the table in memory. The 16-bit offset is added to the segment base address to form the physical address as shown in Figure 15. The tables are automatically referenced by the CPU whenever a segment register is loaded with a selector. All 80286 instructions which load a segment register will reference the memory-based tables without additional software. The memory-based tables contain 8-byte values called descriptors.

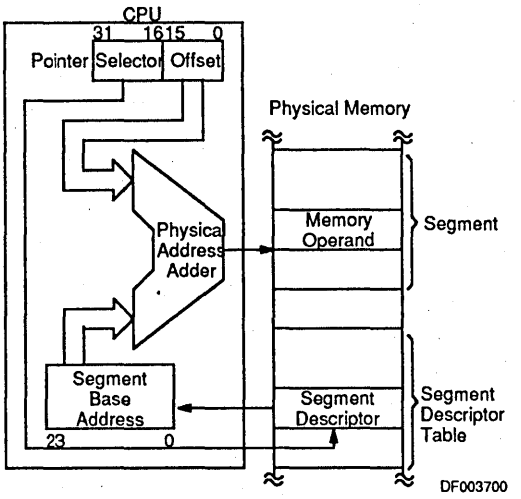


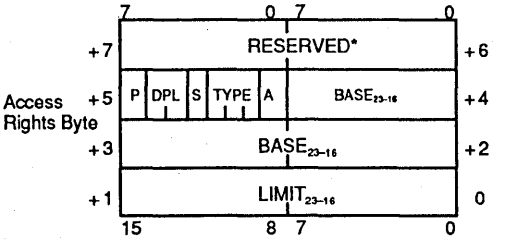
Figure 15. Protected Mode Memory Addressing

Descriptors

Descriptors defined the use of memory. Special types of descriptors also define new functions for transfer of control and task switching. The 80C286 has segment descriptors for code, stack and data segments, and system control descriptors for special system data segments and control transfer operations. Descriptor accesses are performed as locked bus operations to assure descriptor integrity in multi-processor systems.

Code and Data Segment Descriptors (S = 1)

Besides segment base addresses, code and data descriptors contain other segment attributes, including segment size (1 to 64K bytes), access rights (read-only, read/write, execute-only, and execute/read), and presence in memory (for virtual memory systems) (see Figure 16). Any segment usage violating a segment attribute indicate by the segment descriptor will prevent the memory cycle and cause an exception or interrupt.



*Must be set to 0 for compatibility with iAPX 386 and future upgrades

DF003710

Access Rights Byte Definition

	Bit Position	Name		Function	
Type Field Definition	7	Present (P)	P = 1	Segment is mapped into physical memory.	
			P = 0	No mapping to physical memory exists; base and limit are not used. Segment privilege attribute used in privilege tests.	
	6–5	Descriptor Privilege Level (DPL)			
	4	Segment Descriptor (S)	S = 1	Code or Data Segment descriptor	
			S = 0	Non-segment descriptor	
	3	Executable (E)	E = 0	Data segment descriptor type is:	Data Segment
	2	Expansion Direction (ED)	ED = 0	Grow up segment, offsets must be < limit.	Data Segment
			ED = 1	Grow down segment, offsets must be > limit.	Data Segment
	1	Writable (W)	W = 0	Data segment may not be written into.	Data Segment
			W = 1	Data segment may be written into.	Data Segment
	3	Executable (E)	E = 1	Code Segment Descriptor type is:	Code Segment
	2	Conforming (C)	C = 1	Code segment may only be executed when CPL > DPL.	Code Segment
	1	Readable (R)	R = 0	Code segment may not be read.	Code Segment
			R = 1	Code segment may be read.	Code Segment
	0	Accessed (A)	A = 0	Segment has not been accessed.	
			A = 1	Segment selector has been loaded into segment register or used by selector test instructions.	

Figure 16. Code and Data Segment Descriptors

Code and data are stored in two types of segments: code segments and data segments. Both types are identified and defined by segment descriptors. Code segments are identified by the executable (E) bit set to 1 in the descriptor access rights byte. The access rights byte of both code and data segment descriptor types have three fields in common: present (P) bit, Descriptor Privilege Level (DPL), and accessed (A) bit. If P = 0, any attempted use of this segment will cause a not-present exception. DPL specifies the privilege level of the segment descrip-

tor. DPL controls when the descriptor may be used by a task (refer to privilege discussion). The A bit shows whether the segment has been previously accessed for usage profiling, a necessity for virtual memory systems. The CPU will always set this bit when accessing the descriptor. Data segments (S = 1, E = 0) may be either read-only or read-write as controlled by the W bit of the access rights byte. Read-only (W = 0) data segments may not be written into.

Data segments may grow in two directions, as determined by the Expansion Direction (ED) bit; upward (ED = 0) for data segments, and downward (ED = 1) for a segment containing a stack. The limit field for a data segment descriptor is interpreted differently depending on the ED bit (see Figure 16).

A code segment (S = 1, E = 1) may be execute-only or execute/read as determined by the Readable (R) bit. Code segments may never be written into and execute-only code segments (R = 0) may not be read. A code segment may also have an attribute called Conforming (C). A conforming code segment may be shared by programs that execute at different privilege levels. The DPL of a conforming code segment defines the range of privilege levels at which the segment may be executed (refer to privilege discussion).

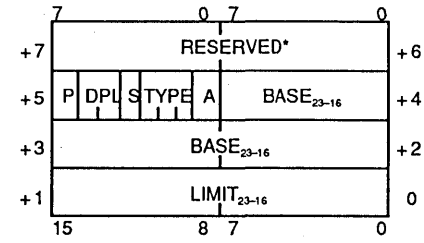
System Segment Descriptors

(S = 0, TYPE 1-3)

In addition to code and data segment descriptors, the protected mode 80C286 defines system segment descriptors. These descriptors define special system data segments which contain a table of descriptors (Local Descriptor Table Descriptor) or segments which contain the execution state of a task (Task State Segment Descriptor).

Figure 17 gives the formats for the special system data segment descriptors. The descriptors contain a 24-bit base address of the segment and a 16-bit limit. The access byte defines the type of descriptor, its state and privilege level. The descriptor contents are valid, and the segment is in physical memory if P = 1. If P = 0, the segment is not valid. The DPL field is only used in Task State Segment descriptors and indicates the privilege level at which the descriptor may be used (see Privilege). Since the Local Descriptor Table descriptor may only be used by a special privileged instruction, the DPL field is not used. Bit 4 of the access byte is 0 to indicate that it is a system control descriptor. The Type field specifies the descriptor type as indicated in Figure 17.

System Segment Descriptor



*Must be set to 0 for compatibility with iAPX 386 and future upgrades

TB0000880

System Segment Descriptor Fields

Name	Value	Description
TYPE	1	Available Task State Segment
	2	Local Descriptor Table Descriptor
	3	Busy Task State Segment
P	0	Descriptor contents are not valid
	1	Descriptor contents are valid
DPL	0–3	Descriptor Privilege Level
BASE	24-bit number	Base Address of special system data segment in real memory
LIMIT	16-bit number	Offset of last byte in segment

Figure 17. System Segment Format

Gate Descriptors

(S = 0, TYPE = 4-7)

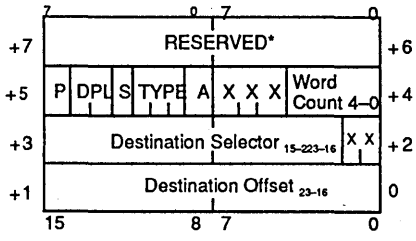
Gates are used to control access to entry points within the target code segment. The gate descriptors are call gates, task gates, interrupt gates and trap gates. Gates provide a level of indirection between the source and destination of the control transfer. This indirection allows the CPU to automatically perform protection checks and control the entry point of the destination. Call gates are used to change privilege levels (see Privilege); task gates are used to perform a task switch; and interrupt and trap gates are used to specify interrupt service routines. The interrupt gate disables interrupts (resets IF) while the trap gates does not.

Figure 18 shows the format of the gate descriptors. The descriptor contains a destination pointer that points to the descriptor of the target segment and the entry point offset. The destination selector in an interrupt gate, trap

gate, and call gate must refer to a code segment descriptor. These gate descriptors contain the entry point to prevent a program from constructing and using an illegal entry point. Task gates may only refer to a task state segment. Since task gates invoke a task switch, the destination offset is not used in the task gate.

Exception 13 is generated when the gate is used if a destination selector does not refer to the correct descriptor type. The Word Count field is used in the call gate descriptor to indicate the number of parameters (0–31 words) to be automatically copied from the caller's stack to the stack of the called routine when a control transfer changes privilege levels. The Word Count field is not used by any other gate descriptor.

Gate Descriptor



* Must be set to 0 for compatibility with iAPX 386 and future upgrades

TB0000880

Gate Descriptor Fields

Name	Value	Description
TYPE	4	–Call Gate
	5	–Task Gate
	6	–Interrupt Gate
	7	–Trap Gate
P	0	–Descriptor Contents are not valid
	1	–Descriptor Contents are valid
DPL	0–3	Descriptor Privilege Level
WORD COUNT	0–31	Number of words to copy from callers stack to called procedures stack. Only used with call gate.
DESTINATION SELECTOR	16-bit selector	Selector to the target code segment (Call, Interrupt or Trap Gate) Selector to the target task state segment (Task Gate)
DESTINATION OFFSET	16-bit offset	Entry point within the target code segment

Figure 18. Gate Descriptor Format

The access byte format is the same for all gate descriptors. P = 1 indicates that the gate contents are valid. P = 0 indicates the contents are not valid and causes exception 11 if referenced. DPL is the descriptor Privilege Level and specifies when this descriptor may be used by a task (refer to privilege discussion). Bit 4 must equal 0 to indicate a system control descriptor. The Type field specifies the descriptor type as indicated in Figure 18.

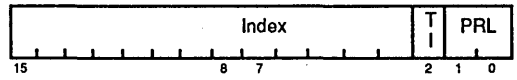
Segment Descriptor Cache Registers

A segment descriptor register is assigned to each of the four segment registers (CS, SS, DS, ES). Segment descriptors are automatically loaded (cached) into a segment descriptor cache register (Figure 20) whenever the associated segment register is loaded with a selector. Only segment descriptors may be loaded into segment descriptor cache registers. Once loaded, all references to that segment of memory use the cached descriptor information instead of reaccessing memory. The descriptor cache registers are not visible to programs. No instructions exist to store their contents. They only change when a segment register is loaded.

Selector Fields

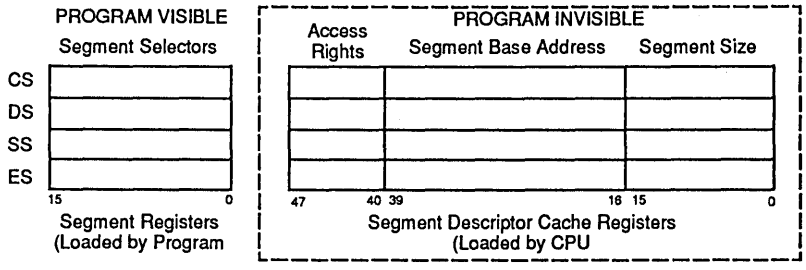
A protected mode selector has three fields: descriptor entry index, local or global descriptor table indicator (TI), and selector privilege (RPL), as shown in Figure 19. These fields select one of two memory-based tables of descriptors, select the appropriate table entry, and allow high-speed testing of the selector's privilege attribute (refer to privilege discussion).

Selector



Bits	Name	Function
1–0	Requested Privilege Level (RPL)	Indicates Selector Privilege Level Desired
2	Table Indicator (TI)	TI = 0 To use Global Descriptor Table (GDT) TI = 1 Use Local Descriptor Table (LDT)
15–3	Index	Select Descriptor Entry in Table

Figure 19. Selector Fields



DF003720

Figure 20. Descriptor Cache Registers

Local and Global Descriptor Tables

Two tables of descriptor tables, contain all descriptors accessible by a task at any given time. A descriptor table is a linear array of up to 8192 descriptors. The upper 13 bits of the selector value are an index into a descriptor table. Each table has a 24-bit base register to locate the descriptor table in physical memory and a 16-bit limit register that confines descriptor access to the defined limits of the table as shown in Figure 21. A restartable exception (13) will occur if an attempt is made to reference a descriptor outside the table limits.

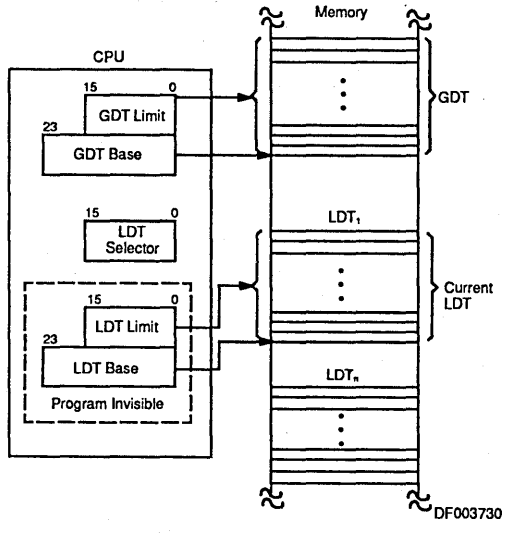


Figure 21. Local and Global Descriptor Table Definition

One table, called the Global Descriptor Table (GDT), contains descriptors available to all tasks. The other table, called the Local Descriptor Table (LDT), contains descriptors that can be private to a task. Each task may have its own private LDT. The GDT may contain all descriptor types except interrupt and trap descriptors. A segment cannot be accessed by a task if its segment descriptor does not exist in either descriptor table at the time of access.

The LGDT and LLDT instructions load the base and limit of the global and local descriptor tables. LGDT and LLDT are protected. They may only be executed by trusted programs operating at level 0. The LGDT instruction loads a six byte field containing the 16-bit table limit and 24-bit base address of the Global Descriptor Table as shown in Figure 22. The LLDT instruction loads a selector which refers to a descriptor in the Local Descriptor Table. This descriptor contains the base address and limit for an LDT, as shown in Figure 17.

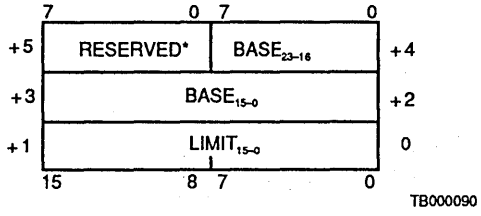


Figure 22. Global Descriptor Table and Interrupt Descriptor Data Type

* Must be set to 0 for compatibility with iAPX 386 and future upgrades.

Interrupt Descriptor Table

The protected mode 80C286 has a third descriptor table, called the Interrupt Descriptor Table (IDT) (see Figure 23), used to define up to 256 interrupts. It may contain only task gates, interrupt gates and trap gates. The IDT (Interrupt Descriptor Table) has a 24-bit base and 16-bit limit register in the CPU. The protected LIDT instruction loads these registers with a six-byte value of identical form to that of the LGDT instruction (see Figure 22 and Protected Mode Initialization).

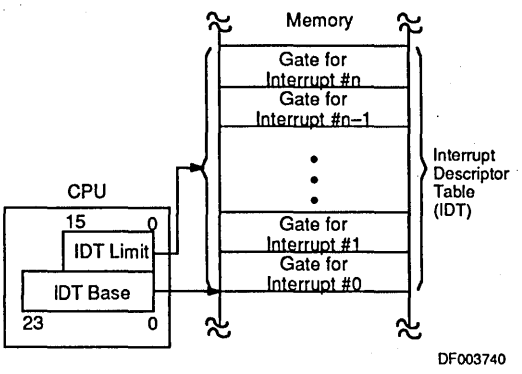


Figure 23. Local and Global Descriptor Table Definition

References to IDT entries are made via INT instructions, external interrupt vectors, or exceptions. The IDT must be at least 256 bytes in size to allocate space for all reserved interrupts.

Privilege

The 80C286 has a four-level hierarchical privilege system which controls the use of privileged instructions and access to descriptors (and their associated segments) within a task. Four-level privilege, as shown in Figure 24, is an extension of the user/supervisor mode commonly found in minicomputers. The privilege levels are numbered 0 through 3. Level 0 is the most privileged level. Privilege levels provide protection within a task. (Tasks are isolated by providing private LDT's for each task.) Operating system routines, interrupt handlers, and other system software can be included and protected within the virtual address space of each task using the four levels of privilege. Tasks may also have a separate stack for each privilege level.

Tasks, descriptors, and selectors have a privilege level attribute that determines whether the descriptor may be used. Task privilege effects the use of instructions and descriptors Descriptor and selector privilege only effect access to the descriptor.

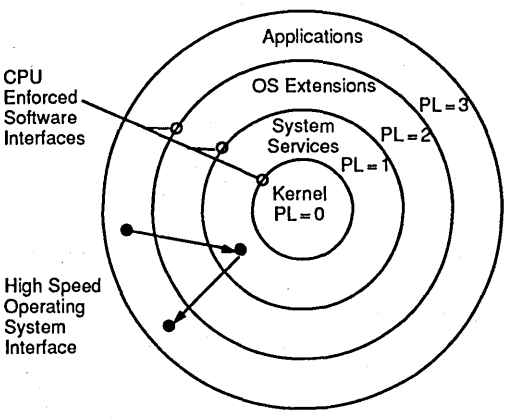


Figure 24. Hierarchical Privilege Levels

Task Privilege

The task always executes at one of the four privilege levels. A task privilege level at any specific instant is called the Current Privilege Level (CPL) and is defined by the lower two bits of the CS register. CPL cannot change during execution in a single code segment. A task's CPL may only be changed by control transfers through gate descriptors to a new code segment (See Control Transfer). Tasks begin executing at the CPL value specified by the code segment when the task is initiated via a task switch operation. A task executing at Level 0 can access all data segments defined in the GDT and the task's LDT and is considered the most trusted level. A task executed at Level 3 has the most restricted access to data and is considered the least trusted level.

Descriptor Privilege

Descriptor privilege is specified by the Descriptor Privilege Level (DPL) field of the descriptor access byte. DPL specifies the least trusted privilege level (CPL) at which a task may access the descriptor. Descriptors with DPL = 3 are the least protected (i.e. have the least restricted access) since tasks can access them when CPL = 0, 1, 2, or 3. This rule applies to all descriptors, except LDT descriptors.

Selector Privilege

Selector privilege is specified by the Requested Privilege Level (RPL) field in the least significant two bits of a selector. Selector RPL may establish a less trusted privilege level than the current privilege level for the use of a selector. This level is called the task's effective privilege level (EPL). RPL can only reduce the scope of a task's access to data with this selector. A task's effective privilege is the numeric maximum of RPL and CPL. A selector with RPL = 0 imposes no additional restriction on its use while a selector with RPL = 3 can only refer to segments at privilege Level 3 regardless of the task's CPL. RPL is generally used to verify that pointer parameters passed to a more trusted procedure are not allowed to

use data at a more privileged level than the caller (refer to pointer testing instructions).

Descriptor Access and Privilege Validation

Determining the ability of a task to access a segment involves the type of segment to be accessed, the instruction used, the type of descriptor used and CPL, RPL, and DPL. The two basic types of segment accesses are control transfer (selectors loaded into CS) and data (selectors loaded into DS, ES, or SS).

Data Segment Access

Instructions that load selectors into DS and ES must refer to a data segment descriptor or readable code segment descriptor. The CPL of the task and the RPL of the selector must be the same as or more privileged (numerically equal to or lower than) than the descriptor DPL. In general, a task can only access data segments at the same or less privileged levels than the CPL or RPL (whichever is numberically higher) to prevent a program from accessing data it cannot be trusted to use.

An exception to the rule is a readable conforming code segment. This type of code segment can be read from any privilege level.

If the privilege checks fail (e.g. DPL is numerically less than the maximum of CPL and RPL) or an incorrect type of descriptor is referenced (e.g. gate descriptor or execute only code segment), exception 13 occurs. If the segment is not present, exception 11 is generated.

Instructions that load selectors into SS must refer to data segment descriptors for writable data segments. The descriptor privilege (DPL) and RPL must equal CPL. All other descriptor types of privilege level violation will cause exception 13. A not present fault causes exception 12.

Control Transfer

Four types of control transfer can occur when a selector is loaded into CS by a control transfer operation (see Table 10). Each transfer type can only occur if the operation which loaded the selector references the correct descriptor type. Any violation of these descriptor usage rules (e.g. JMP through a call gate or RET to a Task State Segment) will cause exception 13.

The ability to reference a descriptor for control transfer is also subject to rules of privilege. A CALL or JUMP instruction may only reference a code segment descriptor with DPL equal to the task CPL or a conforming segment with DPL of equal or greater privilege than CPL. The RPL of the selector used to reference the code descriptor must have as much privilege as CPL.

RET and IRET instructions may only reference code segment descriptors with descriptor privilege equal to or less privileged than the task CPL. The selector loaded into CS is the return address from the stack. After the return, the selector RPL is the task's new CPL. If CPL changes, the old stack pointer is popped after the return address.

When a JMP or CALL references a Task State Segment descriptor, the descriptor DPL must be the same or less privileged than the task's CPL. Reference to a valid Task State Segment descriptor causes a task switch (see Task Switch Operation). Reference to a Task State Segment descriptor at a more privileged level than the task's CPL generates exception 13.

When an instruction or interrupt references a gate descriptor, the gate DPL must have the same or less privilege than the task CPL. If DPL is at a more privileged level than CPL, exception 13 occurs. If the destination selector contained in the gate references a code segment descriptor, the code segment descriptor DPL must be the same or more privileged than the task CPL. If not, Exception 13 is issued. After the control transfer, the code segment descriptor DPL is the task's new CPL. If the destination selector in the gate references a task state segment, a task switch is automatically performed (see Task Switch Operation).

The privilege rules on control transfer require:

– JMP or CALL direct to a code segment (code segment descriptor) can only be to a conforming segment with DPL of equal or greater privilege than CPL or a non-conforming segment at the same privilege level.

– interrupts within the task or calls that may change privilege levels can only transfer control through a gate at the same or a less privileged level than CPL to a code segment at the same or more privileged level than CPL.

– return instructions that don't switch tasks can only return control through a gate at the same or a less privileged level than CPL to a code segment at the same or more privileged level than CPL.

–return instructions that don't switch tasks can only return control to a code segment at the same or less privileged level.

– task switch can be performed by a call, a jump or an interrupt which references either a task gate or task state segment at the same or less privileged level.

Table 9. Descriptor Types Used for Control Transfer

Control Transfer Types	Descriptor Operation Types	Descriptor Referenced	Table
Intersegment within the same privilege level	JMP, CALL, RET, IRET*	Code Segment	GDT/LDT
Intersegment to the same or higher privilege level	CALL	Call Gate	GDT/LDT
Interrupt within task may change CPL	Interrupt Instruction, Exception, External Interrupt	Trap or Interrupt Gate	IDT
Intersegment to a lower privilege level (changes task CPL)	RET, IRET*	Code Segment	GDT/LDT
Task Switch	CALL, JMP	Task State Segment	GDT
	CALL, JMP	Task Gate	GDT/LDT
	IRET**		
	Interrupt Instruction, Exception, External Interrupt	Task Gate	IDT

* NT (Nested Task bit of flag word) = 0
** NT (Nested Task bit of flag word) = 1

Privilege Level Changes

Any control transfer that changes CPL within the task causes a change of stacks as part of the operation. Initial values of SS:SP for privilege levels 0, 1, and 2 are kept in the task state segment (refer to Task Switch Operation). During a JMP or CALL control transfer, the new stack pointer is loaded into the SS and SP registers and the previous stack pointer is pushed onto the new stack.

When returning to the original privilege level, its stack is restored as part of the RET or IRET instruction operation. For subroutine calls that pass parameters on the stack and cross privilege levels, a fixed number of words, as specified in the gate, are copied from the previous stack to the current stack. The intersegment RET instruction with a stack adjustment value will correctly restore the previous stack pointer upon return.

Protection

The 80C286 includes mechanisms to protect critical instructions that affect the CPU execution state (e.g., HLT) and code or data segments from improper usage. These mechanisms are grouped under the term "protection" and have three forms:

- Restricted usage of segments (e.g., no write allowed to read-only data segments). The only segments available for use are defined by descriptors in the Local Descriptor Table (LDT) and Global Descriptor Table (GDT).
- Restricted access to segments via the rules of privilege and descriptor usage.
- Privileged instructions or operations that may only be executed at certain privilege levels as determined by the CPL and I/O Privilege Level (IOPL). The IOPL is defined by bits 14 and 13 of the flag word.

These checks are performed for all instructions and can be split into three categories: segment load checks (Table 10), operand reference checks (Table 11), and privileged instruction checks (Table 12). Any violation of the rules shown will result in an exception. A not-present exception related to the stack segment causes exception 12.

The IRET and POPF instructions do not perform some of their defined functions if CPL is not of sufficient privilege (numerically small enough). Precisely these are:

- The IF bit is not changed if CPL > IOPL. The IOPL field of the flag word is not changed if CPL > 0.

No exceptions or other indication are given when these conditions occur.

Table 10. Segment Register Load Checks

Error Description	Exception Number
Descriptor table limit exceeded	13
Segment descriptor not-present	11 or 12
Privilege rules violated	13
Invalid descriptor/segment type segment register load: — Read only data segment load to SS — Special control descriptor load to DS, ES, SS — Execute only segment load to DS, ES, SS — Data segment load to CS — Read/Execute code segment load to SS	13

Table 11. Operand Reference Checks

Error Description	Exception Number
Write into code segment	13
Read from execute-only code segment	13
Write to read-only data segment	13
Segment limit exceeded[1]	12 or 13

Note: Carry out in offset calculations is ignored.

Table 12. Privileged Instruction Checks

Error Description	Exception Number
CPL ≠ 0 when executing the following instructions: LIDT, LLDT, LGDT, LTR, LMSW, CTS, HLT	13
CPL > IOPL when executing the following instructions INS, IN, OUTS, OUT, STI, CLI, LOCK	13

Exceptions

The 80C286 detects several types of exceptions and interrupts in protected mode (see Table 13). Most are restartable after the exceptional condition is removed. Interrupt handlers for most exceptions receive an error code, pushed on the stack after the return address, that identifies the selector involved (0 if none). The return address normally points to the failing instruction, including all leading prefixes. For a processor extension segment overrun exception, the return address will not point at the ESC instruction that caused the exception; however, the processor extension registers may contain the address of the failing instruction.

Table 13. Protected Mode Exceptions

Interrupt Vector	Function	Return Address At Failing Instruction?	Always Restartable?	Error Code on Stack?
8	Double exception detected	Yes	No[2]	Yes
9	Processor extension segment overrun	No	No[2]	No
10	Invalid task state segment	Yes	Yes	Yes
11	Segment not present	Yes	Yes	Yes
12	Stack segment overrun or segment not present	Yes	Yes[1]	Yes
13	General protection	Yes	No[2]	Yes

Notes 1. When a PUSHA or POPA instruction attempts to wrap around the stack segment, the machine state after the exception will not be restartable because stack segment wrap around is not permitted. This condition is identified by the value of the saved SP being either 0000(H), 0001(H), FFFE(H), or FFFF(H).

2. These exceptions indicate a violation to privilege rules or usage rules has occurred. Restart is generally not attempted under those conditions.

3. All these checks are performed for all instructions and can be split into three categories: Segment Load Checks (Table 10), Operand Reference Checks (Table 11), and Privileged Instruction Checks (Table 12). Any violation of the rules shown will result in an exception. A not-present exception causes exception 11 or 12 and is restartable.

Special Operations

Task Switch Operation

The 80C286 provides a built-in task switch operation which saves the entire 80C286 execution state (registers, address space, and a link to the previous task), loads a new execution state, and commences execution in the new task. Like gates, the task switch operation is invoked by executing an inter-segment JMP or CALL instruction which refers to a Task State Segment (TSS) or task gate descriptor in the GDT or LDT. An INT n instruction, exception, or external interrupt may also invoke the task switch operation by selecting a task gate descriptor in the associated IDT descriptor entry.

The TSS descriptor points at a segment (see Figure 25) containing the entire 80C286 execution state while a task gate descriptor contains a TSS selector. The limit field must be > 002B(H).

Each task must have a TSS associated with it. The current TSS is identified by a special register in the 80C286 called the Task Register (TR). This register contains a selector referring to the task state segment descriptor that defines the current TSS. A hidden base and limit register associated with TR are loaded whenever TR is loaded with a new selector.

The IRET instruction is used to return control to the task that called the current task or was interrupted. Bit 14 in the flag register is called the Nested Task(NT) bit. It controls the function of the IRET instruction. If NT = 0, the IRET instruction performs the regular current task return; when NT = 1, IRET performs a task switch operation back to the previous task.

When a CALL, JMP or INT instruction initiates a task switch, the old and new TSS will be marked busy and the back link field of the new TSS set to the old TSS selector. The NT bit of the new task is set by CALL or INT initiated task switches. An interrupt that does not cause a task switch will clear NT. NT may also be set or cleared by POPF or IRET instructions.

The task state segment is marked busy by changing the descriptor type field from Type 1 to Type 3. Use of a selector that references a busy task state segment causes Exception 13.

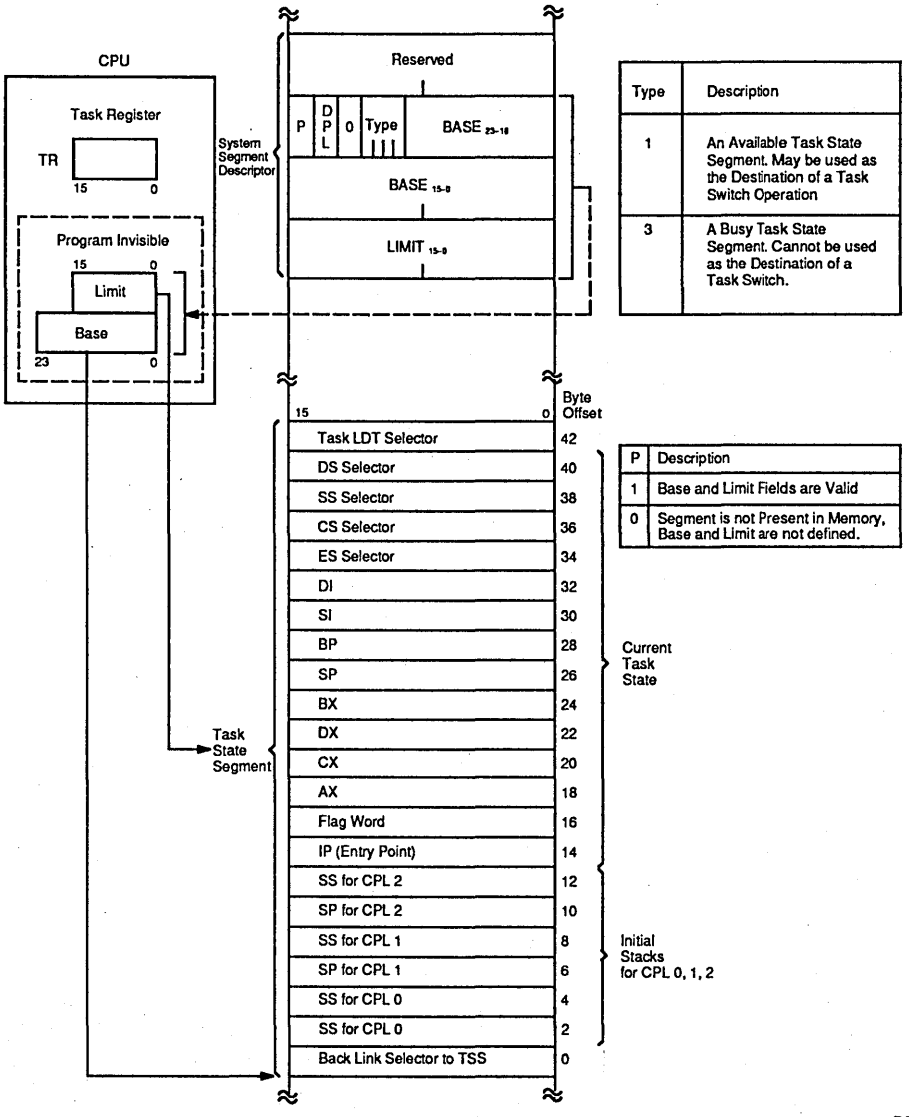
Processor Extension Context Switching

The context of a processor extension is not changed by the task switch operation. A processor extension context need only be changed when a different task attempts to use the processor extension (which still contains the context of a previous task). The 80C286 detects the first use of a processor extension after a task switch by causing the processor extension not present exception (7). The interrupt handler may then decide whether a context change is necessary.

Whenever the 80C286 switches tasks, it sets the Task Switched (TS) bit of the MSW. TS indicates that a processor extension context may belong to a different task than the current one. The processor extension not present exception (7) will occur when attempting to execute an ESC or WAIT instruction if TS = 1 and a processor extension is present (MP = 1 in MSW).

Pointer Testing Instructions

The 80C286 provides several instructions to speed pointer testing and consistency checks for maintaining system integrity (see Table 14). These instructions use the memory management hardware to verify that a selector value refers to an appropriate segment without risking an exception. A condition flag indicates whether use of the selector or segment will cause an exception.



DF003750

Figure 25. Task State Segment and TSS Registers

Table 14. Pointer Test Instructions

Instruction	Operands	Function
ARPL	Selector, Register	Adjust Requested Privilege Level: adjusts the RPL of the selector to the numeric maximum of current selector RPL value and the RPL value in the register. Set zero flag if selector RPL was changed.
VERR	Selector	VERify for Read: sets the zero flag is the segment referred to by the selector can be read.
VERW	Selector	VERify for Write: sets the zero flag if the segment referred to by the selector can be written.
LSL	Register, Selector	Load Segment Limit: reads the segment limit into the register if privilege rules and descriptor type allow. Set zero flag if successful.
LAR	Register, Selector	Load Access Rights: reads the descriptor access rights byte into the register if privilege rules allow. Set zero flag if successful.

Double Fault and Shutdown

If two separate exceptions are detected during a single instruction execution, the 80C286 performs the double fault exception (8). If an exception occurs during processing of the double fault exception, the 80C286 will enter shutdown. During shutdown no further instructions or exceptions are processed. Either NMI (CPU remains in protected mode) or RESET (CPU exits protected mode) can force the 80C286 out of shutdown. Shutdown is externally signalled via a HALT bus operation with A_1 HIGH.

Protected Mode Initialization

The 80C286 initially executes in real address mode after RESET. To allow initialization code to be placed at the top of physical memory, A_{23-20} will be HIGH when the 80C286 performs memory references relative to the CS register, until CS is changed. A_{23-20} will be zero for references to the DS, ES, or SS segments. Changing CS in real address mode will force A_{23-20} LOW whenever using CS thereafter. The initial CS:IP value of FFOO:FFFO provides 64K bytes of code space for initialization code without changing CS.

Before placing the 80C286 into protected mode, several registers must be initialized. The GDT and IDT base registers must refer to a valid GDT and IDT. After executing the LMSW instruction to set PE, the 80C286 must immediately execute an intrasegment JMP instruction to clear the instruction queue of instructions decoded in real address mode.

To force the 80C286 CPU registers to match the initial protected mode state assumed by software, execute a JMP instruction with a selector referring to the initial TSS used in the system. This will load the task register, local descriptor table register, segment registers and initial general register state. The TR should point at a valid TSS since a task switch operation involves saving the current task state.

System Interface

The 80C286 system interface appears in two forms: a local bus and a system bus. The local bus consists of address, data, status, and control signals at the pins of the CPU. A system bus is any buffered version of the local bus. A system bus may also differ from the local bus in terms of coding of status and control lines and/or timing and loading of signals. The 80C286 family includes several devices to generate standard system buses such as the IEEE 796 Standard MULTIBUS®.

Bus Interface Signals and Timing

The 80C286 microsystem local bus interfaces the 80286 to local memory and I/O components. The interface has 24 address lines, 16 data lines, and 8 status and control signals.

The 80C286 CPU, and peripheral chips performing the duties of clock generator, bus controller, bus arbiter, transceivers, and latches, provide a buffered and decoded system bus interface. A clock generator peripheral chip generates the system clock and synchronizes $\overline{\text{READY}}$ and $\overline{\text{RESET}}$. A bus controller peripheral chip converts bus operation status encoded by the 80C286 into command and bus control signals. These components can provide the timing and electrical power drive levels required for most system bus interfaces including the multibus.

Bus Hold Circuitry

Bus-hold circuitry has been used on 80C286 pins to avoid high current conditions that may be caused by floating inputs to CMOS devices. Figure 25A shows the circuit that will maintain the last valid logic state if no driving source is present (i.e., an unconnected pin or driving source that goes to a high impedance state). Figure 25B shows the circuit that will maintain a high impedance logic state if no driving source is present. In order to overdrive the "bus-hold" circuits, an external driver must be able to sink or source approximately 400 μAmps at valid input voltage levels. Since this bus-hold circuitry is active and not a resistive type element, the power supply current associated with it is negligible.

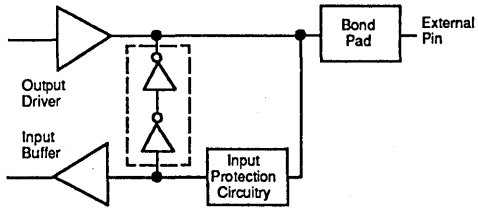


Figure 25A. Bus Hold Circuitry—Pins 36–51, 66, 67

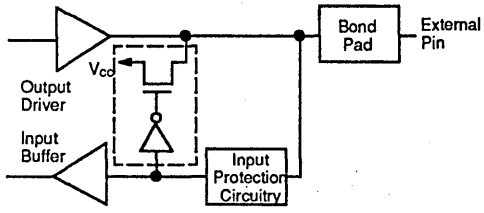


Figure 25B. Bus Hold Circuitry—Pins 4–6, 53, 54, 68

Physical Memory and I/O Interface

A maximum of 16 megabytes of physical memory can be addressed in protected mode. One megabyte can be addressed in real address mode. Memory is accessible as bytes or words. Words consist of any two consecutive bytes addressed with the least significant byte stored in the lowest address.

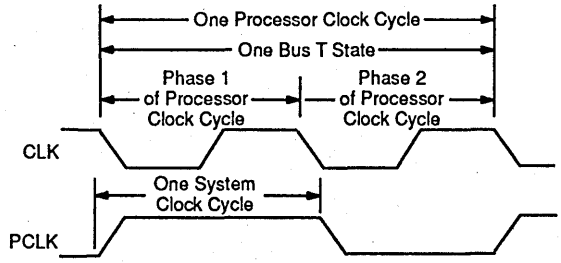
Byte transfers occur on either half of the 16-bit local data bus. Even bytes are accessed over D_{7-0} while odd bytes are transferred over D_{15-8}. Even-addressed words are transferred over D_{15-0} in one bus cycle, while odd-addressed words require two bus operations. The first transfers data on D_{15-8}, and the second transfers data on D_{7-0}. Both byte data transfers occur automatically, transparent to software.

Two bus signals, A_0 and $\overline{BHE}$, control transfers over the lower and upper halves of the data bus. Even address byte transfers are indicated by A_0 LOW and $\overline{BHE}$ HIGH. Odd address byte transfers are indicated by A_0 HIGH and $\overline{BHE}$ LOW. Both A_0 and $\overline{BHE}$ are LOW for even address word transfers.

The I/O address space contains 64K addresses in both modes. The I/O space is accessible as either bytes or words, as is memory. Byte wide peripheral devices may be attached to either the upper or lower byte of the data bus. Byte-wide I/O devices attached to the upper data byte (D_{15-8}) are accessed with odd I/O addresses. Devices on the lower data byte are accessed with even I/O addresses. An interrupt controller such as the 8259A must be connected to the lower data byte (D_{7-0}) for proper return of the interrupt vector.

Bus Operation

The 80C286 uses a double frequency system clock (CLK input) to control bus timing. All signals on the local bus are measured relative to the system CLK input. The CPU divides the system clock by 2 to produce the internal processor clock, which determines bus state. Each processor clock is composed of two system clock cycles named phase 1 and phase 2. The 82284 clock generator output (PCLK) identifies the next phase of the processor clock. (See Figure 26.)



AF003241

Figure 26. System and Processor Clock Relationships

Six types of bus operations are supported: memory read, memory write, I/O read, I/O write, interrupt acknowledge, and halt/shutdown. Data can be transferred at a maximum rate of one word per two processor clock cycles.

The 80C286 bus has three basic states: idle (T_i), send status (T_s), and perform command (T_c). The 80C286 CPU also has a fourth local bus state called hold (T_h). T_h indicates that the 80C286 has surrendered control of the local bus to another bus master in response to a HOLD request.

Each bus state is one processor clock long. Figure 27 shows the four 80C286 local bus states and allowed transitions.

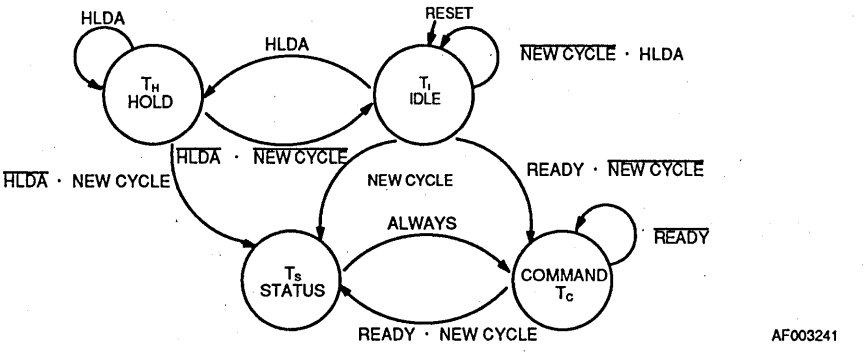


Figure 27. 80C286 Bus States

Bus States

The idle (T_I) state indicates that no data transfers are in progress or requested. The first active state, T_S, is signalled by either status line $\overline{S1}$ or $\overline{S0}$ going LOW also identifying phase 1 of the processor clock. During T_S, the command encoding, the address, and data (for a write operation) are available on the 80C286 output pins. The bus controller peripheral chip decodes the status signals and generates Multibus compatible read/write command and local transceiver control signals.

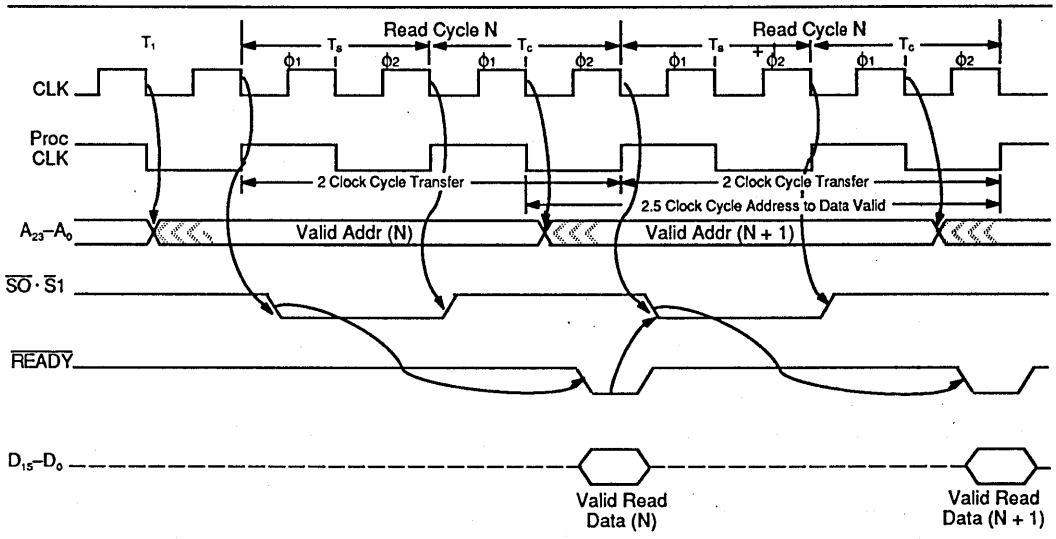
After T_S, the perform command (T_C) state is entered. Memory or I/O devices respond to the bus operation during T_C, either transferring read data to the CPU or accepting write data. T_C states may be repeated as often as necessary to assure sufficient time for the memory or I/O device to respond. The $\overline{READY}$ signal determines whether T_C is repeated. A repeated T_C state is called a wait state.

During hold (T_H), the 80C286 will float all address, data, and status output pins, enabling another bus master to use the local bus. The 80C286 HOLD input signal is used to place the 80C286 into the T_H state. The 80C286 HLDA output signal indicates that the CPU has entered T_H.

Pipelined Addressing

The 80C286 uses a local bus interface with pipelined timing to allow as much time as possible for data access. Pipelined timing allows bus operations to be performed in two processor cycles, while allowing each individual bus operation to last for three processor cycles.

The timing of the address outputs is pipelined such that the address of the next bus operation becomes available during the current bus operation. Or in other words, the first clock of the next bus operation is overlapped with the last clock of the current bus operation. Therefore, address decode and routing logic can operate in advance of the next bus operation. External address latches may hold the address stable for the entire bus operation and provide additional AC and DC buffering.



Pipelining: valid address (N + 1) available in last phase of bus cycle (N).

WF007840

Figure 28. Basic Bus Cycle

The 80C286 does not maintain the address of the current bus operation during all Tc states. Instead, the address for the next bus operation may be emitted during phase 2 of any Tc. The address remains valid during phase 1 of the first Tc to guarantee hold time, relative to ALE, for the address latch inputs.

Bus Control Signals

The bus controller peripheral chip provides control signals: address latch enable (ALE), Read/Write commands, data transmit/receive (DT/$\overline{R}$), and data enable (DEN) that control the address latches, data transceivers, write enable, and output enable for memory and I/O systems.

The Address Latch Enable (ALE) output determines when the address may be latched. ALE provides at least one system CLK period of address hold time from the end of the previous bus operation until the address for the next bus operation appears at the latch outputs. This address hold time is required to support Multibus and common memory systems.

The data bus transceivers are controlled by the bus controller peripheral chip outputs Data Enable (DEN) and Data Transmit/Receive (DT/$\overline{R}$). DEN enables the data transceivers while DT/$\overline{R}$ controls transceiver direction. DEN and DT/$\overline{R}$ are timed to prevent bus contention between the bus master, data bus transceivers, and system data bus transceivers.

Command Timing Controls

Two system timing customization options, command extension and command delay, are provided on the 80C286 local bus.

Command extension allows additional time for external devices to respond to a command and is analogous to inserting wait states on the 8086. External logic can control the duration of any bus operation such that the operation is only as long as necessary. The $\overline{READY}$ input signal can extend any bus operation for as long as necessary.

Command delay allows an increase of address or write data set-up time to system bus command active for any bus operation by delaying when the system bus command becomes active. Command delay is controlled by the bus controller peripheral chip CMDLY input. After Ts, the bus controller samples CMDLY at each failing edge of CLK. If CMDLY is HIGH, the bus controller peripheral chip will not activate the command signal. When CMDLY is LOW, the bus controller peripheral chip will activate the command signal. After the command becomes active, the CMDLY input is not sampled.

When a command is delayed, the available response time from command active to return read data or accept write data is less. To customize system bus timing, an address decoder can determine which bus operations require delaying the command. The CMDLY input does not affect the timing of ALE, DEN, or DT/$\overline{R}$.

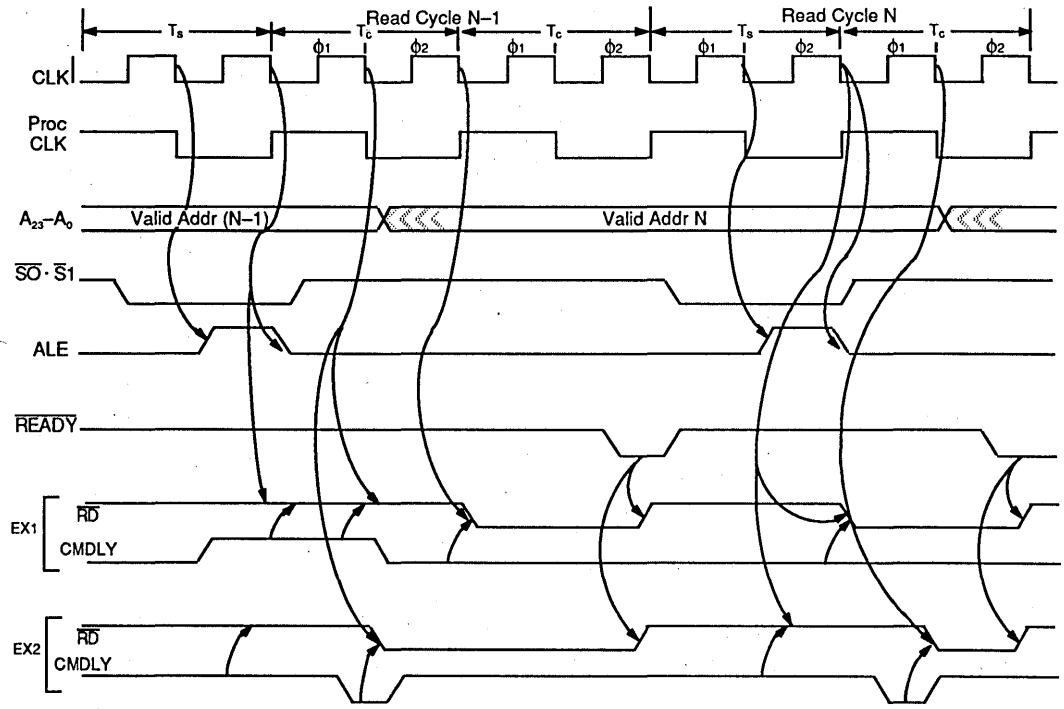


Figure 29. CMDLY Controls and Leading Edge of the Command

Figure 29 illustrates four uses of CMDLY. Example 1 shows delaying the read command two system CLKs for cycle N-1 and no delay for cycle N, and example 2 shows delaying the read command one system CLK for cycle N-1 and one system CLK delay for cycle N.

Bus Cycle Termination

At maximum transfer rates, the 80C286 bus alternates between the status and command states. The bus status signals become inactive after Ts so that they may correctly signal the start of the next bus operation after the completion of the current cycle. No external indication of Tc exists on the 80C286 local bus. The bus master and bus controller enter Tc directly after Ts and continue executing Tc cycles until terminated by $\overline{\text{READY}}$.

$\overline{\text{READY}}$ Operation

The current bus master and bus controller peripheral chip terminate each bus operation simultaneously to achieve maximum bus bandwidth. Both are informed in advance by $\overline{\text{READY}}$ active which identifies the last Tc cycle of the current bus operation. The bus master and bus controller must see the same sense of the $\overline{\text{READY}}$ signal, thereby requiring $\overline{\text{READY}}$ be synchronous to the system clock.

Synchronous Ready

The clock generator peripheral chip provides $\overline{\text{READY}}$ synchronization from both synchronous and asynchronous sources (See Figure 30). The synchronous ready input ($\overline{\text{SRDY}}$) of the clock generator is sampled with the falling edge of CLK at the end of phase 1 of each Tc. The state of $\overline{\text{SRDY}}$ is then broadcast to the bus master and bus controller via the $\overline{\text{READY}}$ output line.

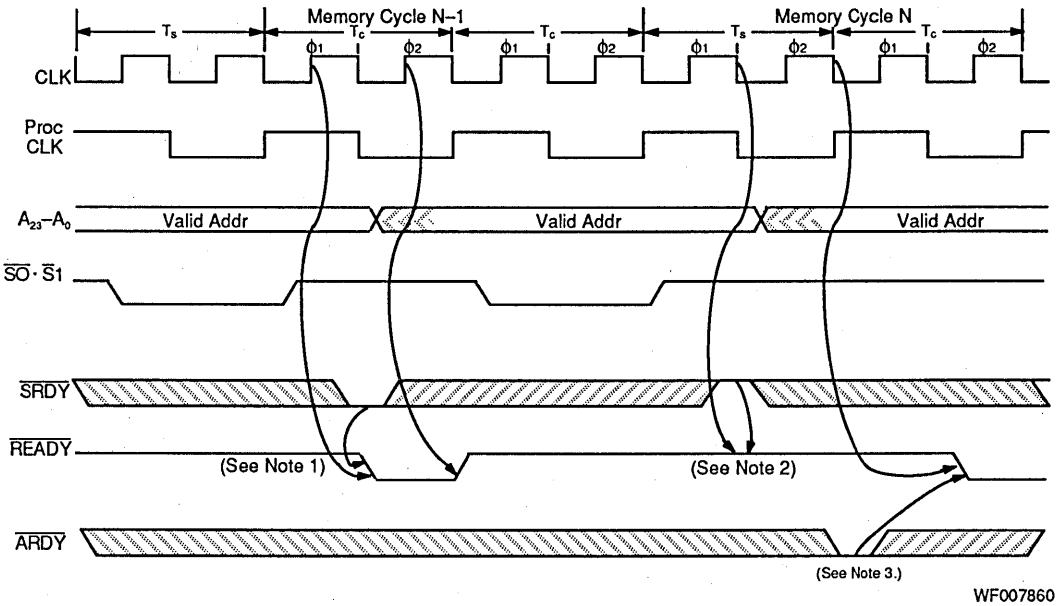


Figure 30. Synchronous and Asynchronous Ready

Notes: 1. $\overline{\text{SRDYEN}}$ is active LOW.
2. If $\overline{\text{SRDYEN}}$ is HIGH, the state of $\overline{\text{SRDY}}$ will not effect $\overline{\text{READY}}$.
3. $\overline{\text{ARDYEN}}$ is active LOW.

Asynchronous Ready

Many systems have devices or subsystems that are asynchronous to the system clock. As a result, their ready outputs cannot be guaranteed to meet the clock generator peripheral chip $\overline{\text{SRDY}}$ set-up and hold time requirements. The clock generator peripheral chip asynchronous ready input ($\overline{\text{ARDY}}$) is designed to accept such signals. The $\overline{\text{ARDY}}$ input is sampled at the beginning of each Tc cycle by 82284 synchronization logic. This provides a system CLK cycle time to resolve its value before broadcasting it to the bus master and bus controller.

$\overline{\text{ARDY}}$ or $\overline{\text{ARDYEN}}$ must be HIGH at the end of Ts. $\overline{\text{ARDY}}$ cannot be used to terminate bus cycle with no wait status.

Each ready input of 82284 has an enable pin ($\overline{\text{SRDYEN}}$ and $\overline{\text{ARDYEN}}$) to select whether the current bus operation will be terminated by the synchronous or asynchronous ready. Either of the ready inputs may terminate a bus operation. These enable inputs are active low and have the same timing as their respective ready inputs. Address decode logic usually selects whether the current bus operation should be terminated by $\overline{\text{ARDY}}$ or $\overline{\text{SRDY}}$.

Data Bus Control

Figures 31, 32, and 33 show how the DT/$\overline{\text{R}}$, DEN, data bus, and address signals operate for different combinations of read, write, and idle bus operations. DT/$\overline{\text{R}}$ goes active (LOW) for a read operation. DT/$\overline{\text{R}}$ remains HIGH before, during, and between write operations.

The data bus is driven with write data during the second phase of Ts. The delay in write data timing allows the read data drivers, from a previous read cycle, sufficient time to enter three-state OFF before the 80C286 begins driving the local data bus for write operations. Write data will always remain valid for one system clock past the last Tc to provide sufficient hold time for Multibus or other similar memory or I/O systems. During write-read or write-idle sequences the data bus enters three-state OFF during the second phase of the processor cycle after the last Tc. In a write-write sequence the data bus does not enter three-state OFF between Tc and Ts.

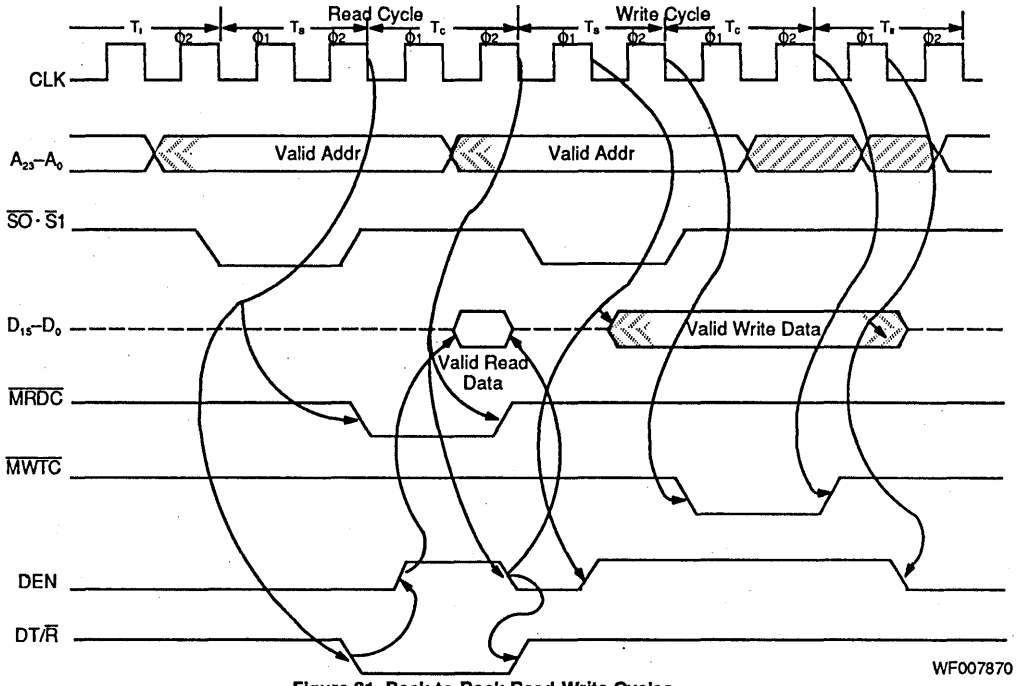


Figure 31. Back-to-Back Read-Write Cycles

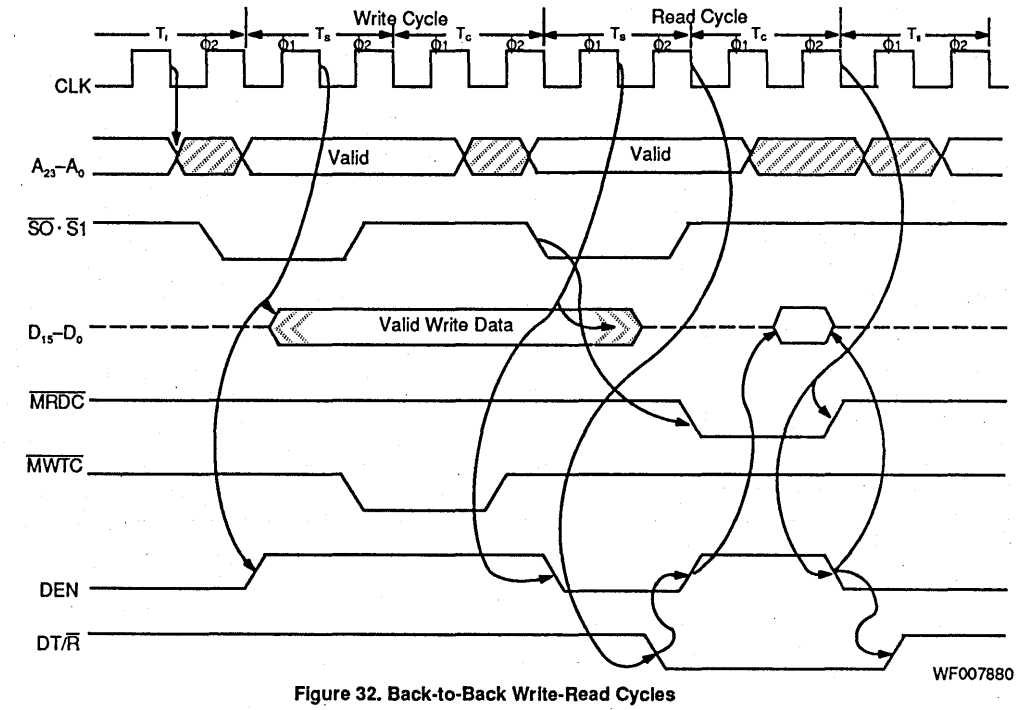


Figure 32. Back-to-Back Write-Read Cycles

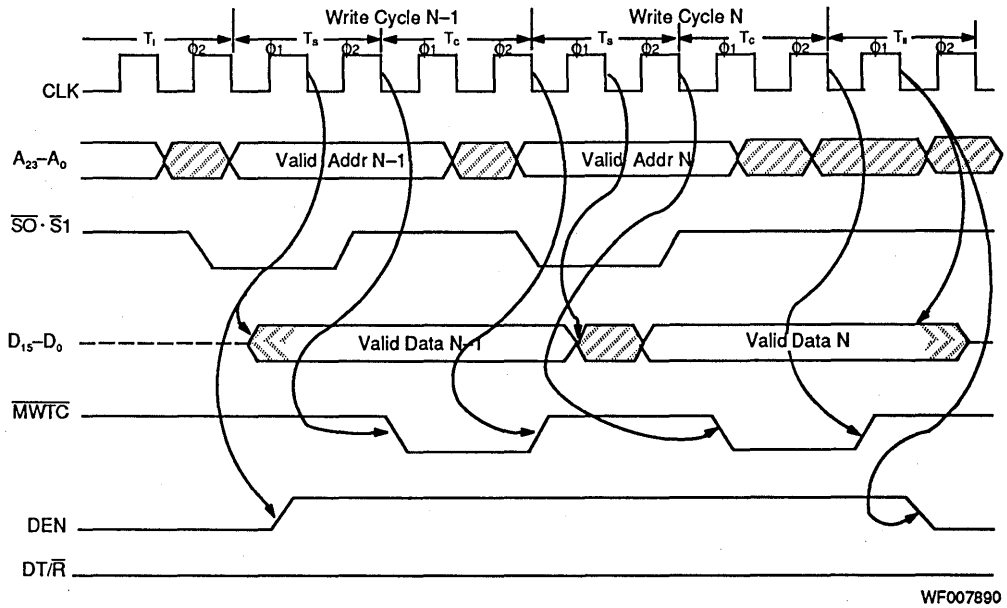


Figure 33. Back-to-Back Write-Write Cycles

Bus Usage

The 80C286 may be used for several functions: instruction data transfers, data transfers by other bus masters, instruction fetching, processor extension data transfers, interrupt acknowledge, and halt/shutdown. This section describes local bus activities which have special signals or requirements.

HOLD and HLDA

HOLD and HLDA allow another bus master to gain control of the local bus by placing the 80C286 bus into the Th state. The sequence of events required to pass control between the 80C286 and another local bus master are shown in Figure 34.

In this example, the 80C286 is initially in the T_h, state as signaled by HLDA being active. Upon leaving T_h, as signaled by HLDA going inactive, a write operation is started. During the write operation another local bus master requests the local bus from the 80C286 as shown by the HOLD signal. After completing the write operation, the 80C286 performs one T_i bus cycle, to guarantee write data hold time, then enters T_h as signaled by HLDA going active.

The CMDLY signal and $\overline{ARDY}$ ready are used to start and stop the write bus command, respectively. Note that $\overline{SRDY}$ must be inactive or disabled by $\overline{SRDYEN}$ to guarantee $\overline{ARDY}$ will terminate the cycle.

HOLD must not be active during the time from the leading edge of RESET until 34 CLKs following the trailing edge of RESET unless the 80C286 is in the Halt condition. To ensure that the 80C286 remains in the Halt condition until the processor Reset operation is complete, no interrupts should occur after the execution of HLT until 34 CLKs after the trailing edge of the RESET pulse.

Lock

The CPU asserts an active lock signal during Interrupt-Acknowledge cycles, the XCHG instruction, and during some descriptor accesses. Lock is also asserted when the LOCK prefix is used. The LOCK prefix may be used with the following ASM-286 assembly instructions; MOVS, INS, and OUTS. For bus cycles other than Interrupt-Acknowledge cycles, Lock will be active for the first and subsequent cycles of a series of cycles to be locked. Lock will not be shown active during the last cycle to be locked. For the next-to-last cycle, Lock will become inactive at the end of the first T_c regardless of the number of wait-states inserted. For Interrupt-Acknowledge cycles, Lock will be active for each cycle, and will become inactive at the end of the first T_c for each cycle regardless of the number of wait-states inserted.

Instruction Fetching

The 80C286 Bus Unit (BU) will fetch instructions ahead of the current instruction being executed. This activity is called prefetching. It occurs when the local bus would otherwise be idle and obeys the following rules:

A prefetch bus operation starts when at least two bytes of the 6-byte prefetch queue are empty.

The prefetcher normally performs word prefetches independent of the byte alignment of the code segment base in physical memory.

The prefetcher will perform only a byte code fetch operation for control transfers to an instruction beginning on a numerically odd physical address.

Prefetching stops whenever a control transfer or HLT instruction is decoded by the IU and placed into the instruction queue.

In real address mode, the prefetcher may fetch up to 5 bytes beyond the last control transfer or HLT instruction in a code segment.

In protected mode, the prefetcher will never cause a segment overrun exception. The prefetcher stops at the last physical memory word of the code segment. Exception 13 will occur if the program attempts to execute beyond the last full instruction in the code segment.

If the last byte of a code segment appears on an even physical memory address, the prefetcher will read the next physical byte of memory (perform a word code fetch). The value of this byte is ignored and any attempt to execute it causes exception 13.

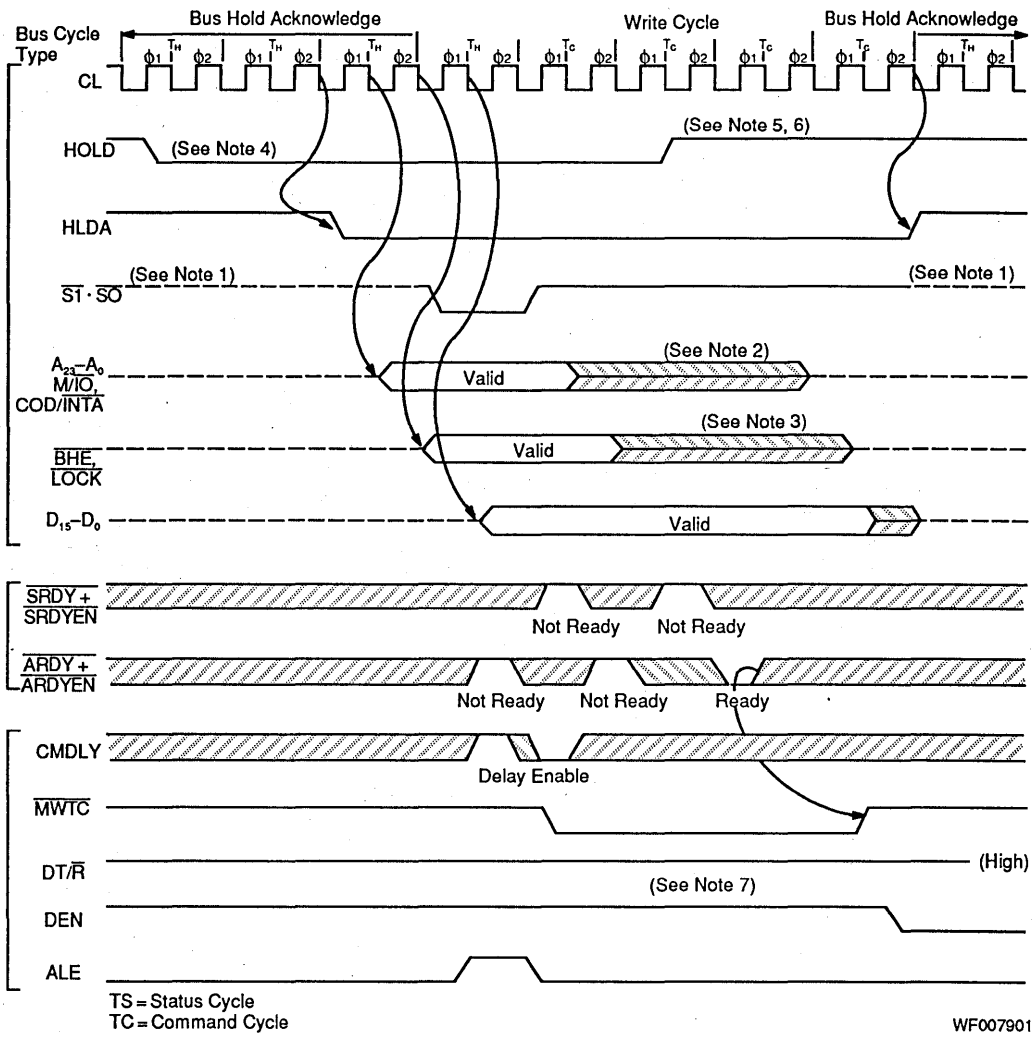


Figure 34. MULTIBUS Write Terminated by Asynchronous Ready with Bus Hold

Notes: 1. Status lines are not driven by 80C286, yet remain high due to pull-up resistors in the peripheral chips during HOLD state.

2. Address, M/IO and COD/INTA may start may start floating during any TC depending on when internal 80C286 bus arbiter decides to release bus to external HOLD. The float starts in Ø2 of TC.
3. BHE and LOCK may start floating after the end of any TC depending on when internal 80286 bus arbiter decides to release bus to external HOLD.
4. The minimum HOLD ↓ to HLDA ↓ time is shown. Maximum is one T_H longer.
5. The earliest HOLD ↑ time is shown which will always allow a subsequent memory cycle if pending.
6. The minimum HOLD ↑ to HLDA ↑ time is shown. Maximum is a function of the instruction, type of bus cycle and other machine status (i.e., Interrupts, Waits, Lock, etc.)
7. Asynchronous ready allows termination of the cycle. Synchronous ready does not signal ready in this example. Synchronous ready state is ignored after ready is signaled via the asynchronous input.

Processor Extension Transfers

The processor extension interface uses I/O port addresses 00F8(H), and 00FA(H), and 00FC(H) which are part of the I/O port address range and is a reserved area. An ESC instruction with EM = 0 and TS = 0 will perform I/O bus operations to one or more of these I/O port addresses independent of the value of IOPL and CPL.

ESC instructions with memory references enable the CPU to accept PEREQ inputs for processor extension operand transfers. The CPU will determine the operand starting address and read/write status of the instruction. For each operand transfer, two or three bus operations, one word transfer with I/O port address 00FA(H), and one or two bus operations with memory are performed. Three bus operations are required for each word operand aligned on an odd byte address.

Interrupt Acknowledge Sequence

Figure 35 illustrates an interrupt acknowledge sequence performed by the 80C286 in response to an INTR input. An interrupt acknowledge sequence consists of two INTA bus operations. The first allows a master Programmable Interrupt Controller (PIC) to determine which if any of its slaves should return the interrupt vector. An eight bit vector is read by the 80C286 during the second INTA bus operation to select an interrupt handler routine from the interrupt table.

The Master Cascade Enable (MCE) signal of the bus controller peripheral is used to enable the cascade address drivers, during INTA bus operations (see Figure 35), onto the local address bus for distribution to slave interrupt controllers via the system address bus. The 80C286 emits the $\overline{\text{LOCK}}$ signal (active LOW) during T_S of the first INTA bus operation. A local bus "hold" request will not be honored until the end of the second INTA bus operation.

Three idle processor clocks are provided by the 80C286 between INTA bus operations to allow for the minimum INTA to INTA time and CAS (cascade address) out delay of the interrupt controller peripheral. The second INTA bus operation must always have at least one extra T_C state added via logic controlling $\overline{\text{READY}}$. A_{23}–A_0 are in three-state OFF until after the first T_C state of the second INTA bus operation. This prevents bus contention between the cascade address drivers and CPU address drivers. The extra T_C state allows time for the 80C286 to resume driving the address lines for subsequent bus operations.

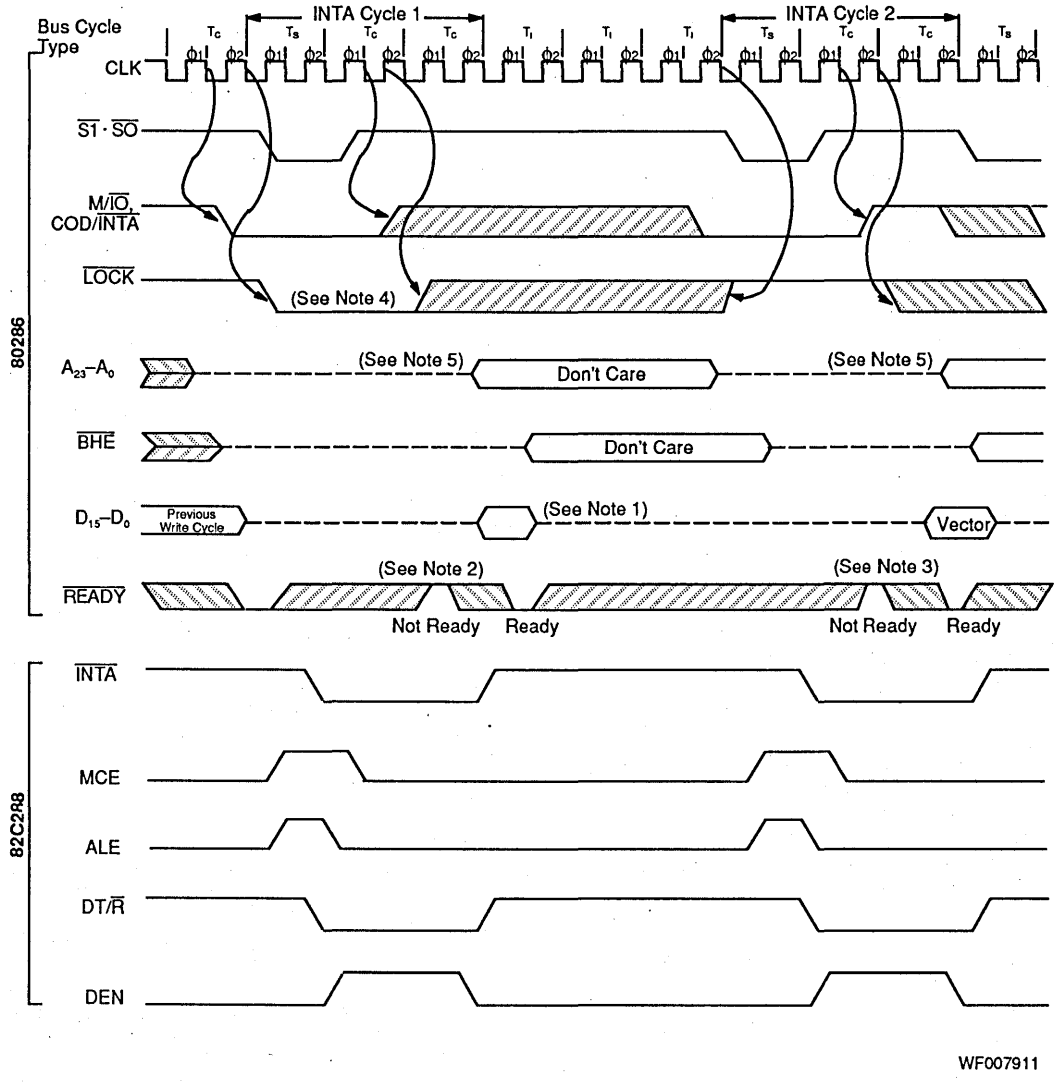


Figure 35. Interrupt Acknowledge Sequence

Notes: 1. Data is ignored.

2. First INTA cycle should have at least one wait state inserted to meet the interrupt controller peripheral minimum INTA pulse width.
3. Second INTA cycle must have at least one wait state inserted since the CPU will not drive A_{23}–A_0, $\overline{BHE}$, and $\overline{LOCK}$ until after the first T_c state.
   The CPU imposed one/clock delay prevents bus contention between cascade address buffer being disabled by MCE ↓ and address outputs.
   Without the wait state, the 80C286 address will not be valid for a memory cycle started immediately after the second INTA cycle. The interrupt controller peripheral also requires one wait state for minimum INTA pulse width.
4. $\overline{LOCK}$ is active for the first INTA cycle to prevent the bus arbiter peripheral from releasing the bus between ITNA cycles in a multi-master system.
5. A_{23}–A_0 exits three-state OFF during Ø2 of the second T_c in the INTA cycle.

Local Bus Usage Priorities

The 80C286 local bus is shared among several internal units and external HOLD requests. In case of simultaneous requests, their relative priorities are:

(Highest) Any transfers which assert $\overline{LOCK}$ either explicitly (via the LOCK instruction prefix) or implicitly (i.e., segment descriptor) access, interrupt acknowledge sequence, or an XCHG with memory).

The second of the two byte bus operations required for an odd aligned word operand.

Local bus request via HOLD input.

Processor extension data operand transfer via PEREQ input.

Data transfer performed by EU as part of an instruction.

(Lowest) An instruction prefetch request from BU. The EU will inhibit prefetching two processor clocks in advance of any data transfers to minimize waiting by EU for a prefetch to finish.

Halt or Shutdown Cycles

The 80C286 externally indicates halt or shutdown conditions as a bus operation. These conditions occur due to a HLT instruction or multiple protection exceptions while attempting to execute one instruction. A halt or shutdown bus operation is signalled when $\overline{S1}$, $\overline{S0}$ and COD/$\overline{INTA}$ are LOW and M/$\overline{IO}$ is HIGH. A_1 HIGH indicates halt, and A_1 LOW indicates shutdown. The bus controller does not issue ALE, nor is $\overline{READY}$ required to terminate a halt or shutdown bus operation.

During halt or shutdown, the 80C286 may service PEREQ or HOLD requests. A processor extension segment overrun exception during shutdown will inhibit further service of PEREQ. Either NMI or RESET will force the 80C286 out of either halt or shutdown. An INTR, if interrupts are enabled, or a processor extension segment overrun exception will also force the 80C286 out of halt.

System Configurations

The versatile bus structure of the 80C286 microsystem, with a full complement of support chips, allows flexible configuration of a wide range of systems. The basic configuration, shown in Figure 36, is similar to an iAPX 86 maximum mode system. It includes the CPU plus an interrupt controller, clock generator, and the Bus Controller. The iAPX 86 latches (29843 and 29845) and transceivers (29833 and 29863) may be used in an 80C286 microsystem.

As indicated by the dashed lines in Figure 36, the ability to add processor extensions is an integral feature of 80C286 microsystems. The processor extension interface allows external hardware to perform special functions and transfer data concurrent with CPU execution of other instructions. Full system integrity is maintained because the 80C286 supervises all data transfers and instruction execution for the processor extension.

The 80C286 with the 80C286 numeric processor extension (NPX) uses this interface. The iAPX C286/C287 has all the instructions and data types of an iAPX 86/87 or iAPX 88/87. The 80C286 NPX can perform numeric calculations concurrently with CPU program execution. Numerics code and data have the same integrity as all other information protected by the 80C286 protection mechanism.

The 80C286 can overlap chip select decoding and address propagation during the data transfer for the previous bus operation. This information is latched into the 29843/45s by ALE during the middle of a T_S cycle. The latched chip select and address information remains stable during the bus operation while the next cycle's address is being decoded and propagated into the system. Decode logic can be implemented with a high-speed bipolar PROM.

The optional decode logic shown in Figure 36 takes advantage of the overlap between address and data of the 80C286 bus cycle to generate advanced memory and I/O-select signals. This minimizes system performance degradation caused by address propagation and decode delays. In addition to selecting memory and I/O, the advanced selects may be used with configurations supporting local and system buses to enable the appropriate bus interface for each bus cycle. The COD/INTA and M/IO signals are applied to the decode logic to distinguish between interrupt, I/O, code and data bus cycles.

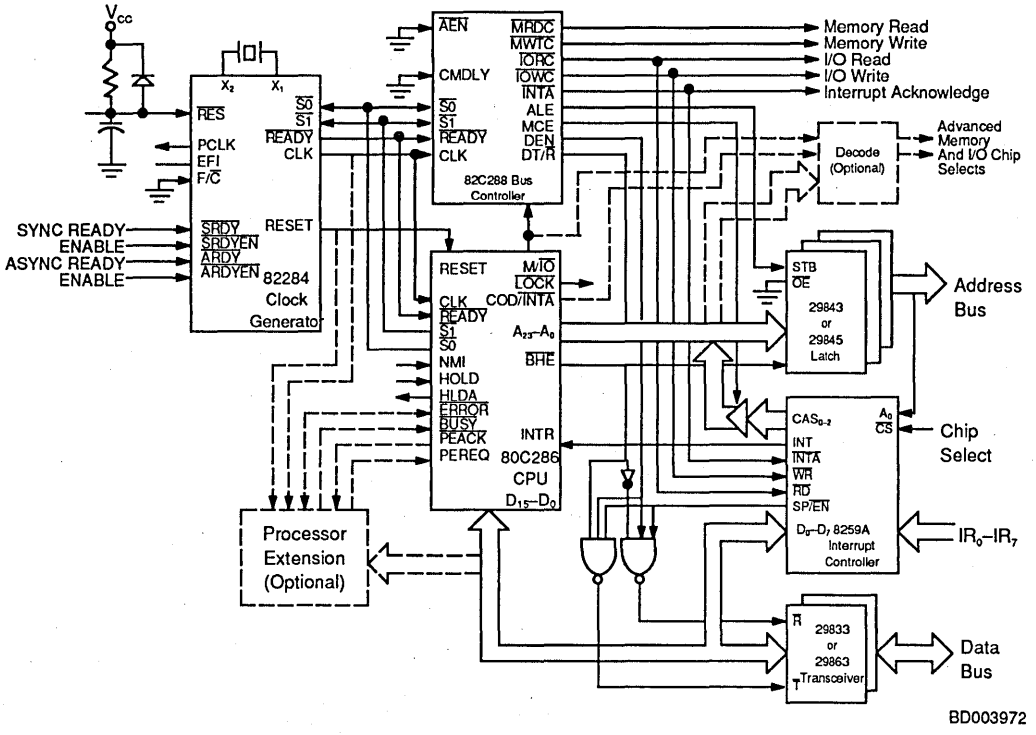


Figure 36. Basic 80C286 System Configuration

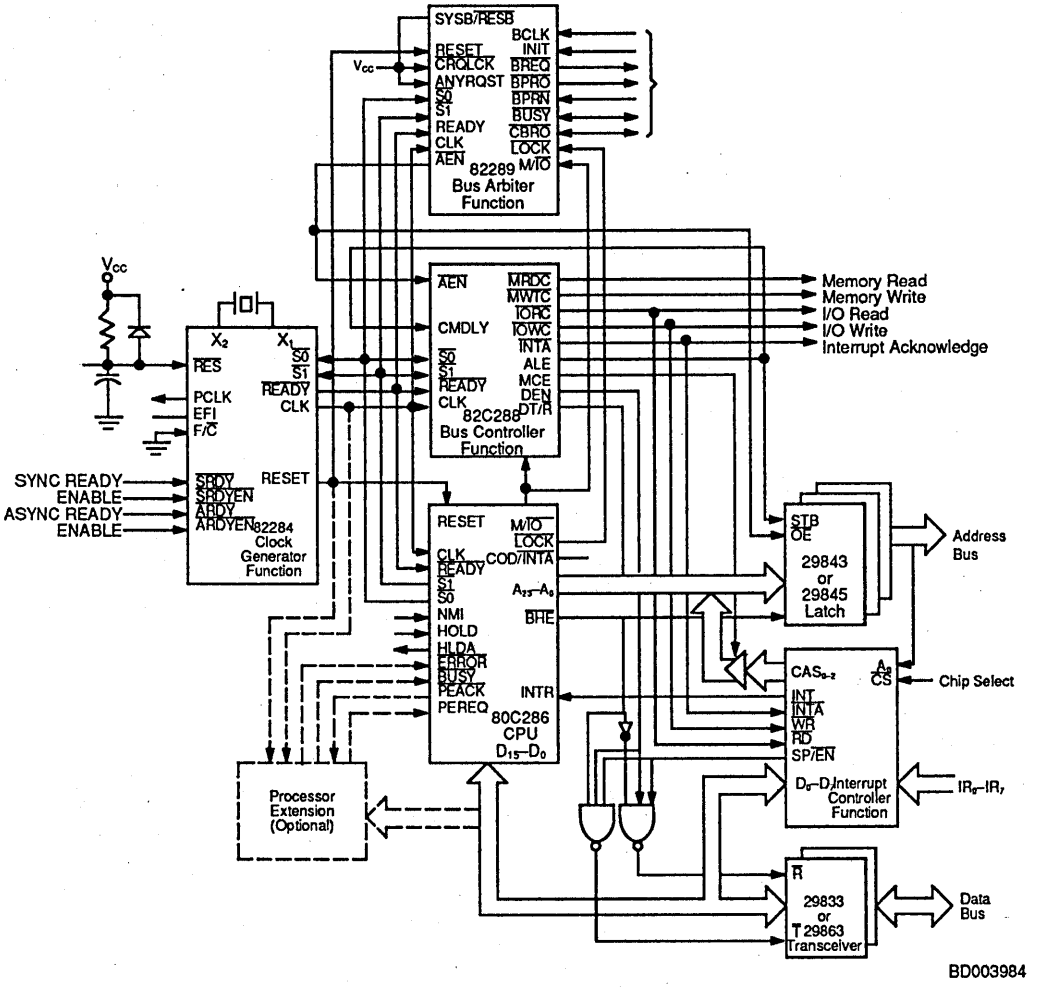
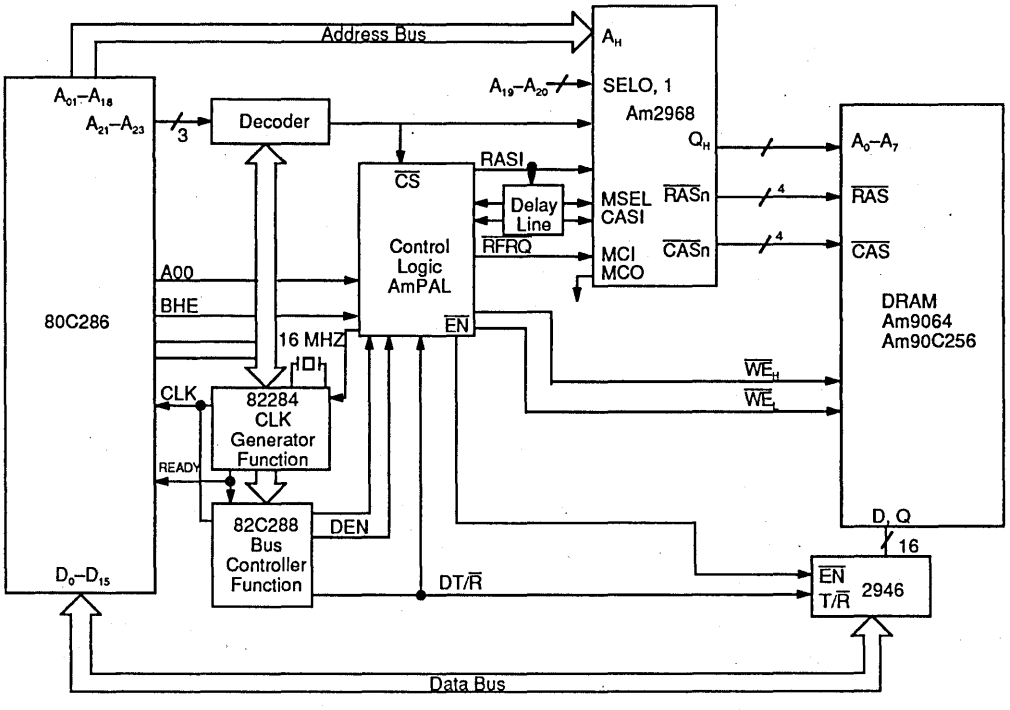


Figure 37. Multibus System Bus Interface

By adding the bus arbiter chip, the 80C286 provides a Multibus system bus interface as shown in Figure 37. The ALE output of the 82C288 for the Multibus bus is connected to its CMDLY input to delay the start of commands one system CLK as required to meet Multibus address and write data set-up times. This arrangement will add at least one extra TC state to each bus operation which uses the Multibus.

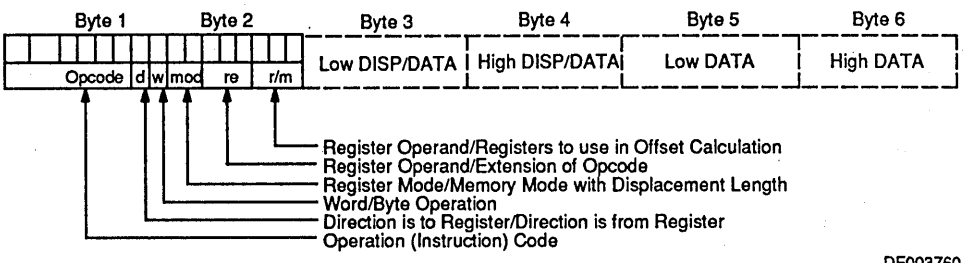
A second bus controller and additional latches and transceivers could be added to the local bus of Figure 37. This configuration allows the 80C286 to support an on-board bus for local memory and peripherals and the Multibus for system bus interfacing.



BD005152

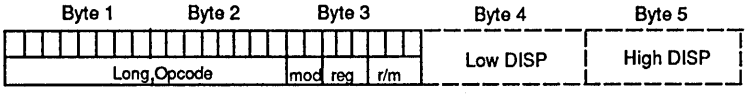
Figure 38. 80C286 Interface with the Am2968 Dynamic Memory Controller

Figure 38 shows the interface of the 80286 with the Am2968 Dynamic Memory Controller. The interface is a timing controller which consists of some control logic and a delay line. The timing controller runs asynchronously to the CPU. It arbitrates between memory requests and refresh requests by generating the proper signals to the dynamic memory controller and memory. The design described is a simple, cost-effective solution to interfacing the 80C286 with the Am2968. A further description about DRAM selection based on processor speed may be found in the Am2968 Application Note.



DF003760

A. Short Opcode Format Example



B. Long Opcode Format Example

DF003770

Figure 39. 80C286 Instruction Format Examples

80C286 INSTRUCTION SET SUMMARY

Instruction Timing Notes

The instruction clock counts listed below establish the maximum execution rate of the 80C286. With no delays in bus cycles, the actual clock count of an 80C286 program will average 5% more than the calculated clock count, due to instruction sequences which execute faster than they can be fetched from memory.

To calculate elapsed times for instruction sequences, multiply the sum of all instruction clock counts, as listed in the table below, by the processor clock period. A 12.5 processor clock has a clock period of 80 nanoseconds and requires an 80C286 system clock (CLK input) of 25 MHz.

Instruction Clock Count Assumptions

1. The instruction has been prefetched, decoded, and is ready for execution. Control transfer instruction clock counts include all time required to fetch, decode, and prepare the next instruction for execution.
2. Bus cycles do not require wait states.
3. There are no processor extension data transfer or local bus HOLD requests.
4. No exceptions occur during instruction execution.

Instruction Set Summary Notes

Addressing displacements selected by the MOD field are not shown. If necessary they appear after the instruction fields shown.

Above/below refers to unsigned value

Greater refers to positive signed value

Less refers to less positive (more negative) signed values

if d = 1 then to register; if d = 0 then from register

if w = 1 then word instruction; if w = 0 then byte instruction

if s = 0 then 16-bit immediate data to form the operand

if s = 0 then an immediate data byte is sign-extended to form the 16-bit operand

x = don't care

z = used for string primitives for comparison with ZF FLAG

If two clock counts are given, the smaller refers to a register operand and the larger refers to a memory operand.

* = add one clock if offset calculation requires summing 3 elements

n = number of bytes of code in next instruction

Level (L)—Lexical nesting level of the procedure

The following comments describe possible exceptions, side effects, and allowed usage for instructions in both operating modes of the 80C286.

Real Address Mode Only

1. This is a protected mode instruction. Attempted execution in real address mode will result in an undefined opcode exception (6).
2. A segment overrun exception (13) will occur if a word operand reference at offset FFFF(H) is attempted.

3. This instruction may be executed in real address mode to initialize the CPU for protected mode.
4. The IOPL and NT fields will remain 0.
5. Processor extension segment overrun interrupt (9) will occur if the operand exceeds the segment limit.

Either Mode

6. An exception may occur, depending on the value of the operand.
7. $\overline{\text{LOCK}}$ is automatically asserted regardless of the presence or absence of the LOCK instruction prefix.
8. $\overline{\text{LOCK}}$ does not remain active between all operand transfers.

Protected Virtual Address Mode Only

9. A general protection exception (13) will occur if the memory operand cannot be used due to either a segment limit or access rights violation. If a stack segment limit is violated, a stack segment overrun exception (12) occurs.
10. For segment load operations, the CPL, RPL, and DPL must agree with privilege rules to avoid an exception. The segment must be present to avoid a not-present exception (11). If the SS register is the destination, and a segment not-present violation occurs, a stack exception (12) occurs.
11. All segment descriptor accesses in the GDT or LDT made by this instruction will automatically assert $\overline{\text{LOCK}}$ to maintain descriptor integrity in multiprocessor systems.
12. JMP, CALL, INT, RET, IRET instructions referring to another code segment will cause a general protection exception (13) if any privilege rule is violated.
13. A general protection exception (13) occurs if CPL $\neq$ 0.
14. A general protection exception (13) occurs if CPL > IOPL.
15. The IF field of the flag word is not updated if CPL > IOPL. The IOPL field is updated only if CPL = 0.
16. Any violation of privilege rules as applied to the selector operand do not cause a protection exception; rather, the instruction does not return a result and the zero flag is cleared.
17. If the starting address of the memory operand violates a segment limit, or an invalid access is attempted, a general protection exception (13) will occur before the ESC instruction is executed. A stack segment overrun exception (12) will occur if the stack limit is violated by the operand's starting address. If a segment limit is violated during an attempted data transfer then a processor extension segment overrun exception (9) occurs.
18. The destination of an INT, JMP, CALL, RET, or IRET instruction must be in the defined limit of a code segment or a general protection exception (13) will occur.

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... +8.0 V
Input, Output or I/O
Voltage Applied GND−1.V to VDD+1.0 V
Power Dissipation/Speed
25 MHz ..... 2.1 W
20 MHz ..... 1.7 W
16 MHz ..... 1.4 W
12 MHz ..... 1.2 W
Storage Temperature Range .... −65°C to +150°C
Junction Temperature ..... +175°C
Lead Temperature
(Soldering, Ten Seconds) ..... +275°C

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Operating Voltage Range ..... +4.5 V to +5.5 V
80C286–20 and –25 Only ..... +4.75 V to +5.25 V
Operating Temperature Range .. 0 to +70°C Ambient
(also meets 0 to 100°C Case Temperature for laptop requirements)

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

V_{CC} = +5 V ± 10%, for 80C286–12 and 80C286–16, V_{CC} = +5 V ±5% for 80C286–20 and 80C286–25, T_A = 0°C to +70°C

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IL}	Input LOW Voltage		–0.5	0.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC}+0.5	V
V_{ILC}	CLK Input LOW Voltage		–0.5	1.0	V
V_{IHC}	CLK Input HIGH Voltage		3.6	V_{CC}+0.5	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA	–	0.4	V
V_{OH}	Output HIGH Voltage	I_{OH} = –2.0 mA I_{OH} = –100 mA	3.0 V_{CC}–0.4	–	V V
I_I	Input Leakage Current	V_{IN} = GND or V_{CC} Pins 29, 31, 57, 59, 61, 63–64	–10	10	µA
I_{SH}	Input Sustaining Current on $\overline{BUSY}$ and $\overline{ERROR}$ Pins	V_{IN} = GND (see Note 5)	–30	–500	µA
I_{BHL}	Input Sustaining Current HIGH	V_{IN} = 1.0 V (see Note 1)	38	200	µA
I_{BHH}	Input Sustaining Current HIGH	V_{IN} = 3.0 V (see Note 2)	–50	–400	µA
I_O	Output Leakage Current	V_O = GND or V_{CC} Pins 1, 7–8, 10–28, 32–34	–10	10	µA
I_{CCOP}	Active Power Supply Current	80C286–12 (see Note 4) 80C286–16 (see Note 4) 80C286–20 (see Note 4) 80C286–25 (see Note 4)	– – – –	220 260 310 360	mA mA mA mA
I_{CCSB}	Standby Power Supply Current	(see Note 3)	–	5	mA

PRELIMINARY

Notes:

1. I_{BHL} should be measured after lowering V_{IN} to GND and then raising to 1.0 V on the following pins: 36–51, 66, 67.
2. I_{BHH} should be measured after raising V_{IN} to V_{CC} and then lowering to 3.0 V on the following pins: 4–6, 36–51, 66–68.
3. I_{CCSB} tested with the clock stopped in phase two of the processor clock cycle. V_{IN} = V_{CC} or GND, V_{CC} = 5.5 V, outputs unloaded.
4. I_{CCOP} measured at 12.5 MHz for the 80C286–12, 16 MHz for the 80C286–16, and 20 MHz for the 80C286–20. V_{IN} = 2.4 V or 0.4 V, V_{CC} = 5.5 V, outputs unloaded.
5. I_{SH} should be measured after raising V_{IN} to V_{CC} and then lowering to GND on pins 53 and 54.

CAPACITANCE (T_Z = +25°C; All Measurements Referenced to Device GND)

Parameter Symbol	Parameter Description	Typ.	Unit	Test Conditions
C_{CLK}	CLK Input Capacitance	10	pF	FREQ = 1MHz
C_{IN}	Other Input Capacitance	10	pF	
$C_{I/O}$	I/O Capacitance	10	pF	

PRELIMINARY

SWITCHING CHARACTERISTICS over operating range

V_{CC} = +5 V ± 10%, T_A = 0°C to +70°C (80C286–12 and 80C286–16)
V_{CC} = +5 V ± 5%, T_A = 0°C to +70°C (80C286–20 and 80C286–25)
AC Timings are referenced to 0.8 V and 2.0 V points of the signals as illustrated in datasheet waveforms, unless otherwise specified.

Parameter Symbol	Parameter Description	Test Conditions	12.5 MHz		16 MHz		Unit
			Min.	Max.	Min.	Max.	
Timing Requirements							
1	System Clock (CLK) Period		40	–	31	–	ns
2	System Clock (CLK) LOW Time	@ 1.0 V	11	–	7	–	ns
3	System Clock (CLK) HIGH Time	@ 3.6 V	13	–	11	–	ns
17	System Clock (CLK) RISE Time	1.0 V to 3.6 V	–	8	–	5	ns
18	System Clock (CLK) FALL Time	3.6 V to 1.0 V	–	8	–	5	ns
4	Asynchronous Inputs SETUP Time	(Note 1)	15	–	5	–	ns
5	Asynchronous Inputs HOLD Time	(Note 1)	15	–	5	–	ns
6	RESET SETUP Time		10	–	10	–	ns
7	RESET HOLD Time		0	–	0	–	ns
8	Read Data SETUP Time		5	–	5	–	ns
9	Read Data HOLD Time		4	–	3	–	ns
10	$\overline{\text{READY}}$ SETUP Time		20	–	12	–	ns
11	$\overline{\text{READY}}$ HOLD Time		20	–	5	–	ns
20	Input RISE/FALL Times	0.8 V to 2.0 V	–	8	–	6	ns
Timing Responses							
12A	Status/$\overline{\text{PEACK}}$ Active Delay	1, (Notes 3, 7)	1	22	1	18	ns
12B	Status/$\overline{\text{PEACK}}$ Inactive Delay	1, (Notes 3, 6)	1	24	1	20	ns
13	Address Valid Delay	1, (Notes 2, 3)	1	32	1	27	ns
14	Write Data Valid Delay	1, (Notes 2, 3)	0	31	0	28	ns
15	Address/Status/Data Float Delay	2, (Note 5)	0	32	0	29	ns
16	HLDA Valid Delay	1, (Notes 3, 8)	0	25	0	25	ns
19	Address Valid to Status SETUP Time	1, (Notes 3, 4)	22	–	16	–	ns

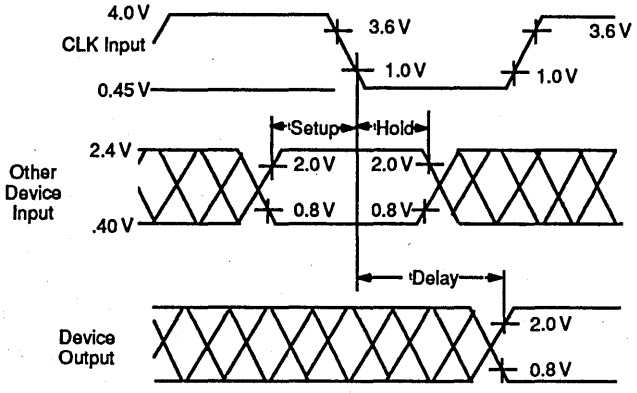
PRELIMINARY

Notes:
1. Asynchronous inputs are INTR, NMI, HOLD, PEREQ, ERROR, and BUSY. This specification is given only for testing purposes, to assure recognition at a specific CLK edge.
2. Delay from 1.0 V on the CLK to 0.8 V or 2.0 V.
3. Output load: C_L = 100 pF.
4. Delay measured from address either reaching 0.8 V or 2.0 V (valid) to status going active reaching 0.8 V or status going inactive reaching 2.0 V.
5. Delay from 1.0 V on the CLK to Float (no current drive) condition.
6. Delay from 1.0 V on the CLK to 0.8 V for min. (HOLD time) and to 2.0 V for max. (inactive delay).
7. Delay from 1.0 V on the CLK to 2.0 V for min. (HOLD time) and to 0.8 V for max. (active delay).
8. Delay from 1.0 V on the CLK to 2.0 V.

Switching Test Conditions

Test Condition	I_L (Constant Current Source	CL
1	2.0 mA	100 pF
2	6 mA (V_{OH} to Float) 8 mA (V_{OL} to Float)	100 pF

PRELIMINARY



AC Setup, Hold and Delay Time Measurement—General WF024251

SWITCHING CHARACTERISTICS (continued)

V_{CC} = +5 V ± 10%, T_A = 0°C to +70°C (80C286–12 and 80C286–16)
V_{CC} = +5 V ± 5%, T_A = 0°C to +70°C (80C286–20 and 80C286–25)
AC timings are referenced to 0.8 V and 2.0 V points of the signals as illustrated in datasheet waveforms, unless otherwise specified.

Parameter Symbol	Parameter Description	Test Conditions	20 MHz		25 MHz		Unit
			Min.	Max.	Min.	Max.	
Timing Requirements							
1	System Clock (CLK) Period		25	–	20	–	ns
2	System Clock (CLK) LOW Time	@ 1.0 V	6	–	5	–	ns
3	System Clock (CLK) HIGH Time	@ 3.6 V	9	–	7	–	ns
17	System Clock (CLK) RISE Time	1.0 V to 3.6 V	–	4	–	4	ns
18	System Clock (CLK) FALL Time	3.6 V to 1.0 V	–	4	–	4	ns
4	Asynchronous Inputs SETUP Time	(Note 1)	4	–	4	–	ns
5	Asynchronous Inputs HOLD Time	(Note 1)	4	–	4	–	ns
6	RESET SETUP Time		10	–	10	–	ns
7	RESET HOLD Time		0	–	0	–	ns
8	Read Data SETUP Time		3	–	3	–	ns
9	Read Data HOLD Time		2	–	2	–	ns
10	$\overline{\text{READY}}$ SETUP Time		10	–	9	–	ns
11	$\overline{\text{READY}}$ HOLD Time		3	–	3	–	ns
20	Input RISE/FALL Times	0.8 V to 2.0 V	–	6	–	6	ns
Timing Responses							
12A	Status/$\overline{\text{PEACK}}$ Active Delay	1, (Notes 3, 6)	1	15	1	9	ns
12B	Status/$\overline{\text{PEACK}}$ Inactive Delay	1, (Notes 3, 6)	1	16	1	13	ns
13	Address Valid Delay	1, (Notes 2, 3)	1	23	1	18	ns
14	Write Data Valid Delay	1, (Notes 2, 3)	0	27	–	24	ns
15	Address/Status/Data Float Delay	2, (Note 5)	0	25	–	15	ns
16	HLDA Valid Delay	1, (Notes 2, 3)	0	20	–	19	ns
19	Address Valid to Status SETUP Time	1, (Notes 3, 4)	13	–	13	–	ns

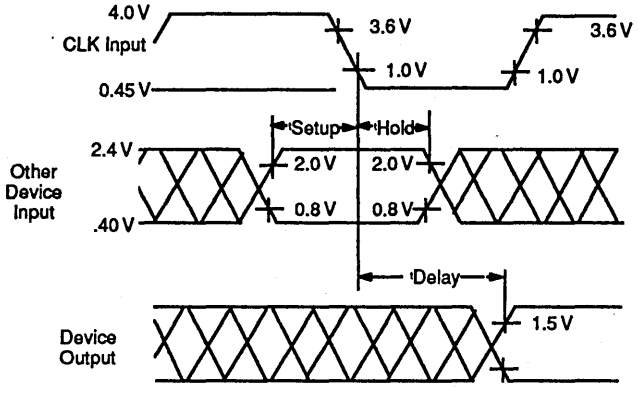
PRELIMINARY

Notes:
1. Asynchronous inputs are INTR, NMI, HOLD, PEREQ, ERROR, and BUSY. This specification is given only for testing purposes, to assure recognition at a specific CLK edge.
2. Delay from 1.0 V on the CLK to 0.8 V or 2.0 V.
3. Output load: C_L = 100 pF.
4. Delay measured from address either reaching 0.8 V or 2.0 V (valid) to status going active reaching 0.8 V or status going inactive reaching 2.0 V.
5. Delay from 1.0 V on the CLK to Float (no current drive) condition.
6. Delay from 1.0 V on the CLK to 0.8 V for min. (HOLD time) and to 2.0 V for max. (inactive delay).

Switching Test Conditions

Test Condition	I_L (Constant Current Source	CL
1	2.0 mA	100 pF
2	6 mA (V_{OH} to Float)	100 pF
	8 mA (V_{OL} to Float)	

PRELIMINARY

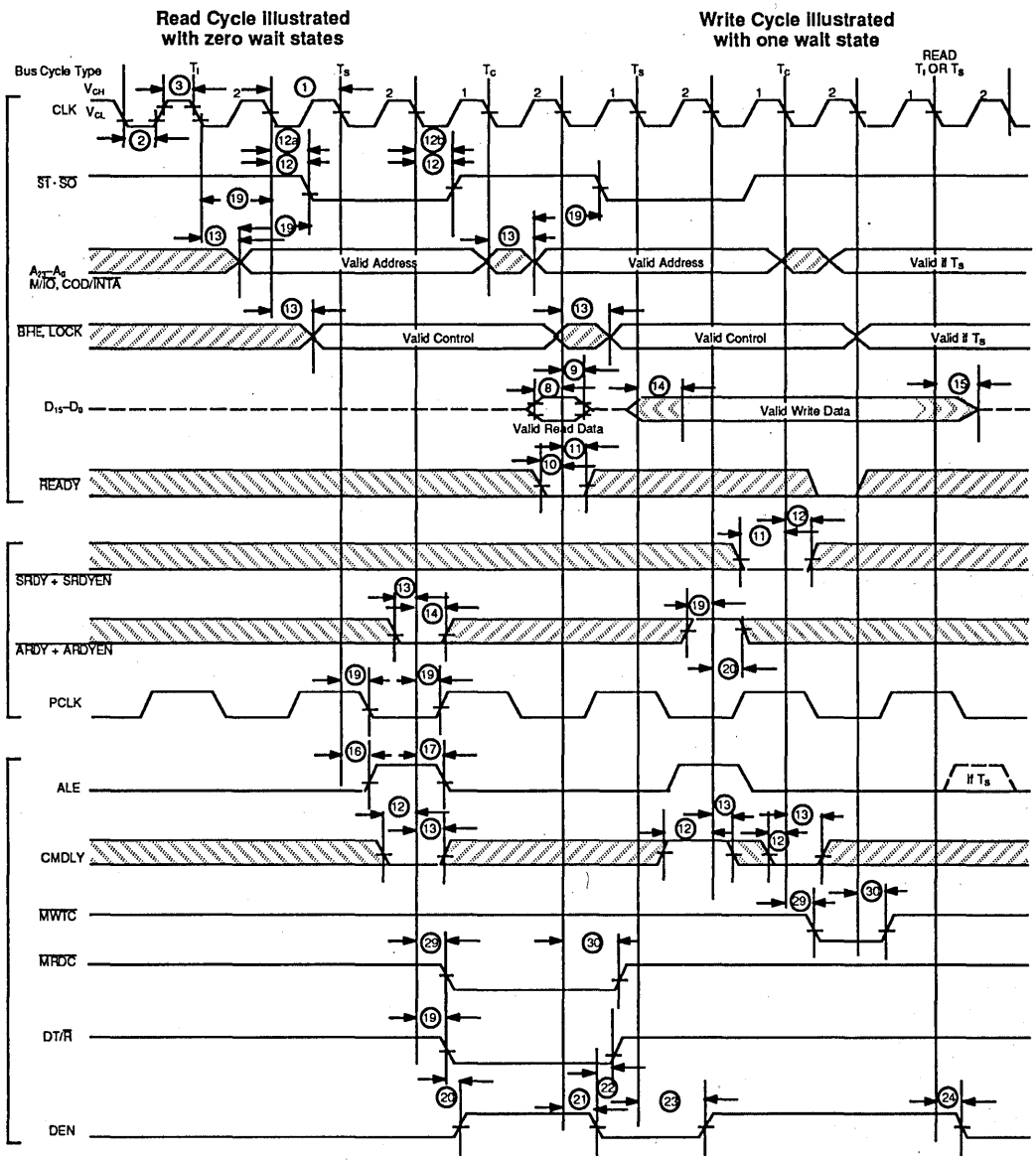


AC Setup, Hold and Delay Time Measurement—For 20/25 MHz only

WF024251

SWITCHING WAVEFORMS

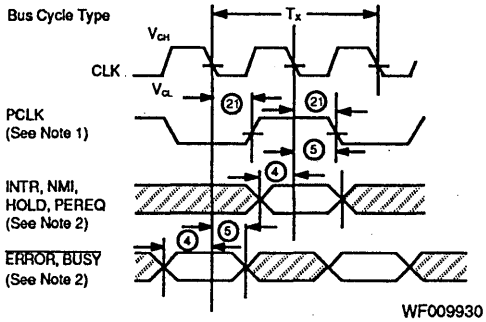
MAJOR CYCLE TIMING



WF007982

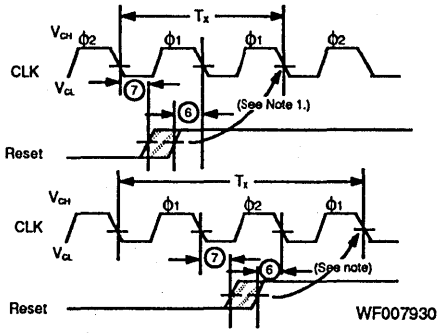
Note: The modified timing is due to the $\overline{\text{CMDLY}}$ signal being active.

80C286 Asynchronous Input Signal Timing



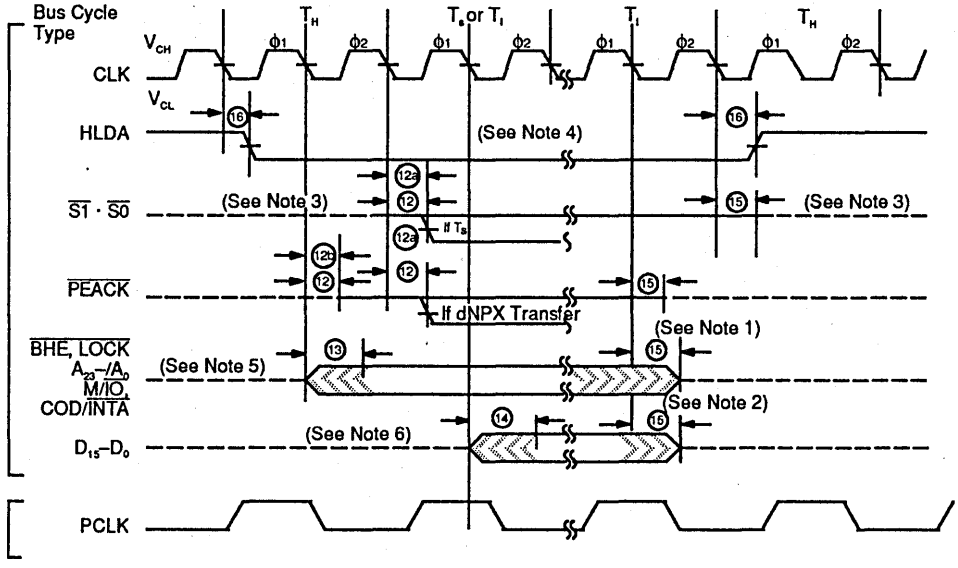
Notes: 1. PCLK indicates which processor cycle phase will occur on the next CLK. PCLK may not indicate the correct phase until the first bus cycle is performed.
2. These inputs are asynchronous. The setup and hold times shown assure recognition for testing purposes.

80C286 Reset Input Timing and Subsequent Processor Cycle Phase



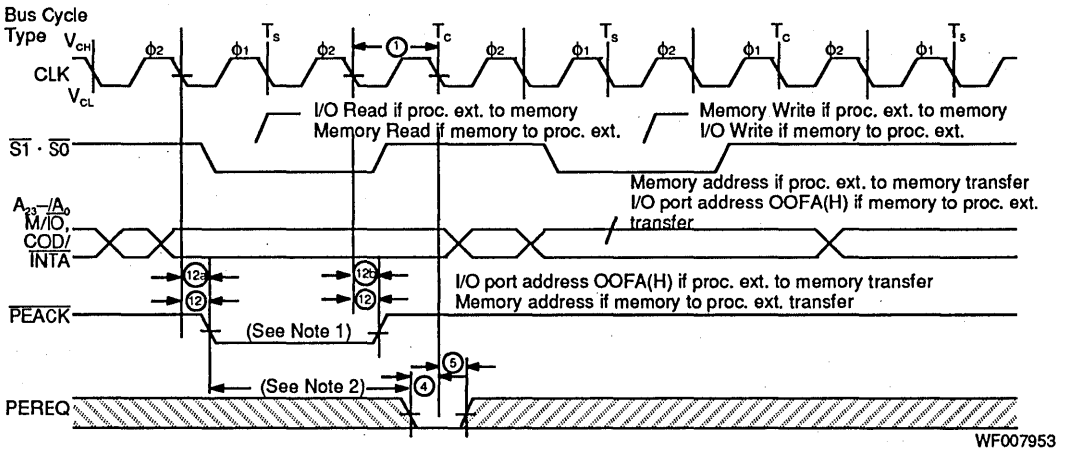
Note When RESET meets the set-up time shown, the next CLK will start or repeat $\phi 1$ of a processor cycle.

Exiting and Entering Hold



Notes: 1. These signals may not be driven by the 80C286 during the time shown. the worst case in terms of latest float time is shown.
2. The data bus will be driven as shown if the last cycle before T_i in the diagram was a write T_c.
3. The 80C286 floats its status pins during T_H. External 20 kΩ resistors keep these signals high (see Table 15).
4. For HOLD request set-up to HLDA, refer to Figure 34.
5. $\overline{BHE}$ and $\overline{LOCK}$ are driven at this time but will not become valid until T_s.
6. The data bus will remain in three-state OFF if a read cycle is performed.

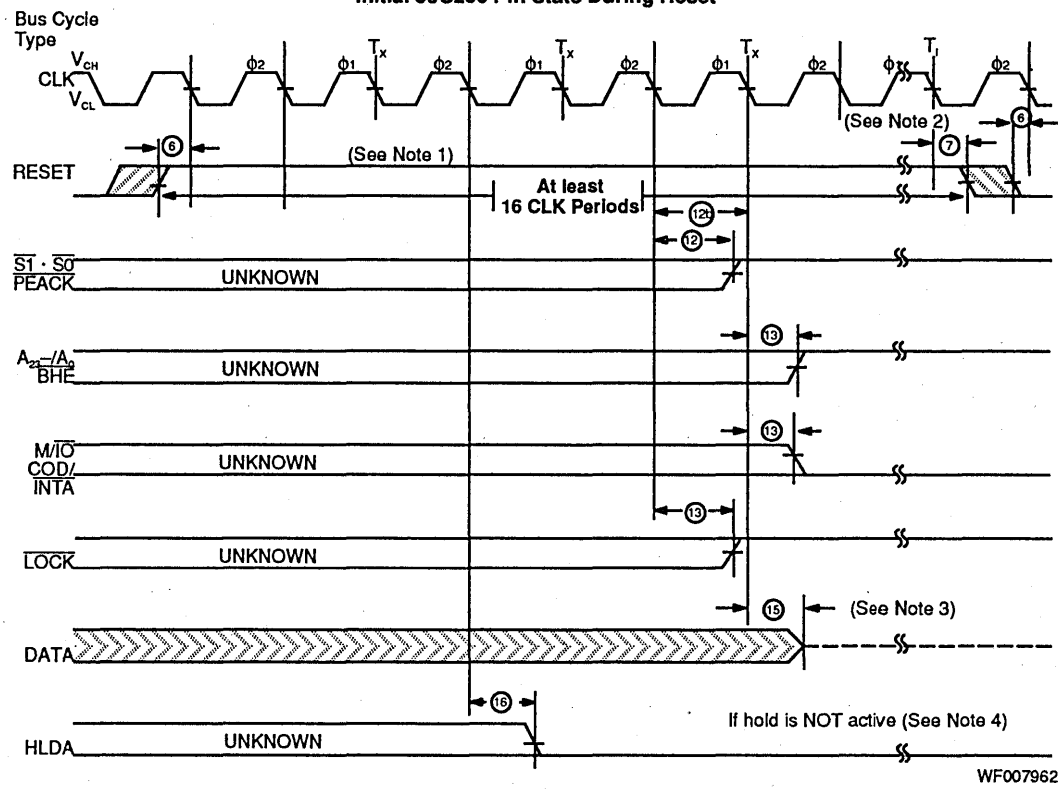
80C286 PEREQ/PEACK Timing Required PEREQ Timing for One Transfer Only



Notes: 1. $\overline{PEACK}$ always goes active during the first bus operation of a processor extension data operand transfer sequence. The first bus operation will be either a memory read at operand address or I/O read at port address OOFA(H).

2. To prevent a second processor extension data operand transfer, the worst case maximum time (shown above) is: 3 x 1–12a max–4 min. The actual, configuration dependent, maximum time is: 3 x 1–12a max–4 min + A x 2 x 1. A is the number of extra T_c states added to either the first or second bus operation of the processor extension data operand transfer sequence.

Initial 80C286 Pin State During Reset



Notes: 1. Set-up time for RESET ↑ may be violated with the consideration that Ø 1 of the processor clock may begin one system CLK period later.

2. Set-up and hold times for RESET ↓ must be met for proper operation, but RESET ↓ may occur during ϕ1 or ϕ2.

3. The data bus is only guaranteed to be in three-state OFF at the time shown.

4. HOLD is acknowledged during RESET, causing HLDA to go active and the appropriate pins to float. If HOLD remains active while RESET goes inactive, the 80C86 remains in HOLD state and will not perform any bus accesses until HOLD is deactivated.

80C286 INSTRUCTION SET SUMMARY

Function	Format				Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
DATA TRANSFER								
MOV = Move:								
Register to Register/Memory	1000100w	mod reg r/m			2,3*	2,3*	2	9
Register/memory to register	1000101w	mod reg r/m			2,5*	2,5*	2	9
Immediate to register/memory	1100011w	mod 000 r/m	data	data if w=1	2,3*	2,3*	2	9
Immediate to register	1011w reg	data	data if w=1		2	2		
Memory to accumulator	1010000w	addr-low	addr-high		5	5	2	9
Accumulator to memory	1010001w	addr-low	addr-high		3	3	2	9
Register/memory to segment register	10001110	mod 0 reg r/m			2,5*	17,19*	2	9,10,11
Segment register to register/memory	10001100	mod 0 reg r/m			2,3*	2,3*	2	9
PUSH = Push:								
Memory	11111111	mod 110 r/m			2,3*	2,3*	2	9
Register	01010reg				3	3	2	9
Segment register	000reg110				3	3	2	9
Immediate	011010s0	data	data if s=0		3	3	2	9
PUSHA = Push All	01100000				17	17	2	9
POP = Pop								
Memory	10001111	mod 000 r/m			5*	5*	2	9
Register	01011reg				5	5	2	9
Segment register	000reg111	(reg ≠01)			5	20	2	9,10,11
POPA = Pop All	01100001				19	19	2	9
XCHG = Exchange:								
Register/memory with register	1000011w	mod reg r/m			3,5*	3,5*	2,7	7,9
Register with accumulator	10010reg				3	3		
IN = Input from:								
Fixed port	1110110w	port			5	5		14
Variable port	1110110w				5	5		14
OUT = Output to:								
Fixed port	1110011w	port			3	3		14
Variable port	1110111w				3	3		14
XLAT = Translate byte to AL	11010111				5	5		9
LEA = Load EA to register	10001101	mod reg r/m			3*	3*		
LDS = Load pointer to DS	11000101	mod reg r/m	(mod p 11)		7*	21*	2	9,10,11
LES = Load pointer to ES	11000100	mod reg r/m	(mod p 11)		7*	21*	2	9,10,11
LAHF = Load AH with flags	10011111				2	2		
SAHF = Store AH into flags	10011110				2	2		
PUSHF = Push flags	10011100				3	3	2	9
POPF = Pop flags	10011101				5	5	2,4	9,15

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

Function	Format				Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
ARITHMETIC								
ADD=Add:								
Reg/memory with register to either	000000dw	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	100000sw	mod 0 0 r/m	data	data if s:w=01	3,7*	3,7*	2	9
Immediate to accumulator	0000010w	data	data if w=1		3	3		
Immediate to register	1011w reg	data	data if w=1		2	2		
ADC = Add with carry:								
Reg/memory with register to either	000100dw	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	100000sw	mod 0 1 0 r/m	data	data if s:w=01	3,7*	3,7*	2	9
Immediate to accumulator	0001010w	data	data if w=1		3	3		
INC =Increment:								
Register/memory	1111111w	mod 0 0 0 r/m			2,7*	2,7*	2	9
Register	01000reg				2	2		
SUB =Subtract:								
Reg/memory and register to either	001010dw	mod reg r/m			2,7*	2,7*	2	9
Immediate from register/memory	100000sw	mod 1 0 1 r/m	data	data if s:w=1	3,7*	3,7*	2	9
Immediate from accumulator	0001110w	data	data if w=1		3	3		
SBB = Subtract with borrow:								
Reg/memory and register to either	000110dw	mod reg r/m			2,7*	2,7*	2	9
Immediate from register/memory	100000sw	mod 0 1 1 r/m	data	data if s:w=01	3,7*	3,7*	2	9
Immediate from accumulator	0001110w	data	data if w=1		3	3		
DEC = Decrement:								
Register/memory	1111111w	mod 0 0 1 r/m			2,7*	2,7*	2	9
Register	01001reg				2	2		
CMP = Compare:								
Register/memory with register	0011101w	mod 0 0 1 r/m			2,6*	2,6*	2	9
Register with register/memory	0011100w	mod reg r/m			2,7*	2,7*	2	9
Immediate with register/memory	100000sw	mod 1 1 1 r/m	data	data if s:w=01	3,6*	3,6*	2	9
Immediate with accumulator	0011110w	data	data if w=1		3	3		
NEG =Change sign	1111011w	mod 0 1 1 r/m			2	7*	2	7
AAA = ASCII adjust for add	00110111				3	3		
DAA = Decimal adjust for add	00100111				3	3		
AAS = ASCII adjust for subtract	00111111				3	3		
DAS = Decimal adjust for subtract	00101111				3	3		
MUL = Multiply (unsigned)	1111011w	mod 1 0 0 r/m						
Register-Byte					13	13		
Register-Word					21	21		
Memory-Byte					16*	16*	2	9
Memory-Word					24*	24*	2	9
IMUL = Integer multiply (signed)	1111011w	mod 1 0 1 r/m						
Register-Byte					13	13		
Register-Word					21	21		
Memory-Byte					16*	16*	2	9
Memory-Word					24*	24*	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80C286 INSTRUCTION SET SUMMARY (continued)

Function	Format				Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
ARITHMETIC (Continued)								
IMUL = Integer Immediate multiply: (signed)	0 1 1 0 1 0 a 1	mod reg r/m	data	data if a = 0	21, 24*	21, 24*	2	9
DIV = Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m			2,3*	2,3*	2	9
Register-Byte					14	14		
Register-Word					22	22		
Memory-Byte					17*	17*	2,6	6,9
Memory-Word					25*	25*	2,6	6,9
IDIV = Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m						
Register-Byte					17	17		
Register-Word					25	25		
Memory-Byte					20*	20*	2	9
Memory-Word					25*	25*	2	9
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0			16	16		
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0			14	14		
CBW = Convert byte to word	1 0 0 1 1 0 0 0				2	2		
CWD = Convert word to double word	1 0 0 1 1 0 0 1				2	2		
LOGIC								
Shift/Rotate Instructions:								
Register/Memory by 1	1 1 0 1 0 0 0 w	mod TTT r/m			2,7*	2,7*	2	9
Register/Memory by CL	1 1 0 1 0 0 1 w	mod TTT r/m			5+n, 8+n*	5+n, 8+n*	2	9
Register Memory by Count	1 1 0 0 0 0 0 w	mod TTT r/m	count		5+n, 8+n*	5+n, 8+n*	2	9
		TTT	**Instruction**					
		0 0 0	ROL					
		0 0 1	ROR					
		0 1 0	RCL					
		0 1 1	RCR					
		1 0 0	SHL/SAL					
		1 0 1	SHR					
		1 1 1	SAR					
AND = And:								
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1		3	3		
TEST = And function to flags, no result								
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m			2,6*	2,6*	2	9
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1	3,6*	3,6*	2	9
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1		3	3		
OR = Or:								
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1		3	3		
XOR = Exclusive or:								
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m			2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1		3	3		
NOT = Invert register/memory	1 1 1 1 0 1 1 w	mod 0 1 0 r/m			2,7*	2,7*	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

Function	Format	Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
STRING MANIPULATION:					
MOVS = Move Byte/word	1010010w	5	5	2	9
CMPS = Compare byte/word	1010011w	8	8	2	9
SCAS = Scan byte/word	1010111w	7	7	2	9
LODS = Load byte/wd to AL/AX	1010110w	5	5	2	9
STOS = Stor byte/wd from AL/A	1010101w	3	3	2	9
INS = Input byte/wd from DX port	0110110w	5	5	2	9,14
OUTS = Output byte/wd to DX port	0110111w	5	5	2	9,14
Repeated by count in CX					
MOVS = Move string	11110010 1010010w	5+4n	5+4n	2	9
CMPS = Compare string	1111001z 1010011w	5+9n	5+9n	2	9
SCAS = Scan string	1111001z 1010111w	5+8n	5+8n	2	9
LODS = Load string	1110010 1010110w	5+4n	5+4n	2	9
STOS = Store string	11110010 1010101w	4+3n	4+3n	2	9
INS = Input string	11110010 0110110w	5+4n	5+4n	2	9,14
OUTS = Output string	11110010 0110111w	5+4n	5+4n	2	9,14
CONTROL TRANSFER					
CALL = Call:					
Direct within segment	11101000 disp-low disp-high	7+m	7+m	2	8
Register memory indirect within segment	11111111 mod 0 1 0 r/m	7+m,11+m	7+m,11+m	2	8,9
Direct intersegment	10011010 segment offset / segment selector	13+m	26+m	2	8,11,12
Protected Mode Only (Direct Intersegment):					
Via call gate to same privilege level			41+m		8,11,12
Via call gate to different privilege level, no parameters			82+m		8,11,12
Via call gate to different privilege level, x parameters			86+4x+m		8,11,12
Via TSS			177+m		8,11,12
Via task gate			182+m		8,11,12
Indirect intersegment	11111111 mod 0 1 1 r/m (mod ≠ 11)	16+m	29+m*	2	8,9,11,12
Protected Mode Only (Indirect Intersegment):					
Via call gate to same privilege level			44+m		8,9,11,12
Via call gate to different privilege level, no parameters			83+m		8,9,11,12
Via call gate to different privilege level, x parameters			90+4x+m		8,9,11,12
Via TSS			180+m		8,9,11,12
Via task gate			185+m		8,9,11,12
JMP = Unconditional jump					
Short/long	11101011 disp-low	7+m	7+m		8
Direct within segment	11101001 disp-low disp-high	7+m	7+m		8
Register/mem indirect within segment	11111111 mod 1 0 0 r/m	7+m,11+m	7+m,11+m	2	8,9
Direct intersegment	11101010 segment offset / segment selector	11+m	23+m	2	8,11,12
Protected Mode Only (Indirect Intersegment):					
Via call gate to same privilege level			38+m		8,11,12
Via TSS			175+m		8,11,12
Via task gate			180+m		8,11,12
Indirect intersegment	11111111 mod 1 0 1 r/m (mod ≠ 11)	15+m	26+m*	2	8,9,11,12

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80C286 INSTRUCTION SET SUMMARY (continued)

Function	Format	Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
CONTROL TRANSFER (continued):					
Protected Mode Only (Indirect Intersegment)					
Via call gate to same privilege level			41+m*		8,9,11,12
Via TSS			178+m*		8,9,11,12
Via task gate			183+m*		8,9,11,12
RET = Return from CALL:					
Within segment	11000011	11+m	11+m	2	8,9
Within seg adding immed to SP	11000010 | data-low | data-high	11+m	11+m	2	8,9
Intersegment	11001011	15+m	25+m	2	8,9,11,12
Intersegment adding immediate to SP	11001010 | data-low | data-high	15+m		2	8,9,11,12
Protected Mode Only (RET): To different privilege level					
JE/JZ = Jump on equal zero	01110100 | disp	7+m or 3	7+m or 3		8
JL/JNGE = Jump on less not greater or equal	01111100 | disp	7+m or 3	7+m or 3		8
JLE/JNG = Jump on less or equal not greater	01111110 | disp	7+m or 3	7+m or 3		8
JB/JNAE = Jump on below not above or equal	01110010 | disp	7+m or 3	7+m or 3		8
JBE/JNA = Jump on below or equal not above	01110110 | disp	7+m or 3	7+m or 3		8
JP/JPE = Jump on parity/ parity even	01111010 | disp	7+m or 3	7+m or 3		8
JO = Jump on overflow	01110000 | disp	7+m or 3	7+m or 3		8
JS = Jump on sign	01111000 | disp	7+m or 3	7+m or 3		8
JNE/JNZ = Jump on not equal not zero	01110101 | disp	7+m or 3	7+m or 3		8
JNL/JGE = Jump on not less greater or equal	01111101 | disp	7+m or 3	7+m or 3		8
JNLE/JG = Jump on not less or equal greater	01111111 | disp	7+m or 3	7+m or 3		8
JNB/JAE = Jump on not below above or equal	01110011 | disp	7+m or 3	7+m or 3		8
JNBE/JA = Jump on not below or equal above	01110111 | disp	7+m or 3	7+m or 3		8
JNP/JPO = Jump on not par/par odd	01111011 | disp	7+m or 3	7+m or 3		8
JNO = Jump on not overflow	01110001 | disp	7+m or 3	7+m or 3		8
JNS = Jump on not sign	01111001 | disp	7+m or 3	7+m or 3		8
LOOP = Loop CX Times	11100010 | disp	8+m or 34	8+m or 4		8
LOOPZ/LOOPE = Loop while zero equal	11100001 | disp	8+m or 34	8+m or 4		8
LOOPNZ/LOOPNE = Loop while not zero equal	11100000 | disp	8+m or 34	8+m or 4		8
JCXZ = Jump on CX zero	11100011 | disp	8+m or 34	8+m or 4		8
ENTER = Enter Procedure	11001000 | data-low | data-high | L				
L = 0		11	11	2	9
L = 1		15	15	2	9
L > 1		16-4(L-1)	16-4(L-1)	2	9
LEAVE = Leave Procedure	11001001	5	5	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80C286 INSTRUCTION SET SUMMARY (continued)

Function	Format	Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
CONTROL TRANSFER (continued):					
INT = Interrupt:					
Type specified	11001101 | type	23+m		2	
Type 3	11001100	23+m		2	
INTO = Interrupt on overflow	11001110	24–m or 3 (3 if no interrupt)	24 – or 3 (3 if no interrupt)	2	
Protected Mode Only:					
Via interrupt or trap gate to same privilege level			40+m		8,11,12
Via interrupt or trap gate to fit different privilege level			78+m		8,11,12
Via Task Gate			167+m		8,11,12
IRET = Interrupt return	11001111	17+m	31+m	2,4	8,9,11, 12,15
Protected Mode Only:					
To different privilege level			55+m		8,9,11, 12,15
To different task (NT = 1)			169+m		8,9,11,12
BOUND = Detect value out of range	01100010 | mod reg r/m	13	13 (Use INT clock count if exception 6	2,6	6,8,9, 11,12
PROCESSOR CONTROL					
CLC = Clear carry	11111000	2	2		
CMC = Complement carry	11110101	2	2		
STC = Set carry	11111001	2	2		
CLD = Clear direction	11111100	2	2		
STD = Set direction	11111101	2	2		
CLI = Clear interrupt	11111010	3	3		14
STI = Set interrupt	11111011	2	2		14
HLT = Halt	11110100	2	2		13
WAIT = Wait	10011011	3	3		
LOCK = Bus lock prefix	11110000	0	0		14
CTS = Clear task switched flag	00001111 | 00000110	2	2	3	13
ESC = Processor Extension Escape	10011TTT | mod LLL r/m (TTT LL are opcode to processor extension)	9–20*	9–20*	5	17
SEG = Segment override prefix	001 reg 110	0	0		

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

Function	Format	Clock Count: Real Address Mode	Clock Count: Protected Virtual Address Mode	Comments: Real Address Mode	Comments: Protected Virtual Address Mode
PROTECTION CONTROL:					
LGDT = Load global descriptor table register	00001111 | 00000001 | mod 010 r/m	11*	11*	2,3	9,13
SGDT = Store global descriptor table register	00001111 | 00000001 | mod 000 r/m	11*	11*	2,3	9,13
LIDT = Load interrupt descriptor table register	00001111 | 00000001 | mod 011 r/m	12*	12*	2,3	9,13
SIDT = Store interrupt descriptor table register	00001111 | 00000001 | mod 001 r/m	12*	12*	2,3	9
LLDT = Load local descriptor table register from table memory	00001111 | 00000000 | mod 010 r/m		17,19*	1	9,11,13
SLDT = Store local descriptor table register to register/memory	00001111 | 00000000 | mod 000 r/m		2,3*	1	9
LTR = Load task register from register/memory	00001111 | 00000000 | mod 011 r/m		17,19*	1	9,11,13
STR = Store task register to register memory	00001111 | 00000000 | mod 001 r/m		2,3*	1	9,11,13
LMSW = Load machine status word from register/memory	00001111 | 00000001 | mod 110 r/m	3,6*	3,6*	2,3	9,13
SMSW = Store machine status word	00001111 | 00000001 | mod 100 r/m	2,3*	2,3*	2,3	9
LAR = Load access rights from register/memory	00001111 | 00000010 | mod reg r/m		14,16*	1	9,16
LSL = Load segment limit from register/memory	00001111 | 00000011 | mod reg r/m		14,16*	1	9,16
ARPL = Adjust requested privilege level from register/memory	01100011 | mod reg r/m		10,11*	2	9
VERR = Verify read access: register/memory	00001111 | 00000000 | mod 100 r/m		14,16*	1	9,16
VERR = Verify write access:	00001111 | 00000000 | mod 101 r/m		14,16*	1	9,16

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

Footnotes

The effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp–high; disp-low

if mod = 000 then EA = (BX) + (SI) + DISP
if mod = 001 then EA = (BX) + (DI) + DISP
if mod = 010 then EA = (BP) + (SI) + DISP
if mod = 011 then EA = (BP) + (DI) + DISP
if mod = 100 then EA = (SI) + DISP
if mod = 101 then EA = (DI) + DISP
if mod = 110 then EA = (BP) + DISP*
if mod = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)
*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0	0	1	reg	1	1	0

REG is assigned according to the following:

REG	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)	
000	AX	000 AL
001	CX	001 CL
010	DX	010 DL
011	BX	011 BL
100	SP	100 AH
101	BP	101 CH
110	SI	110 DH
111	DI	111 BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those

Final

80L286

Low-Power High-Performance Microprocessor with Memory Management and Protection

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- High-performance processor (up to 13.3 times iAPX 86 when using the 16 MHz 80L286)
- Identical to the 80286 except consumes less power
- Available in cost-effective Plastic Leaded Chip Carrier (PLCC) package
- Socketed PLCC footprint is compatible with socketed LCC and PGA footprints
- Surface-mountable PLCC for high density board utilization
- 8, 10, and 12.5 and16 MHz operation
- Large address space
  - 16 megabytes physical
  - 1 gigabyte virtual memory per task
- Integrated memory management, four-level memory protection and support for virtual memory and operating systems

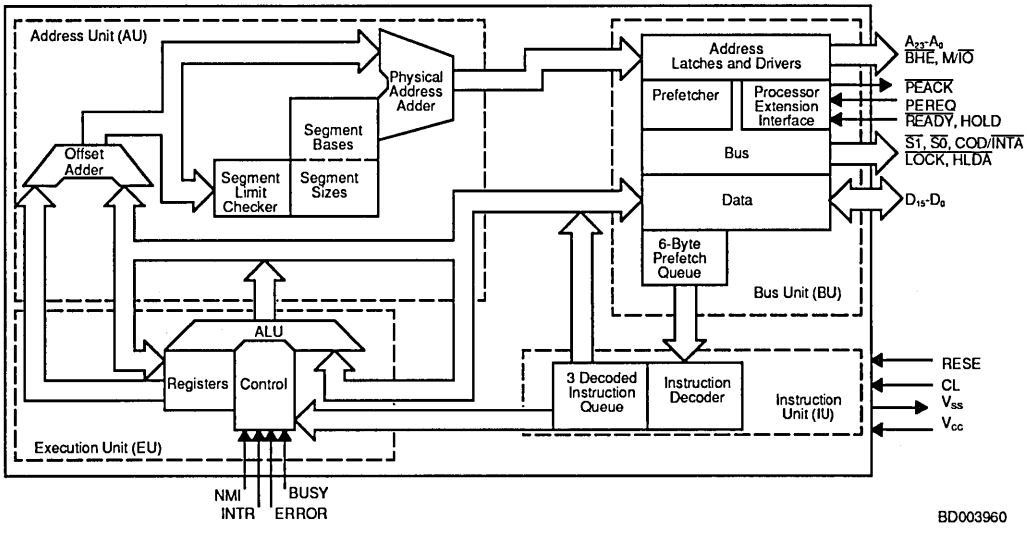
GENERAL DESCRIPTION

The 80L286 is an advanced, high performance microprocessor, identical to the 80286, except consumes less power. Its reduced power enables the 80L286 to be packaged in low-cost, Plastic Leaded Chip Carrier (PLCC) without a heat sink or heat spreader. Cooler operation also enhances reliability. The PLCC package can be surface-mounted or socketed. The footprint of the socketed PLCC package is identical to the socketed LCC or PGA packages so no board layout change is needed. The 80L286 is available in 8, 10, and 12 and 16 MHz speeds and is fully compatible with the 82C288 Bus Controller and the 82284 Clock Driver functions.

The 80L286 is upward compatible with iAPX 86 and 88 software. Using iAPX real address mode, the 80L286 is object code compatible with existing iAPX 86, 88 software. In protected virtual address mode, the 80L286 is source code compatible with iAPX 86, 88 software and may require upgrading to use virtual addresses supported by the 80L286's integrated memory management and protection mechanism. Both modes operate at full 80L286 performance and execute a superset of the iAPX 86 and 88 instructions.

1

BLOCK DIAGRAM



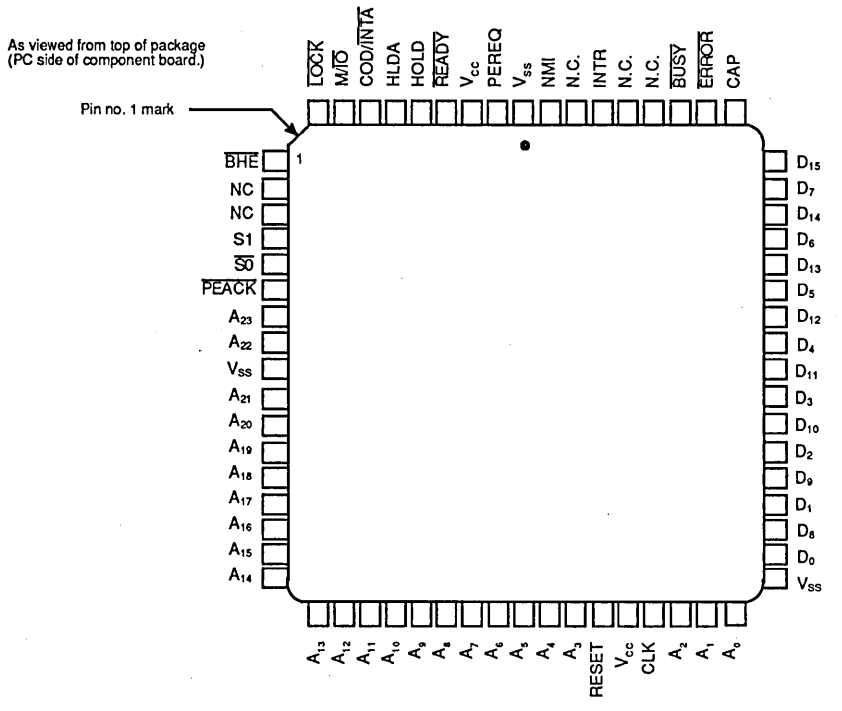
BD003960

GENERAL DESCRIPTION (continued)

The 80L286 provides special operations to support the efficient implementation and execution of operating systems. For example, one instruction can end execution of one task, save its state, switch to a new task, load its state, and start execution of the new task. The 80L286 also supports virtual memory systems by providing a segment-not-present exception and restartable instructions.

CONNECTION DIAGRAM
Top View

PLCC



CD010641

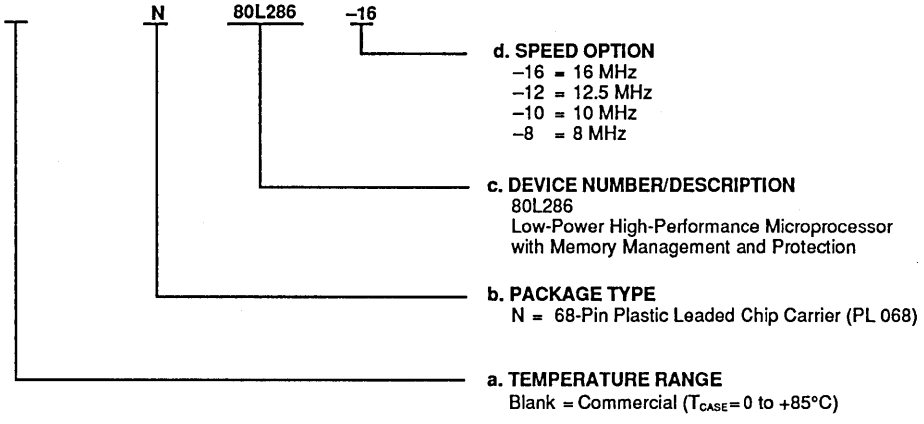
As viewed from top of package (PC side of component board)

ORDERING INFORMATION

Standard Products

AMD commodity products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
N	80L286-16 80L286-12 80L286-10 80L286-8

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

CLK

System Clock (Input; Active HIGH)

System Clock provides the fundamental timing for 80L286 systems. It is a 16 MHz signal divided by two inside the 80L286 to generate the 8 MHz processor clock. The internal divide-by-two circuitry can be synchronized to an external clock generator by a LOW-to-HIGH transition on the RESET input.

D_0-D_{15}

Data Bus (Input/Output; Active HIGH)

Data Bus inputs data during memory, I/O, and interrupt acknowledge read cycles; outputs data during memory and I/O write cycles. The data bus is active HIGH and floats to three-state OFF during bus hold acknowledge.

A_{23}-A_0

Address Bus (Output; Active HIGH)

Address Bus outputs physical memory and I/O port addresses. A_0 is LOW when data is to be transferred on pins D_{7-0}. A_{23}-A_{16} are LOW during I/O transfers. The address bus is active HIGH and floats to three-state OFF during bus hold acknowledge.

$\overline{BHE}$

Bus High Enable (Output; Active LOW)

Bus High Enable indicates transfer of data on the upper byte of the data bus D_{15-8}. Eight-bit oriented devices assigned to the upper byte of the data bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is active LOW and floats to three-state OFF during bus hold acknowledge.

$\overline{BHE}$ and A_0 Encodings

$\overline{BHE}$ Value	A_0 Value	Function
0	0	Word transfer
0	1	Byte transfer on upper half of data bus (D_{15-8})
1	0	Byte transfer on lower half of data bus (D_{7-0})
1	1	Reserved

$\overline{S1}$, $\overline{S0}$

Bus Cycle Status (Output; Active LOW)

Bus Cycle Status indicates initiation of a bus cycle and, along with M/$\overline{IO}$ and COD/$\overline{INTA}$, defines the type of bus cycle. The bus is in a T_S state whenever one or both are LOW. $\overline{S1}$ and $\overline{S0}$ are active LOW and float to three-state OFF during bus hold acknowledge.

80L286 Bus Cycle Status Definition

COD/$\overline{INTA}$	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Bus cycle initiated
0 (LOW)	0	0	0	Interrupt acknowledge
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	None; not a status cycle
0	1	0	0	IF $A_1 = 1$ then halt; else shutdown
0	1	0	1	Memory data read
0	1	1	0	Memory data write
0	1	1	1	None; not a status cycle
1 (HIGH)	0	0	0	Reserved
1	0	0	1	I/O Read
1	0	1	0	I/O Write
1	0	1	1	None; not a status cycle
1	1	0	0	Reserved
1	1	0	1	Memory instruction read
1	1	1	0	Reserved
1	1	1	1	None; not a status cycle

M/$\overline{IO}$

Memory/IO Select (Output)

Memory/IO Select distinguishes memory access from I/O access. If HIGH during T_S, a memory cycle or a halt/shutdown cycle is in progress. If LOW, an I/O cycle or an interrupt acknowledge cycle is in progress M/$\overline{IO}$ floats to three-state OFF during bus hold acknowledge.

COD/$\overline{INTA}$

Code/Interrupt Acknowledge (Output)

Code/Interrupt Acknowledge distinguishes instruction fetch cycles from memory data read cycles. Also distinguishes interrupt acknowledge cycles from I/O cycles. COD/$\overline{INTA}$ floats to three-state OFF during bus hold acknowledge.

$\overline{LOCK}$

Bus Lock (Output; Active LOW)

Bus Lock indicates that other system bus masters are not to gain control of the system bus following the current bus cycle. The $\overline{LOCK}$ signal may be activated explicitly by the "LOCK" instruction prefix or automatically by 80286 hardware during memory XCHG instructions, interrupt acknowledge, or descriptor table access. $\overline{LOCK}$ is active LOW and floats to three-state OFF during hold acknowledge.

PIN DESCRIPTION (continued)

$\overline{\text{READY}}$

Bus Ready (Input; Active LOW)

Bus Ready terminates a bus cycle. Bus cycles are extended without limit until terminated by $\overline{\text{READY}}$ LOW. $\overline{\text{READY}}$ is an active LOW synchronous input requiring set-up and hold times relative to the system clock be met for correct operation. $\overline{\text{READY}}$ is ignored during bus hold acknowledge.

HOLD, HLDA

Bus Hold Request and Hold Acknowledge (Input/Output; Active HIGH)

Bus Hold Request and Hold Acknowledge control ownership of the 80L286 local bus. The HOLD input allows another local bus master to request control of the local bus. When control is granted, the 80L286 will float its bus drivers to three-state OFF and then active HLDA, thus entering the bus hold acknowledge condition. The local bus will remain granted to the requesting master until HOLD becomes inactive which results in the 80L286 deactivating HLDA and regaining control of the local buys. This terminates the bus hold acknowledge condition. HOLD may be asynchronous to the system clock. These signals are active HIGH.

INTR

Interrupt Request (Input; Active HIGH)

Interrupt Request requests the 80L286 to suspend its current program execution and service a pending external request. Interrupt requests are masked whenever the interrupt enable bit in the flag word is cleared. When the 80L286 responds to an interrupt request, it performs two interrupt acknowledge bus cycles to read an 8-bit interrupt vector that identifies the source of the interrupt. To assure program interruption, INTR must remain active until the first interrupt acknowledge cycle is completed. INTR is sampled at the beginning of each processor cycle and must be active HIGH at least two processor cycles before the current instruction ends in order to interrupt before the next instruction. INTR is level sensitive, active HIGH, and may be asynchronous to the system clock.

NMI

Non-maskable Interrupt Request (Input; Active HIGH)

Non-maskable Interrupt Request interrupts the 80L286 with an internally supplied vector value of 2. No interrupt acknowledge cycles are performed. The interrupt enable bit in the 80L286 flag word does not affect this input. The NMI input is active HIGH, may be asynchronous to the system clock, and is edge triggered after internal synchronization. For proper recognition, the input must have been previously LOW for at least four system clock cycles and remain HIGH for at least four system clock cycles.

PEREQ, $\overline{\text{PEACK}}$

Processor Extension Operand Request and Acknowledge (Input/Output)

Processor Extension Operand Request and Acknowledge extends the memory management and protection capabilities of the 80L286 to processor extensions. The PEREQ input requests the 80L286 to perform a data operand transfer for a processor extension. The $\overline{\text{PEACK}}$ output signals the processor extension when the requested operand is being transferred. PEREQ is active HIGH and may be asynchronous to the system clock. $\overline{\text{PEACK}}$ is active LOW.

$\overline{\text{BUSY}}$, $\overline{\text{ERROR}}$

Processor Extension Busy and Error (Input; Active LOW)

Processor Extension Busy and Error indicate the operating condition of a processor extension to the 80L286. An active $\overline{\text{BUSY}}$ input stops 80L286 program execution on WAIT and some ESC instructions until $\overline{\text{BUSY}}$ becomes inactive (HIGH). The 80L286 may be interrupted while waiting for $\overline{\text{BUSY}}$ to become inactive. An active $\overline{\text{ERROR}}$ input causes the 80L286 to perform a processor extension interrupt when executing WAIT or some ESC instructions. These inputs are active LOW and may be asynchronous to the system clock.

RESET

System Reset (Input; Active HIGH)

System Reset clears the internal logic of the 80L286 and is active HIGH. The 80L286 may be reinitialized at any time with a LOW-to-HIGH transition on RESET which remains active for more than 16 system clock cycles. During RESET active, the output pins of the 80C286 enter the state shown below:

80L286 Pin State during Reset

Pin Value	Pin Names
1 (HIGH)	$\overline{S0}$, $\overline{S1}$, $\overline{\text{PEACK}}$, A_{23}-A_0, $\overline{\text{BHE}}$, $\overline{\text{LOCK}}$
0 (LOW)	M/$\overline{IO}$, COD/$\overline{INTA}$, HLDA
three-state OFF	D_{15}-D_0

Operation of the 80L286 begins after a HIGH-to-LOW transition on RESET. The HIGH-to-LOW transition of RESET must be synchronous to the system clock. Approximately 50 system clock cycles are required by the 80L286 for internal initializations before the first bus cycle to fetch code from the power-on execution address is performed.

PIN DESCRIPTION (continued)

A LOW-to-HIGH transition of RESET synchronous to the system clock, will begin a new processor cycle at the next HIGH-to-LOW transition of the system clock. The LOW-to-HIGH transition of RESET may be asynchronous to the system clock; however, in this case it cannot be predetermined which phase of the processor clock will occur during the next system period. Synchronous LOW-to-HIGH transitions of RESET are only required for systems where the processor clock must be phase synchronous to another clock.

V_{SS}

System Ground (Input; Active HIGH)

System Ground: 0 volts.

V_{CC}

System Power (Input; Active HIGH)

System Power: +5 volt power supply.

CAP

Substrate Filter Capacitor (Input; Active HIGH)

Substrate Filter Capacitor: a 0.047 µf ±20% 12 V capacitor must be connected between this pin and ground. This capacitor filters the output of the internal substrate bias generator. A maximum DC leakage current of 1 µA is allowed through the capacitor.

For correct operation of the 80L286, the substrate bias generator must charge this capacitor to its operating voltage. The capacitor charge-up time is 5 milliseconds (max.) after V_{CC} and CLK reach their specified AC and DC parameters. RESET may be applied to prevent spurious activity by the CPU during this time. After this time, the 80L286 processor clock can be phase synchronized to another clock by pulsing RESET LOW synchronous to the system clock.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65 to +150°C
Voltage on Any Pin with
Respect to Ground ..... –1.0 to 7.0 V
Power Dissipation ..... 2.89 Watts

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Operating Voltage Range ..... +4.5 V to +5.5 V
C80C286–20 Only ..... +4.75 V to +5. V
Operating Temperature Range ... 0 to +70°C

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

(T_{CASE} = 0 to 85°C, V_{CC} = 5 V ±5%)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IL}	Input LOW Voltage		–.5	0.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + .5	V
V_{ILC}	CLK Input LOW Voltage		–.5	.6	V
V_{IHC}	CLK Input HIGH Voltage		3.8	V_{CC} + .5	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA		.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = –400 µA	2.4		V
I_{LI}	Input Leakage Current	0 V ≤ V_{IN} ≤ V_{CC}		±10	µA
I_{LO}	Output Leakage Current	.45 V ≤ V_{OUT} ≤ V_{CC}		±10	µA
I_{CC}	Supply Current	T_C = 0°C		550	mA
		T_C = 85°C		475	mA
C_{CLK}	CLK Input Capacitance	F_C = 1 MHz		20	pF
C_{IN}	Other Input Capacitance	F_C = 1 MHz		10	pF
C_O	Input/Output Capacitance	F_C = 1 MHz		20	pF
I_{LO}	Output Leakage Current	0 V ≤ V_{OUT} < .045 V		±1	mA
I_{IL}	Input Sustaining Current on $\overline{BUSY}$ and $\overline{ERROR}$ pins	V_{IN} = 0 V	30	500	µA
I_{LCR}	Input CLK Leakage Current	0.45 ≤ V_{IN} ≤ V_{CC}		±10	µA
I_{LCR}	Input CLK Leakage Current	0 V ≤ V_{IN} ≤ 0.45 V		±1	mA

Note: Low temperature is worst case.

SWITCHING CHARACTERISTICS

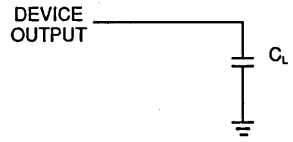
V_{CC} = +5 V ± 5%, T_{CASE} = 0 to +85°C
AC timings are referenced to 0.8 V and 2.0 V points of the signals as illustrated in datasheet waveforms, unless otherwise noted.

Parameter Symbol	Parameter Description	Test Conditions	8 MHz		10 MHz		Unit
			Min.	Max.	Min.	Max.	
1	System Clock (CLK) Period		62	125	50	125	ns
2	System Clock (CLK) LOW Time	@ 1.0 V	15	100	12	109	ns
3	System Clock (CLK) HIGH Time	@ 3.6 V	25	110	16	113	ns
17	System Clock (CLK) RISE Time	1.0 V to 3.6 V		10		8	ns
18	System Clock (CLK) FALL Time	3.6 V to 1.0 V		10		8	ns
4	Asynchronous Inputs SETUP Time	(Note 1)	20		20		ns
5	Asynchronous Inputs HOLD Time	(Note 1)	20		20		ns
6	RESET SETUP Time		28		23		ns
7	RESET HOLD Time		5		5		ns
8	Read Data SETUP Time		10		8		ns
9	Read Data HOLD Time		8		8		ns
10	$\overline{\text{READY}}$ SETUP Time		38		26		ns
11	$\overline{\text{READY}}$ HOLD Time		25		25		ns
12	Status/$\overline{\text{PEACK}}$ Valid Delay	(Notes 2, 30	1	40	–	–	ns
12A	Status/$\overline{\text{PEACK}}$ Active Delay	(Notes 2,3)	–	–	1	22	ns
12B	Status/$\overline{\text{PEACK}}$ Inactive Delay	1(Notes 2,3)	–	–	1	30	ns
13	Address Valid Delay	(Notes 2, 3)	1	60	1	35	ns
14	Write Data Valid Delay	(Notes 2, 3)	0	50	0	30	ns
15	Address/Status/Data Float Delay	(Notes 2, 4)	0	50	0	47	ns
16	HLDA Valid Delay	(Notes 2, 3)	0	50	0	47	ns
19	Address Valid to Status SETUP Time	(Notes 3, 5, 64)	38		27		ns

SWITCHING CHARACTERISTICS (continued)

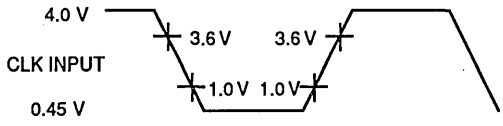
Parameter Symbol	Parameter Description	Test Conditions	12.5 MHz		16 MHz		Unit
			Min.	Max.	Min.	Max.	
1	System Clock (CLK) Period		40	125	31	125	ns
2	System Clock (CLK) LOW Time	@ 1.0 V	11	112	10	113	ns
3	System Clock (CLK) HIGH Time	@ 3.6 V	13	114	12	115	ns
17	System Clock (CLK) RISE Time	1.0 V to 3.6 V		8		5	ns
18	System Clock (CLK) FALL Time	3.6 V to 1.0 V		8		4	ns
4	Asynchronous Inputs SETUP Time	(Note 1)	15		11		ns
5	Asynchronous Inputs HOLD Time	(Note 1)	15		11		ns
6	RESET SETUP Time		18		14		ns
7	RESET HOLD Time		5		3		ns
8	Read Data SETUP Time		5		5		ns
9	Read Data HOLD Time		6		5		ns
10	$\overline{\text{READY}}$ SETUP Time		22		15		ns
11	$\overline{\text{READY}}$ HOLD Time		20		15		ns
12	Status/$\overline{\text{PEACK}}$ Valid Delay	(Notes 2, 30	–	–	–	–	ns
12A	Status/$\overline{\text{PEACK}}$ Active Delay	(Notes 2,3)	3	18	1	18	ns
12B	Status/$\overline{\text{PEACK}}$ Inactive Delay	1(Notes 2,3)	3	20	1	20	ns
13	Address Valid Delay	(Notes 2, 3)	1	32	1	29	ns
14	Write Data Valid Delay	(Notes 2, 3)	0	30	0	22	ns
15	Address/Status/Data Float Delay	(Notes 2, 4)	0	32	0	29	ns
16	HLDA Valid Delay	(Notes 2, 3)	0	25	0	25	ns
19	Address Valid to Status SETUP Time	(Notes 3, 5, 64)	22		22		ns

Notes
1. Asynchronous inputs are INTR, NMI, HOLD, PEREQ, ERROR, and BUSY. This specification is given only for testing purposes, to assure recognition at a specific CLK edge.
2. Delay from 1.0 V on the CLK to 0.8 V or 2.0 V or float on the output as appropriate for valid or floating condition..
3. Output load: $C_L = 100$ pF.
4. Float condition occurs when output current is less than ILO in magnitude.
5. Delay measured from address either reaching 0.8 V or 2.0 V (valid) to status going active reaching 2.0 V or status going inactive reaching 0.8 V.
6. For load capacitance of 10 pF on STATUS/PEACK lines, subtract typically 7 ns for 8 MHz spec, and maximum 7 ns for 10 MHz spec.



NOTE 7:
AC Test Loading on Outputs

TC004190

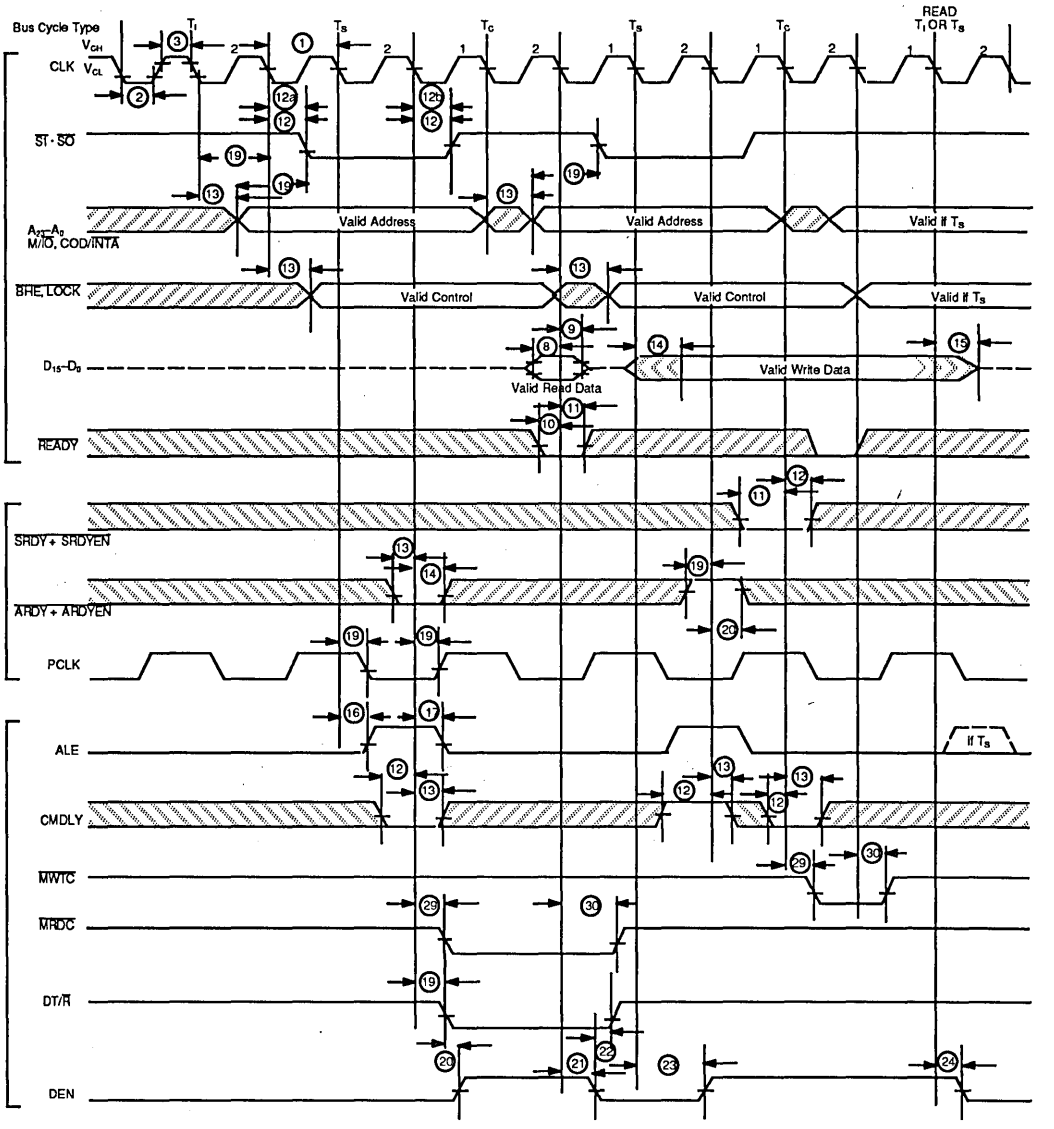


NOTE 8:
AC Drive and Measurement Points—CLK Input

WF024240

SWITCHING WAVEFORMS

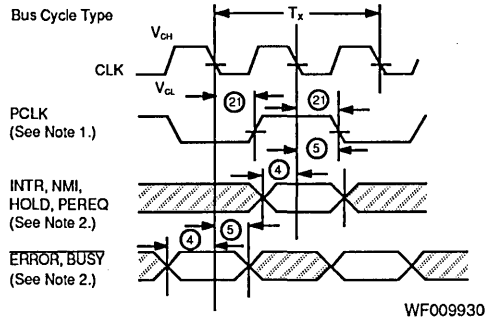
MAJOR CYCLE TIMING



WF007983

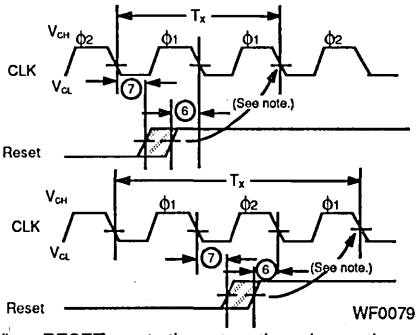
Note: $\overline{\text{MWTC}}$ is valid at this point onlyu if CMDLY is LOW.

80L280 Asynchronous Input Signal Timing



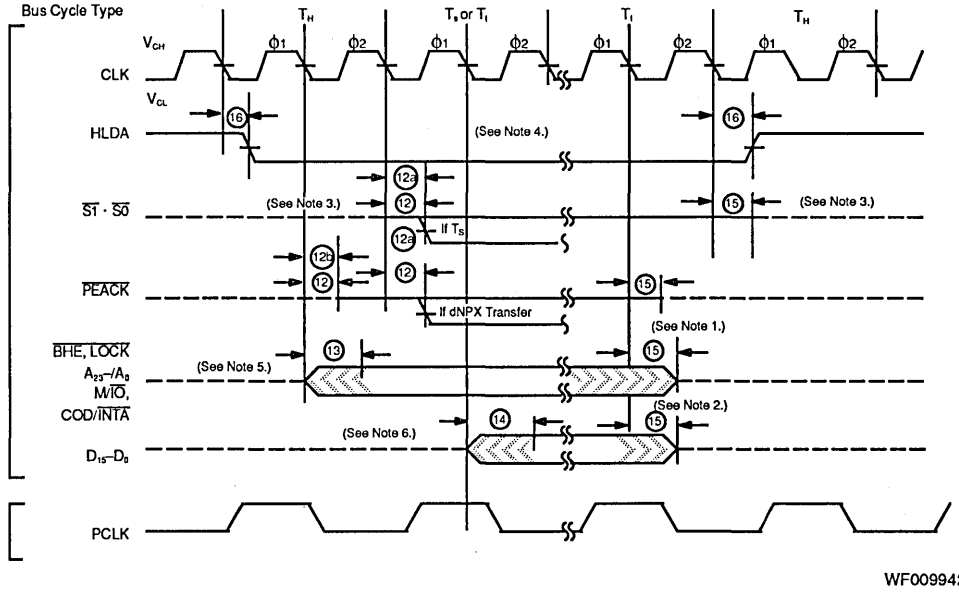
Notes: 1. PCLK indicates which processor cycle phase will occur on the next CLK. PCLK may not indicate the correct phase until the first bus cycle is performed.
2. These inputs are asynchronous. The setup and hold times shown assure recognition for testing purposes.

80L286 Reset Input Timing and Subsequent Processor Cycle Phase



Note: When RESET meets the set-up time shown, the next CLK will start or repeat k1 of a processor cycle.

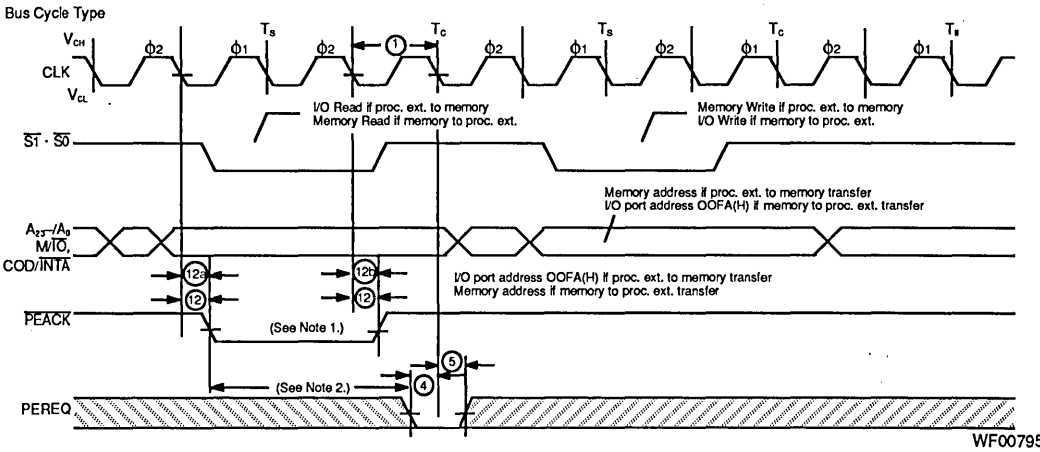
Exiting and Entering Hold



Notes: 1. These signals may not be driven by the 80L286 during the time shown. the worst case in terms of latest float time is shown.
2. The data bus will be driven as shown if the last cycle before T_i in the diagram was a write T_c.
3. The 80C286 floats its status pins during T_H. External 20 kΩ resistors keep these signals high.
4. For HOLD request set-up to HLDA, refer to Figure 34.
5. $\overline{BHE}$ and $\overline{LOCK}$ are driven at this time but will not become valid until T_s.
6. The data bus will remain in three-state OFF if a read cycle is performed.

SWITCHING WAVEFORMS (continued)

80L286 PEREQ/$\overline{\text{PEACK}}$ Timing Required PEREQ Timing for One Transfer Only

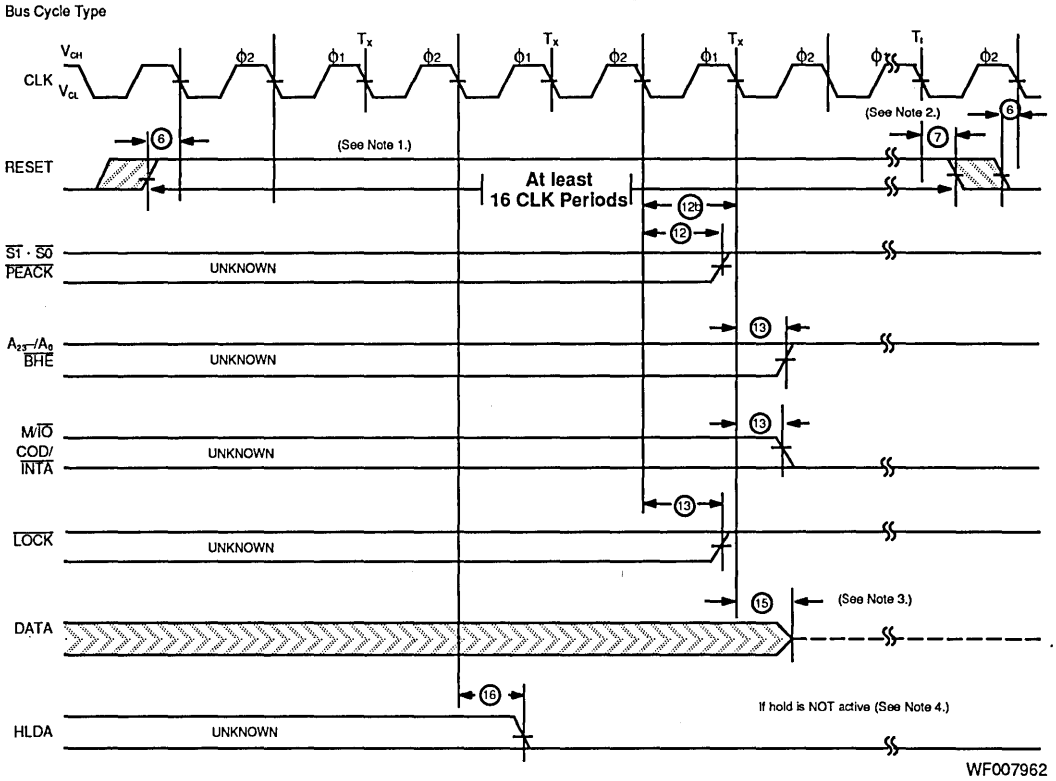


WF007953

Notes: 1. $\overline{\text{PEACK}}$ always goes active during the first bus operation of a processor extension data operand transfer sequence. The first bus operation will be either a memory read at operand address or I/O read at port address OOFA(H).

2. To prevent a second processor extension data operand transfer, the worst case maximum time (shown above) is: 3 x 1–12a max–4 min. The actual, configuration dependent, maximum time is: 3 x 1–12a max–4 min + A x 2 x 1. A is the number of extra T_c states added to either the first or second bus operation of the processor extension data operand transfer sequence.

Initial 80L286 Pin State During Reset



Notes: 1. Set-up time for RESET ↑ may be violated with the consideration that Ø 1 of the processor clock may begin one system CLK period later.

2. Set-up and hold times for RESET ↓ must be met for proper operation, but RESET ↓ may occur during Ø 1 or Ø 2.

3. The data bus is only guaranteed to be in three-state OFF at the time shown.

4. HOLD is acknowledged during RESET, causing HLDA to go active and the appropriate pins to float. If HOLD remains active while RESET goes inactive, the 80C86 remains in HOLD state and will not perform any bus accesses until HOLD is deactivated.

Advance Information

AMD 80C287

80-Bit CMOS Numeric Processor

Advanced Micro Devices

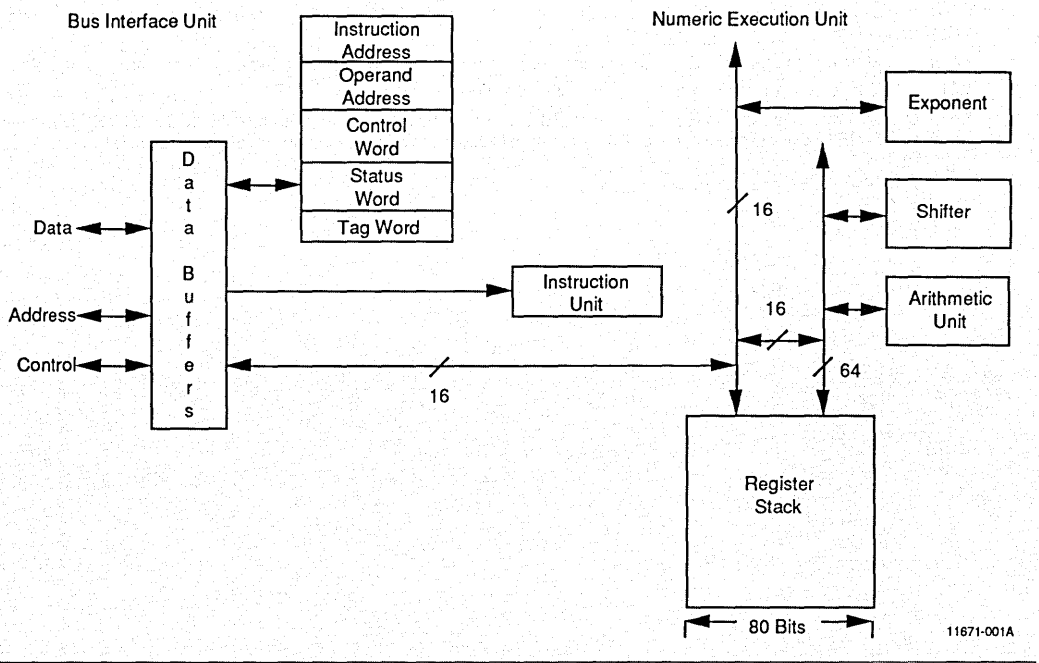
DISTINCTIVE CHARACTERISTICS

- **Pin compatible and functionally equivalent to the Intel 80287**
- **High-performance CMOS process yields 10-MHz, 12-MHz, and 16-MHz speed grades**
- **Available in space-saving 44-pin PLCC as well as 40-pin DIP**
- **80-bit numeric accelerator for 80C286 and 80286-based systems**
- **Compatible with IEEE floating-point standard 754**
- **Static CMOS design does not require a minimum clock rate, resulting in significantly lower power dissipation**
- **Performs single-, double-, and extended-precision floating-point, as well as word, short, and long integer and 18-digit BCD conversions**
- **Adds trigonometric, logarithmic, exponential, and arithmetic instructions to the 80C286 instruction set**

GENERAL DESCRIPTION

The 80C287 is implemented in AMD's advanced static CMOS process that allows for significantly higher speeds at a much lower power dissipation than traditional NMOS versions or standard CMOS. Functionally equivalent to the Intel 80287, the 80C287 is a high-performance arithmetic processor that expands the 80C286 instruction set with floating-point instructions including transcendentals, and integer and BCD conversions. The floating-point operations comply with the IEEE Standard 754. The device is available in 8-, 10-, 12-, and 16-MHz speed grades and is provided in 44-pin PLCC and 40-pin DIP packages. When coupled with the 80C286, the 80C287 provides a complete solution for high-performance numeric processing applications.

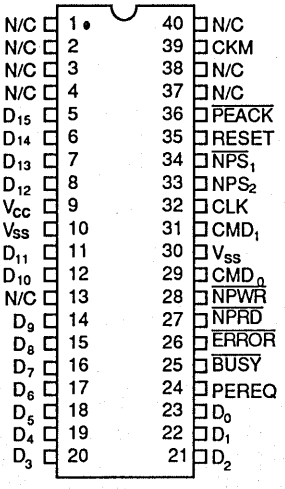
BLOCK DIAGRAM



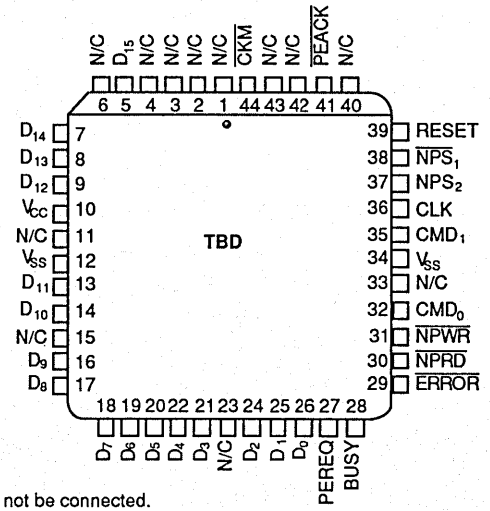
Publication # 11671 Rev. B Amendment /0
Issue Date: June 1989

CONNECTION DIAGRAM

DIP



PLCC



Note: N/C pins should not be connected.
Pin 1 is marked for orientation.

11671-002A

PIN DESCRIPTION

BUSY Busy Status (Output; Active Low)

A LOW level indicates that the 80C287 is currently executing a command.

CKM Clock Mode Signal (Input)

When CKM is HIGH, the CLK is used directly. When CKM is LOW, CLK is divided by three. This input must be either HIGH or LOW 20 CLK cycles before RESET goes LOW.

CLK Clock (Input)

Provides timing for 80C287 operations.

CMD_1, CMD_0 Command Lines (Input)

CMD_1 and CMD_0, along with select inputs, allow the CPU to direct the 80C287 operations. These inputs are timed relative to the read and write strobes.

D_{15}–D_0 Data (Input/Output)

Bidirectional data bus. These inputs are timed relative to the read and write strobes.

ERROR Error Status (Output; Active Low)

Reflects the error summary status bit of the status word. A LOW level indicates that an unmasked exception condition exists.

NPRD Numeric Processor Read (Input; Active Low)

A LOW level enables transfer of data from the 80C287. This input may be asynchronous to the 80C287 clock.

$\overline{NPS_1}$, NPS_2 Numeric Processor Selects (Input)

Indicates the CPU is transferring data to and from the 80C287. Asserting both signals ($\overline{NPS_1}$ LOW and NPS_2 HIGH) enables the 80C287 to transfer floating-point data or instructions. No data transfers involving the 80C287 will occur unless the 80C287 is selected via $\overline{NPS_1}$ and NPS_2. These inputs are timed relative to the read and write strobes.

NPWR Numeric Processor Write (Input; Active Low)

A LOW level enables transfer of data from the 80C287. This input may be asynchronous to the 80C287 clock.

PEACK Processor Extension Acknowledge (Input; Active Low)

A LOW level indicates that the request signal (PEREQ) has been recognized. PEACK causes the request (PEREQ) to be withdrawn when no more transfers are required. PEACK may be asynchronous to the 80C287 clock.

PEREQ Processor Extension Request (Output)

A HIGH level indicates that the 80C287 is ready to transfer data. PEREQ will be disabled upon assertion of PEACK or upon actual data transfer, whichever occurs first, when no more transfers are required.

RESET System Reset (Input)

Reset causes the 80C287 to immediately terminate its present activity and enter a dormant state. Reset must be HIGH for more than four CLK cycles. For proper initialization the HIGH-LOW transition must occur no sooner than 50 µs after V_{CC} and CLK meet their DC and AC specifications.

V_{CC} +5 V Supply (Input)

V_{SS} System Ground (Input)

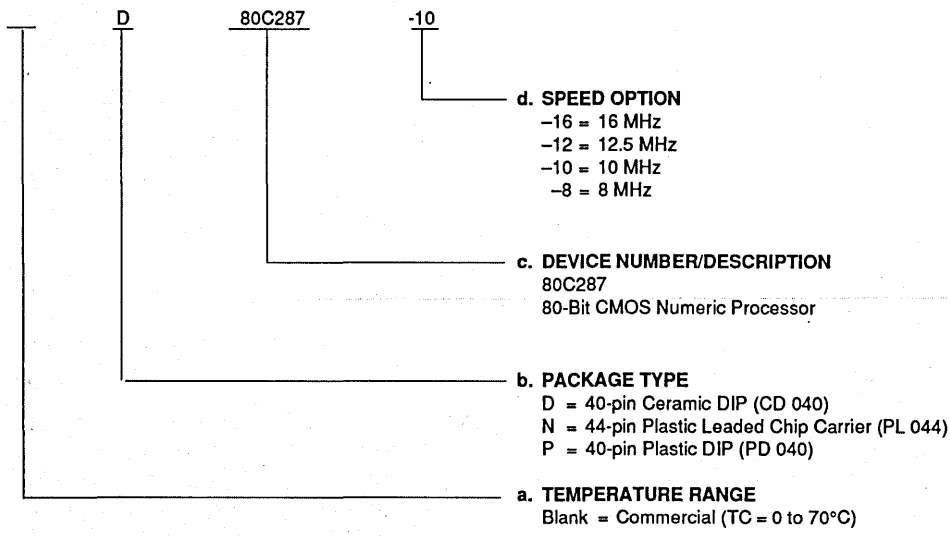
Both pins must be connected to ground.

ORDERING INFORMATION

Commodity Products

AMD products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option (if applicable)
e. Optional Processing



Valid Combination	
D, N, P	80C287–16
	80C287–12
	80C287–10
	80C287–8

Valid Combination

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released valid combinations.

SIMPLIFIED FUNCTIONAL DESCRIPTION

The 80C287 is internally divided into two basic processing units; the numeric execution unit, and the bus interface unit as shown in the block diagram. The numeric execution unit performs numeric instructions. The bus interface unit receives and decodes instructions, executes processor control instructions, and requests operands transfers to and from memory. The 80C286 may execute non-numeric instruction concurrently with numeric instruction executed on the 80C287. Synchronization and error recognition occurs when the next numeric instruction is decoded by the 80C286.

The Numeric Execution Unit

The numeric execution data path is 80 bits wide. All operands are converted to the internal 80-bit format before use. These instructions include arithmetic, transcendental, constant, and data transfer instructions.

The Bus Interface Unit

The bus interface unit decodes the ESC instruction executed by the 80C286. The signal $\overline{BUSY}$ is activated for 80C286/80C287 synchronization and the signal $\overline{ERROR}$ is activated for error detection. $\overline{BUSY}$ is activated when an instruction is transferred and deactivated when the instruction completes. $\overline{ERROR}$ will be asserted if an error has occurred when $\overline{BUSY}$ is deactivated.

The signals PEREQ, $\overline{PEACK}$, $\overline{NPRD}$, $\overline{NPWR}$, $\overline{NPS_1}$, CMD_0, CMD_1, and NPS_2 control data transfers between the 80C287 and the 80C286. The 80C286 performs the actual data transfer with memory.

The Register Stack

The register stack contains eight 80-bit data registers, organized as a push down stack. Operations are performed on the stack top, between the stack top and another register, or between the stack top and memory.

System Configuration with 80C286

A simplified block diagram of the 80C287 interface to a 80C286 CPU is shown in Figure 1. The 80C287 can operate concurrently with the host CPU. The signals PEREQ, $\overline{PEACK}$, $\overline{BUSY}$, $\overline{NPRD}$, $\overline{NPWR}$, CMD_0, and CMD_1 allow the 80C287 to receive instructions and data from the 80C286. Detection of errors are indicated to the CPU by asserting the signal $\overline{ERROR}$. The address decode logic, bus control and timing logic is shown in this implementation using AMD PAL® devices but may also be accomplished using standard chip sets.

The 80C287 operates either directly from the CPU clock or with a dedicated clock. The 80C287 functions at two-thirds the frequency of the 80C286 when operating with the CPU clock (i.e., for a 16-MHz 80C286, the 32-MHz clock is divided down to 10.6 MHz).

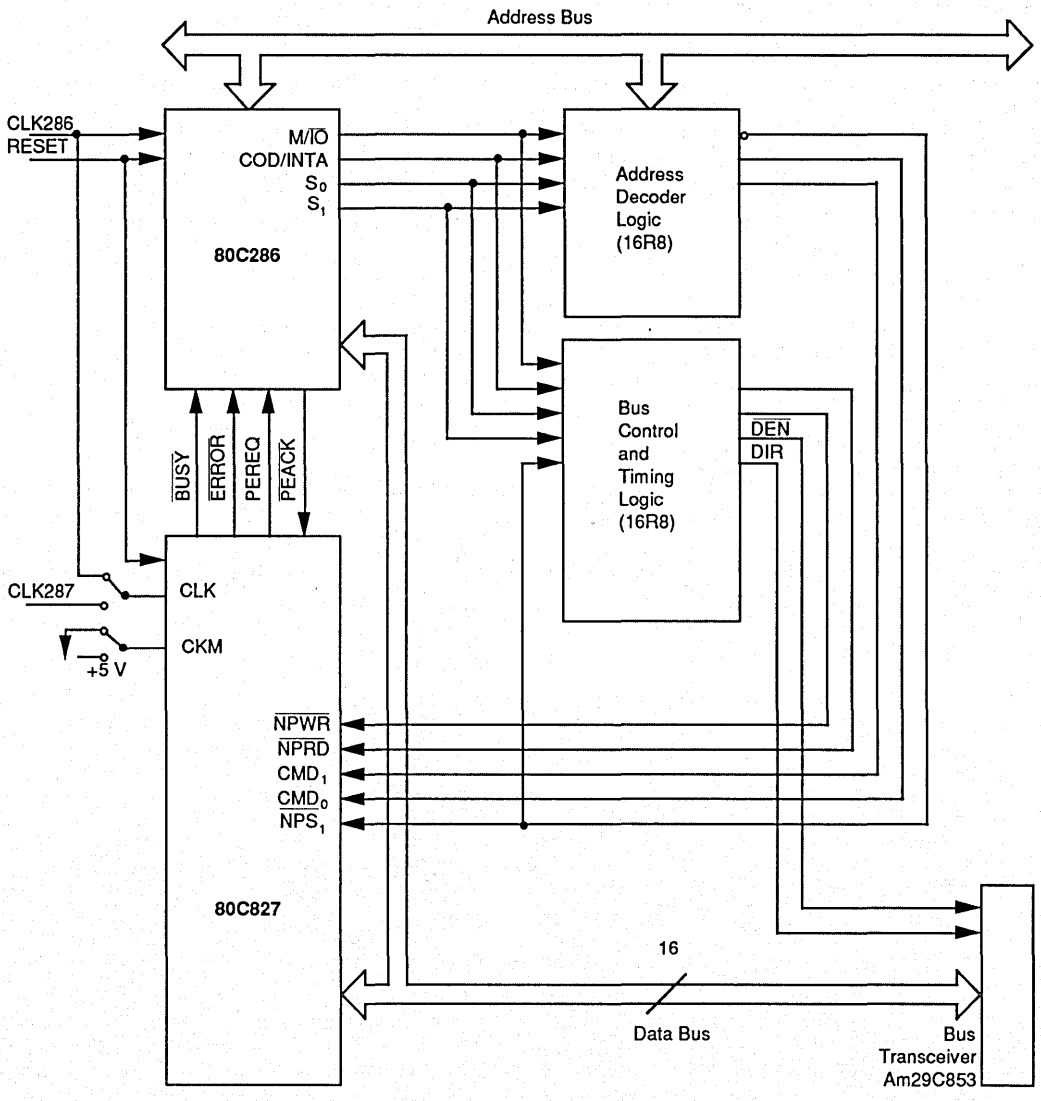


Figure 1. 80C286/80C287 Simplified System Configuration

11671-003A

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65 to +150° C
Ambient Temperature Under Bias ..... –55 to +125° C
Supply Voltage to Ground Potential
Continuous ..... –1.0 to +7.0 V
DC Voltage Applied to Outputs
for HIGH Output State ..... –0.3 V to + V_{CC} +0.3 V
DC Input Voltage ..... –0.3 to V_{CC} +0.3 V
DC Output Current, into LOW Outputs ..... 30 mA
DC Input Current ..... –10 to +10 mA
Power Dissipation (max.) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect devices reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature, Ambient (T_A) ..... 0 to +70°C
(also meets 0 to 100°C Case Temperature (T_C) for laptop requirements)
Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions		Min.	Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC} = Min. $V_{IN} = V_{IL}$ or V_{IH}	I_{OH} = -0.4 mA	2.4		V
V_{OL}	Output LOW Voltage	V_{CC} = Min. $V_{IN} = V_{IL}$ or V_{IH}	I_{OL} = 3 mA		0.45	
V_{IH}	Guaranteed Input Logical HIGH Voltage (see note below)			2.0	V_{CC} +0.5	V
V_{IL}	Guaranteed Input Logical LOW Voltage (see note below)			–0.5	0.8	V
V_{IHC}	Clock Input HIGH Voltage CKM = 1 CKM = 0			 2.0 3.8	 V_{CC} +1.0 V_{CC} +1.0	 V V
V_{ILC}	Clock Input Low Voltage CKM = 1 CKM = 0			 –0.5 –0.5	 0.8 0.6	 V V
I_{LI}	Input Leakage Current	0 V ≤ V_{IN} ≤ V_{CC}			±10	µA
I_{OZH}	Off-State (HIGH Impedance) Output Current	V_{CC} = Max., V_O = 2.4 V			10	µA
I_{OZL}	Off-State (HIGH Impedance) Output Current	V_{CC} = Max., V_O = 0.45 V			–10	µA
I_{CCD}	Power Supply Current, Operating	V_{CC} = Max. Outputs Unloaded		10 mA/MHz		
I_{CCS}	Power Supply Current, Static	V_{CC} = Max., $V_{IN} - V_{CC}$ or GND, I_O = OµA		5 mA		

Advance Information

Note: These input levels provide zero-noise immunity and should only be statically tested in a noise-free environment (not functionally tested).

SWITCHING CHARACTERISTICS over COMMERCIAL operating range

No.	Parameter Description	Test Conditions	80C287-8 Min.	80C287-8 Max.	Unit
1	Clock Period				
	CLM = 1		125		ns
	CLM = 0		50		ns
2	Clock LOW Time				
	CLM = 1		68		ns
	CLM = 0		15		ns
3	Clock HIGH Time				
	CLM = 1		43		ns
	CLM = 0		20		ns
4	Clock Rise Time			10	ns
5	Clock Fall Time			10	ns
6	Data Setup to $\overline{\text{NPWR}}$ Inactive		75		ns
7	Data Hold from $\overline{\text{NPWR}}$ Inactive		18		ns
8	$\overline{\text{NPWR}}$, $\overline{\text{NPRD}}$ Active Time		90		ns
9	Command Valid Setup Time		0		ns
10	PEREQ Active to $\overline{\text{NPRD}}$ Active		130		ns
11	$\overline{\text{PEACK}}$ Active Time		85		ns
12	$\overline{\text{PEACK}}$ Inactive Time		250		ns
13	$\overline{\text{PEACK}}$ Inactive to $\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ Inactive		40		ns
14	$\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ Inactive to $\overline{\text{PEACK}}$ Active		–30		ns
15	Command Valid Hold Time		30		ns
16	$\overline{\text{PEACK}}$ Active Setup to $\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$		40		ns
17	$\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ to CLK Setup		70		ns
18	$\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ CLK Hold		45		ns
19	RESET to CLK Setup		20		ns
20	RESET from CLK Hold		20		ns
21	$\overline{\text{NPRD}}$ Inactive to Data Float			35	ns
22	$\overline{\text{NPRD}}$ Active to Data Valid			60	ns
23	$\overline{\text{ERROR}}$ Active to $\overline{\text{BUSY}}$ Inactive		100		ns
24	$\overline{\text{NPWR}}$, Active to $\overline{\text{BUSY}}$ Active			100	ns
25	$\overline{\text{PEACK}}$ Active to $\overline{\text{PEREQ}}$ Inactive			127	ns
26	$\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ Active to $\overline{\text{PEREQ}}$ Inactive			100	ns
27	Command Inactive Time				
	Write to Write				ns
	Read to Read				ns
	Write to Read				ns
	Read to Write				ns
28	Data Hold from Time $\overline{\text{NPRD}}$ Inactive				ns

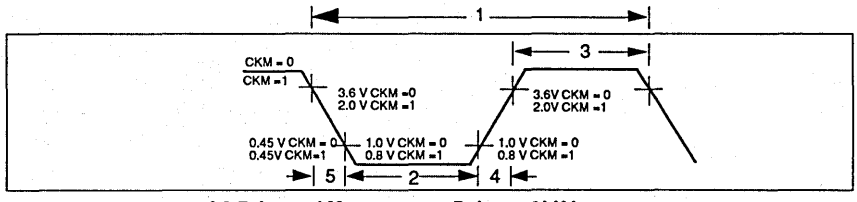
Advance Information

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

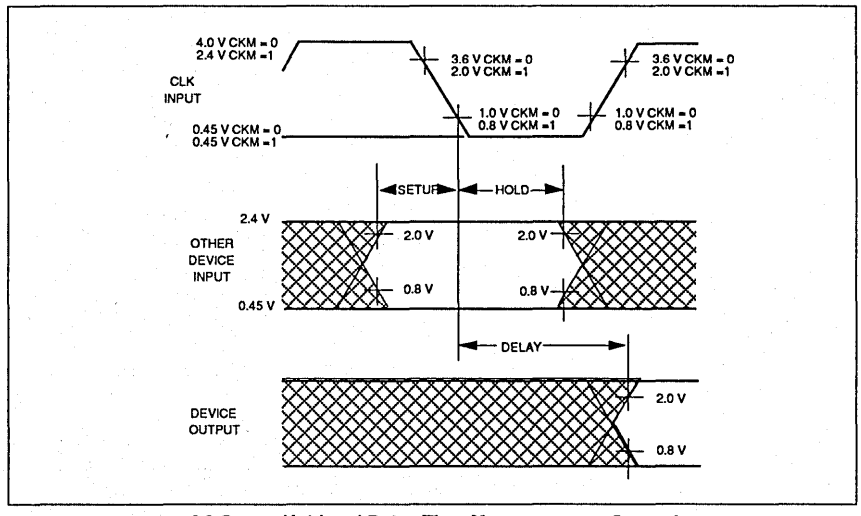
No.	Parameter Description	Test Conditions	80C287-10		80C287-12		80C287-16		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	Clock Period								
	CLM = 1		100		80		62.5		ns
	CLM = 0		40		35		30		ns
2	Clock LOW Time								
	CLM = 1		62		50		37		ns
	CLM = 0		18		13		8		ns
3	Clock HIGH Time								
	CLM = 1		28		22		17		ns
	CLM = 0		18		13		12		ns
4	Clock Rise Time			10		8		4	ns
5	Clock Fall Time			10		8		4	ns
6	Data Setup to NPWR Inactive		75		75		60		ns
7	Data Hold from NPWR Inactive		18		10		10		ns
8	NPWR, NPRD Active Time		90		70		50		ns
9	Command Valid Setup Time		0		0		0		ns
10	PEREQ Active to NPRD Active		100		80		62		ns
11	PEACK Active Time		60		50		36		ns
12	PEACK Inactive Time		200		160		125		ns
13	PEACK Inactive to NPRD, NPWR Inactive		40		32		25		ns
14	NPRD, NPWR Inactive to PEACK Active		–30		–30		–30		ns
15	Command Valid Hold Time		22		18		15		ns
16	PEACK Active Setup to NPRD, NPWR		40		30		30		ns
17	NPRD, NPWR to CLK Setup		53		40		30		ns
18	NPRD, NPWR CLK Hold		37		29		22		ns
19	RESET to CLK Setup		20		20		20		ns
20	RESET from CLK Hold		20		20		20		ns
21	NPRD Inactive to Data Float			21		17		13	ns
22	NPRD Active to Data Valid			60		50		40	ns
23	ERROR Active to BUSY Inactive		100		100		100		ns
24	NPWR, Active to BUSY Active			100		80		60	ns
25	PEACK Active to PEREQ Inactive			100		80		60	ns
26	NPRD, NPWR Active to PEREQ Inactive			100		80		60	ns
27	Command Inactive Time								
	Write to Write		75		60		50		ns
	Read to Read		75		60		50		ns
	Write to Read		75		60		50		ns
	Read to Write		75		60		50		ns
28	Data Hold from Time NPRD Inactive		3		1		1		ns

Advance Information

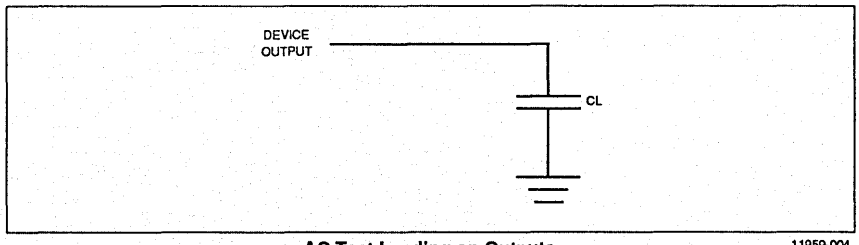
SWITCHING WAVEFORMS



AC Drive and Measurement Points—CLK Input

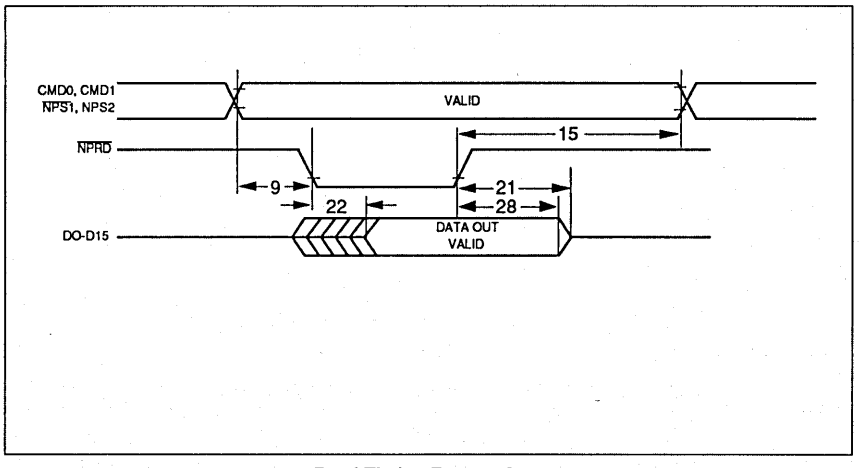


AC Setup, Hold and Delay Time Measurement—General

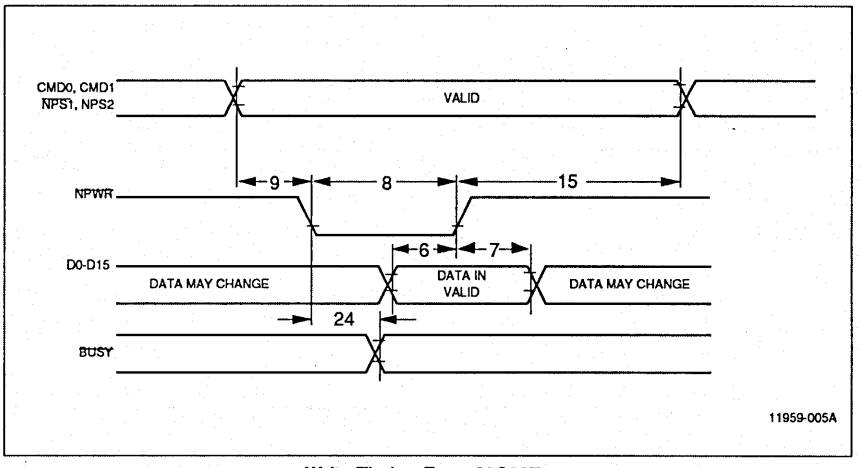


AC Test Loading on Outputs

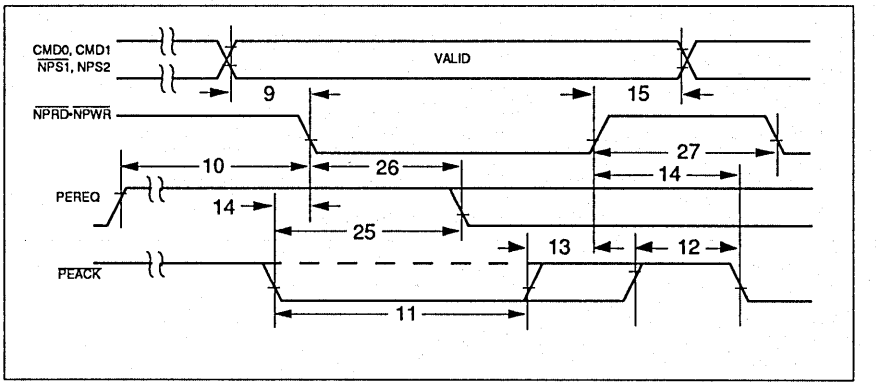
11959-004



Read Timing From 80C287

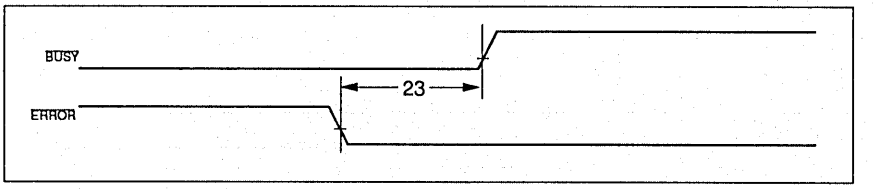


Write Timing From 80C287

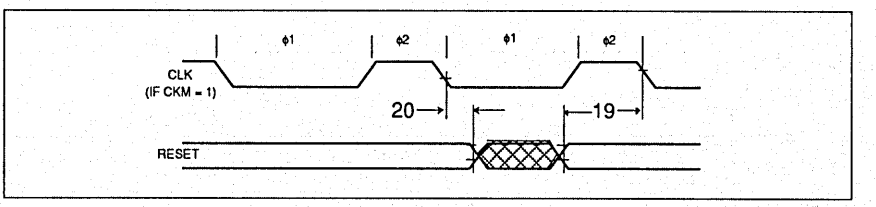


Data Channel Timing (Initiated by 80C287)

11959-006A



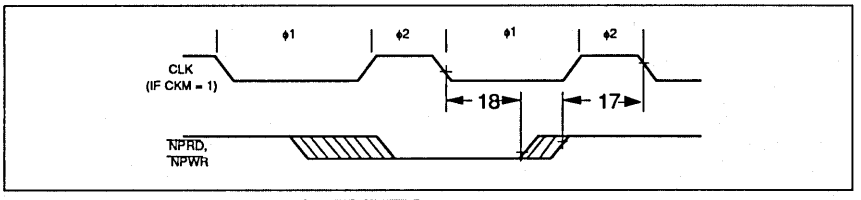
Error Output Timing



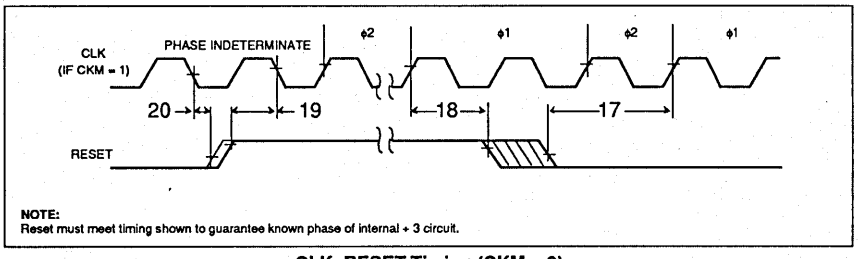
11959-007A

Clk, Reset Timing (Ckm = 1)

NOTE:
Reset, $\overline{NPWR}$, $\overline{NPRD}$ are inputs asynchronous to CLK. Timing requirements above are given for testing purposes only, to assure recognition at a specific CLK edge.

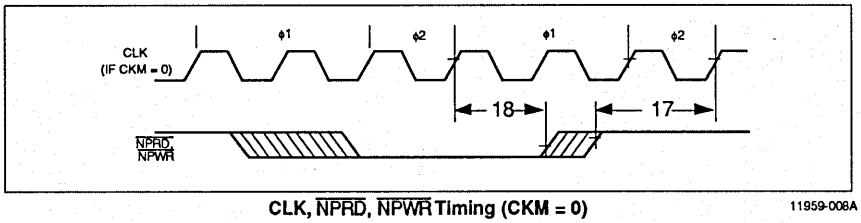


CLK, $\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ Timing (CKM = 1)



NOTE:
Reset must meet timing shown to guarantee known phase of internal ÷ 3 circuit.

CLK, RESET Timing (CKM = 0)



CLK, $\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ Timing (CKM = 0)

11959-008A

Advance Information

AMD 80EC287

Enhanced 80-Bit CMOS Numeric Processor

Advanced Micro Devices

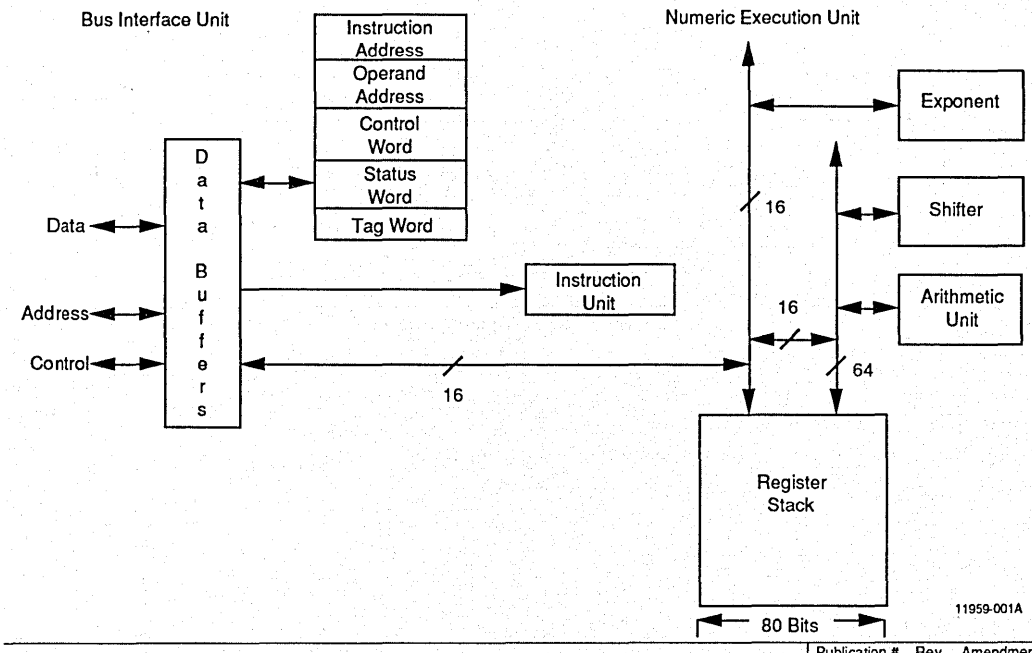
DISTINCTIVE CHARACTERISTICS

- **Pin compatible and functionally equivalent to the Intel 80287**
- **High-performance CMOS process yields 10-MHz, 12-MHz, and 16-MHz speed grades**
- **Enhanced sleep feature automatically shuts off the internal clock when no instruction is executing, reducing power consumption. This feature is transparent to the user**
- **Available in space-saving 44-pin PLCC as well as 40-pin DIP**
- **80-bit numeric accelerator for 80C286 and 80286-based systems**
- **Compatible with IEEE floating-point standard 754**
- **Static CMOS design does not require a minimum clock rate, resulting in significantly lower power dissipation**
- **Performs single-, double-, and extended-precision floating-point, as well as word, short, and long integer and 18-digit BCD conversions**
- **Adds trigonometric, logarithmic, exponential, and arithmetic instructions to the 80C286 instruction set**

GENERAL DESCRIPTION

The 80EC287 is implemented in AMD's advanced static CMOS process that allows for significantly higher speeds at a much lower power dissipation than traditional NMOS versions or standard CMOS. The 80EC287 is a high-performance arithmetic processor that expands the 80C286 instruction set with floating-point instructions including transcendentals, and integer and BCD conversions. The 80EC287 is functionally equivalent to the Intel 80287 and AMD 80C287 plus adds a low power sleep feature for battery powered applications. This enhanced 80EC287 can be a direct replacement for an AMD 80C287. The sleep feature is an automatic inherent feature of the device and thus requires no external entry. The floating-point operations comply with the IEEE Standard 754. The device is available in 12- and 16-MHz speed grades and is provided in 44-pin PLCC and 40-pin DIP packages. When coupled with the 80C286, the 80EC287 provides a complete solution for high-performance numeric processing applications.

BLOCK DIAGRAM

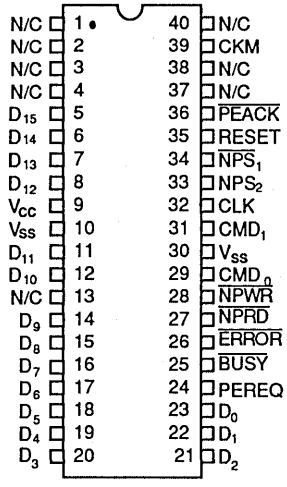


Publication #	Rev.	Amendment
11959	B	/0

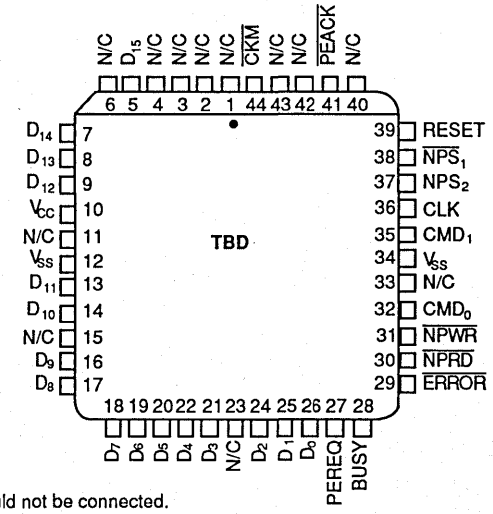
Issue Date: June 1989

CONNECTION DIAGRAM

DIP



PLCC



Note: N/C pins should not be connected.
Pin 1 is marked for orientation.

11959-002A

1

PIN DESCRIPTION

$\overline{BUSY}$ $\overline{Busy\ Status}$ (Output; Active Low)

A LOW level indicates that the 80EC287 is currently executing a command.

CKM Clock Mode Signal (Input)

When CKM is HIGH, the CLK is used directly. When CKM is LOW, CLK is divided by three. This input must be either HIGH or LOW 20 CLK cycles before RESET goes LOW.

CLK Clock (Input)

Provides timing for 80EC287 operations.

CMD_1, CMD_0 Command Lines (Input)

CMD_1 and CMD_0, along with select inputs, allow the CPU to direct the 80EC287 operations. These inputs are timed relative to the read and write strobes.

D_{15}–D_0 Data (Input/Output)

Bidirectional data bus. These inputs are timed relative to the read and write strobes.

$\overline{ERROR}$ $\overline{Error\ Status}$ (Output; Active Low)

Reflects the error summary status bit of the status word. A LOW level indicates that an unmasked exception condition exists.

$\overline{NPRD}$ $\overline{Numeric\ Processor\ Read}$ (Input; Active Low)

A LOW level enables transfer of data from the 80EC287. This input may be asynchronous to the 80EC287 clock.

$\overline{NPS_1}$, NPS_2 Numeric Processor Selects (Input)

Indicates the CPU is transferring data to and from the 80EC287. Asserting both signals ($\overline{NPS_1}$ LOW and NPS_2 HIGH) enables the 80EC287 to transfer floating-point data or instructions. No data transfers involving the 80EC287 will occur unless the 80EC287 is selected via $\overline{NPS_1}$ and NPS_2. These inputs are timed relative to the read and write strobes.

$\overline{NPWR}$ $\overline{Numeric\ Processor\ Write}$ (Input; Active Low)

A LOW level enables transfer of data from the 80EC287. This input may be asynchronous to the 80EC287 clock.

$\overline{PEACK}$ $\overline{Processor\ Extension\ Acknowledge}$ (Input; Active Low)

A LOW level indicates that the request signal (PEREQ) has been recognized. $\overline{PEACK}$ causes the request (PEREQ) to be withdrawn when no more transfers are required. $\overline{PEACK}$ may be asynchronous to the 80EC287 clock.

PEREQ Processor Extension Request (Output)

A HIGH level indicates that the 80EC287 is ready to transfer data. PEREQ will be disabled upon assertion of $\overline{PEACK}$ or upon actual data transfer, whichever occurs first, when no more transfers are required.

RESET System Reset (Input)

Reset causes the 80EC287 to immediately terminate its present activity and enter a dormant state. Reset must be HIGH for more than four CLK cycles. For proper initialization the HIGH-LOW transition must occur no sooner than 50 µs after V_{CC} and CLK meet their DC and AC specifications.

V_{CC} +5 V Supply (Input)

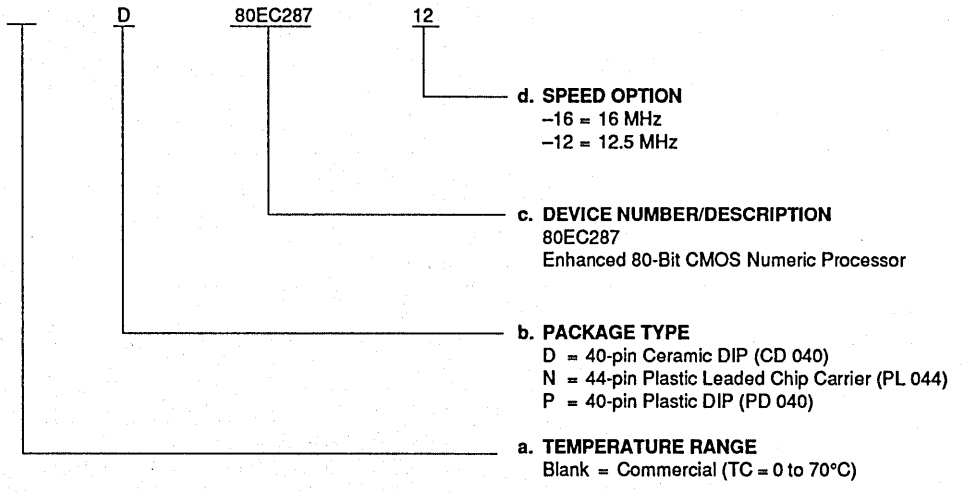
V_{SS} System Ground (Input)

Both pins must be connected to ground.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Temperature Range
b. Package Type
c. Device Number
d. Speed Option (if applicable)
e. Optional Processing



Valid Combination	
D, N, P	80EC287–16
	80EC287–12

Valid Combination

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released valid combinations.

SIMPLIFIED FUNCTIONAL DESCRIPTION

The 80EC287 is internally divided into two basic processing units; the numeric execution unit, and the bus interface unit as shown in the block diagram. The numeric execution unit performs numeric instructions. The bus interface unit receives and decodes instructions, executes processor control instructions, and requests operands transfers to and from memory. The 80C286 may execute non-numeric instruction concurrently with numeric instruction executed on the 80EC287. Synchronization and error recognition occurs when the next numeric instruction is decoded by the 80C286.

The Numeric Execution Unit

The numeric execution data path is 80 bits wide. All operands are converted to the internal 80-bit format before use. These instructions include arithmetic, transcendental, constant, and data transfer instructions.

The Bus Interface Unit

The bus interface unit decodes the ESC instruction executed by the 80C286. The signal $\overline{BUSY}$ is activated for 80C286/80EC287 synchronization and the signal $\overline{ERROR}$ is activated for error detection. $\overline{BUSY}$ is activated when an instruction is transferred and deactivated when the instruction completes. $\overline{ERROR}$ will be asserted if an error has occurred when $\overline{BUSY}$ is deactivated.

The signals PEREQ, $\overline{PEACK}$, $\overline{NPRD}$, $\overline{NPWR}$, $\overline{NPS_1}$, CMD_0, CMD_1, and NPS_2 control data transfers between the 80EC287 and the 80C286. The 80C286 performs the actual data transfer with memory.

The Register Stack

The register stack contains eight 80-bit data registers, organized as a push down stack. Operations are performed on the stack top, between the stack top and another register, or between the stack top and memory.

System Configuration with 80C286

A simplified block diagram of the 80EC287 interface to a 80C286 CPU is shown in Figure 1. The 80EC287 can operate concurrently with the host CPU. The signals PEREQ, $\overline{PEACK}$, $\overline{BUSY}$, $\overline{NPRD}$, $\overline{NPWR}$, CMD_0, and CMD_1 allow the 80EC287 to receive instructions and data from the 80C286. Detection of errors are indicated to the CPU by asserting the signal $\overline{ERROR}$. The address decode logic, bus control and timing logic is shown in this implementation using AMD PAL® devices but may also be accomplished using standard chip sets.

The 80EC287 operates either directly from the CPU clock or with a dedicated clock. The 80EC287 functions at two-thirds the frequency of the 80C286 when operating with the CPU clock (i.e., for a 16-MHz 80C286, the 32-MHz clock is divided down to 10.6 MHz).

Sleep Feature

The 80EC287 clock runs only while an instruction is executing. The internal clock shuts itself off when no instruction is executing, thus reducing power consumption. This feature is completely transparent to the user and requires no external circuitry or design interface.

The 80EC287 is completely static. For absolute minimum power consumption, lower than that of the sleep feature, the external clock can be stopped in phase 2.

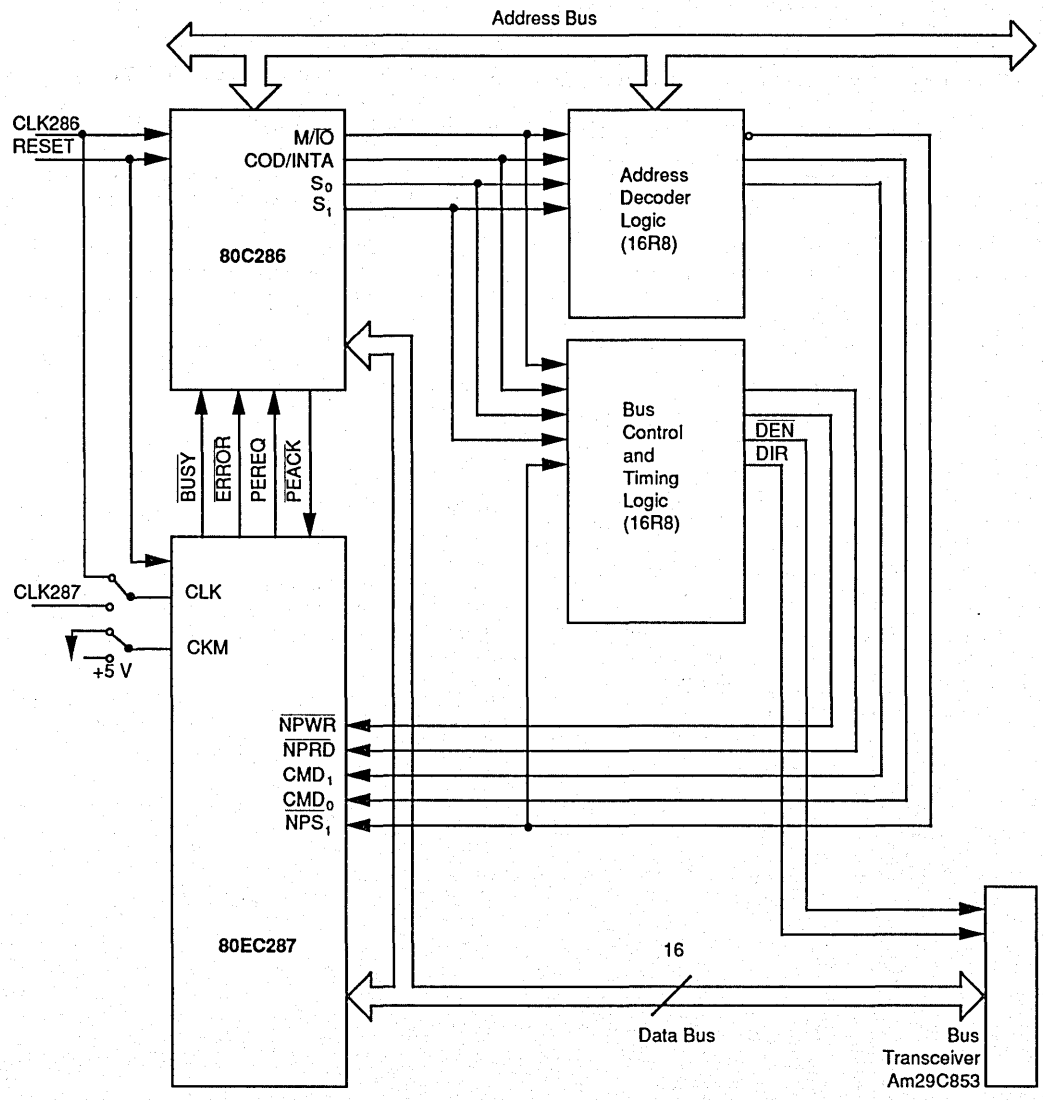


Figure 1. 80C286/80EC287 Simplified System Configuration

11959-003A

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65 to +150° C
Ambient Temperature Under Bias ..... –55 to +125° C
Supply Voltage to Ground Potential
Continuous ..... –1.0 to +7.0 V
DC Voltage Applied to Outputs
for HIGH Output State ..... –0.3 V to + V_{CC} +0.3 V
DC Input Voltage ..... –0.3 to V_{CC} +0.3 V
DC Output Current, into LOW Outputs ..... 30 mA
DC Input Current ..... –10 to +10 mA
Power Dissipation (max.) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect devices reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature, Ambient (T_A) ..... 0 to +70°C
(also meets 0 to 100°C Case Temperature (T_C) for laptop requirements)
Supply Voltage (V_{CC}) ..... + 4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions		Min.	Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC} = Min. V_{IN} = V_{IL} or V_{IH}	I_{OH} = -0.4 mA	2.4		V
V_{OL}	Output LOW Voltage	V_{CC} = Min. V_{IN} = V_{IL} or V_{IH}	I_{OL} = 3 mA		0.45	
V_{IH}	Guaranteed Input Logical HIGH Voltage (Note 1)			2.0	V_{CC} +0.5	V
V_{IL}	Guaranteed Input Logical LOW Voltage (Note 1)			–0.5	0.8	V
V_{IHC}	Clock Input HIGH Voltage CKM = 1 CKM = 0			 2.0 3.8	 V_{CC} +1.0 V_{CC} +1.0	 V V
V_{ILC}	Clock Input Low Voltage CKM = 1 CKM = 0			 –0.5 –0.5	 0.8 0.6	 V V
I_{LI}	Input Leakage Current	0 V ≤ V_{IN} ≤ V_{CC}			±10	µA
I_{OZH}	Off-State (HIGH Impedance) Output Current	V_{CC} = Max., V_O = 2.4 V			10	µA
I_{OZL}	Off-State (HIGH Impedance) Output Current	V_{CC} = Max., V_O = 0.45 V			–10	µA
I_{CCS}	Power Supply Current, static	V_{CC} = Max., V_{IN}-V_{CC} or GND, I_O=0 µA		5 mA		
I_{CCD}	Supply Current, operating	V_{CC}=Max. Outputs Unloaded		10 mA/MHz (Note 2)		
Icc_{SM}	Power Supply Current, Sleep Mode	V_{CC} = Max. Outputs Unloaded		1 mA/MHz		

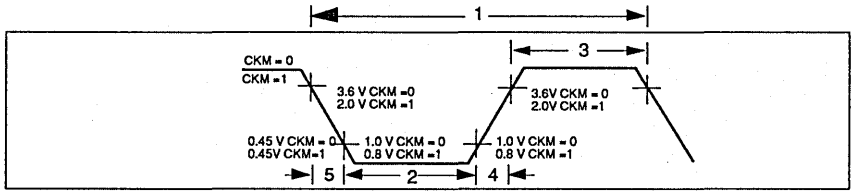
Notes: 1. These input levels provide zero-noise immunity and should only be statically tested in a noise-free environment (not functionally tested).
2. This reduces to I_{CCSM} when no instruction is executing, reducing overall power consumption.

Advance Information

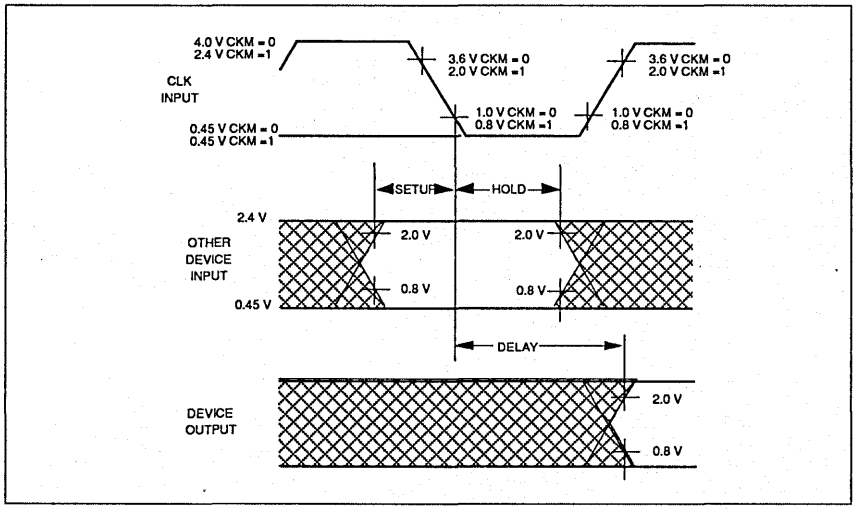
SWITCHING CHARACTERISTICS over COMMERCIAL operating range

No.	Parameter Description	Test Conditions	80EC287-12		80EC287-16		Unit
			Min.	Max.	Min.	Max.	
1	Clock Period						
	CLM = 1		80	00	62.5	00	ns
	CLM = 0		35		30		ns
2	Clock LOW Time						
	CLM = 1		50		37		ns
	CLM = 0		9		8		ns
3	Clock HIGH Time						
	CLM = 1		22		17		ns
	CLM = 0		13		12		ns
4	Clock Rise Time			8		4	ns
5	Clock Fall Time			8		4	ns
6	Data Setup to NPWR Inactive		75		60		ns
7	Data Hold from NPWR Inactive		10		10		ns
8	NPWR, NPRD Active Time		70		50		ns
9	Command Valid Setup Time		0		0		ns
10	PEREQ Active to NPRD Active		80		62		ns
11	PEACK Active Time		50		36		ns
12	PEACK Inactive Time		160		125		ns
13	PEACK Inactive to NPRD, NPWR Inactive		32		25		ns
14	NPRD, NPWR Inactive to PEACK Active		30		–30		ns
15	Command Valid Hold Time		18		15		ns
16	PEACK Active Setup to NPRD, NPWR		30		30		ns
17	NPRD, NPWR to CLK Setup		40		30		ns
18	NPRD, NPWR CLK Hold		29		22		ns
19	RESET to CLK Setup		20		20		ns
20	RESET from CLK Hold		20		20		ns
21	NPRD Inactive to Data Float			17		13	ns
22	NPRD Active to Data Valid			50		40	ns
23	ERROR Active to BUSY Inactive		100		100		ns
24	NPWR, Active to BUSY Active			80		60	ns
25	PEACK Active to PEREQ Inactive			80		60	ns
26	NPRD, NPWR Active to PEREQ Inactive			80		60	ns
27	Command Inactive Time						
	Write to Write		60		50		ns
	Read to Read		60		50		ns
	Write to Read		60		50		ns
	Read to Write		60		50		ns
28	Data Hold from Time NPRD Inactive		1		1		ns

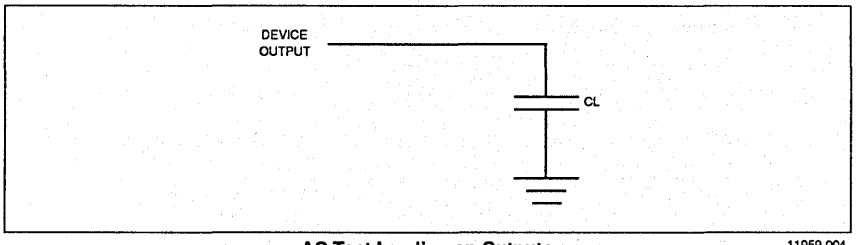
Advance Information



AC Drive and Measurement Points—CLK Input

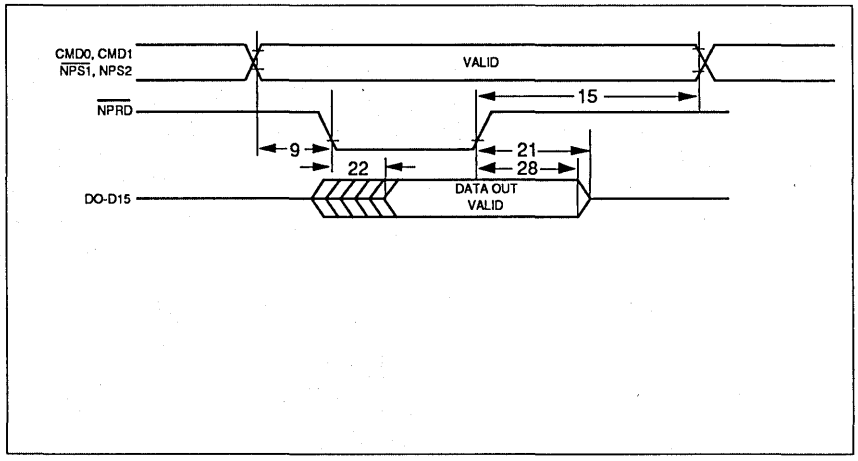


AC Setup, Hold and Delay Time Measurement—General

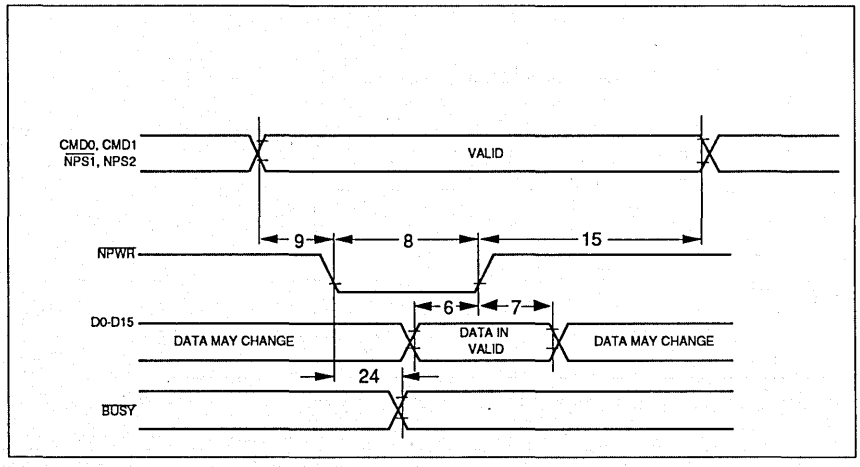


AC Test Loading on Outputs

11959-004

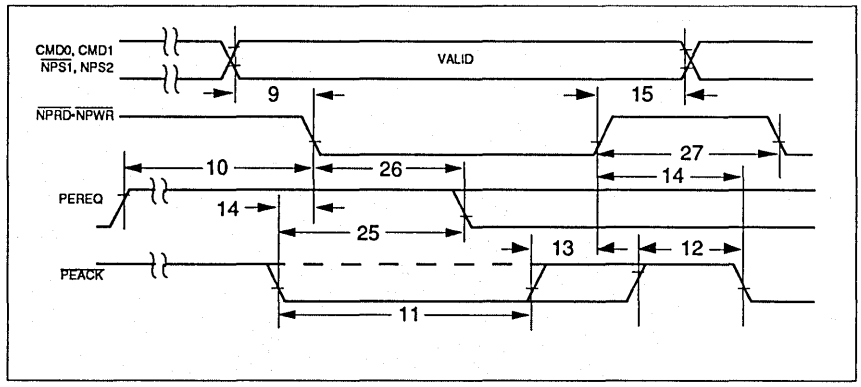


Read Timing From 80EC287



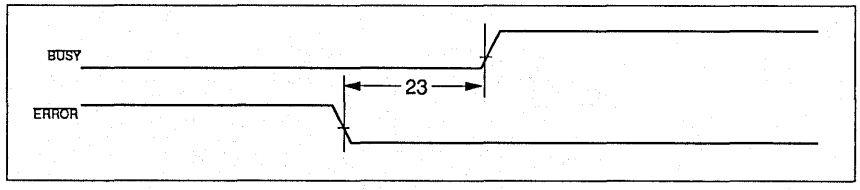
11959-005A

Write Timing From 80EC287

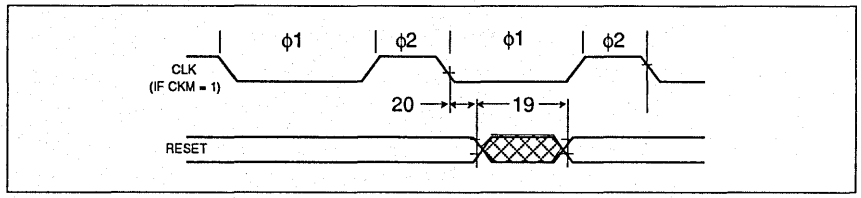


Data Channel Timing (Initiated by 80EC287)

11959-006A



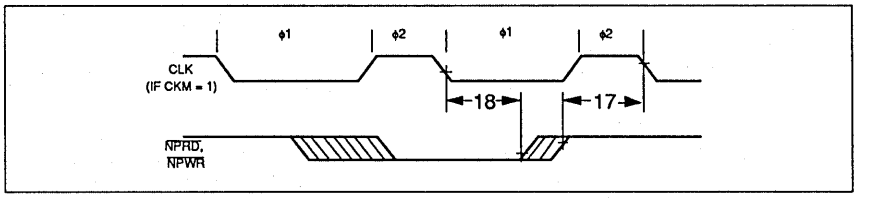
Error Output Timing



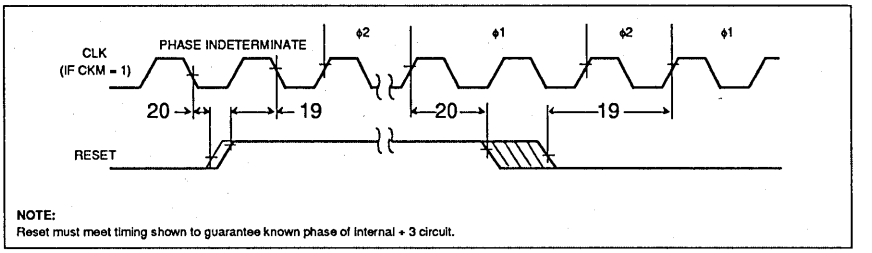
11959-007A

CLK, Reset Timing (CKM = 1)

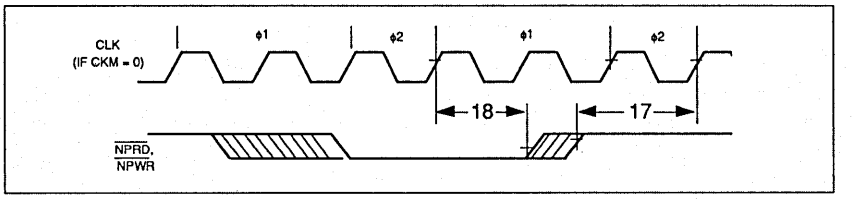
NOTE: Reset, $\overline{\text{NPWR}}$, $\overline{\text{NPRD}}$ are inputs asynchronous to CLK. Timing requirements above are given for testing purposes only, to assure recognition at a specific CLK edge.



CLK, $\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ TIMING (CKM = 1)



CLK, RESET TIMING (CKM = 0)



CLK, $\overline{\text{NPRD}}$, $\overline{\text{NPWR}}$ TIMING (CKM = 0)

Am9517A/8237A*

Multimode DMA Controller

FINAL

DISTINCTIVE CHARACTERISTICS

- Four independent DMA channels, each with separate registers for Mode Control, Current Address, Base Address, Current Word Count and Base Word Count
- Transfer modes: Block, Demand, Single Word, Cascade
- Independent Autoinitialization of all channels
- Memory-to-memory transfers
- Memory block initialization
- Address increment or decrement
- Master system disable
- Enable/disable control of individual DMA requests
- Directly expandable to any number of channels
- End of Process input for terminating transfers
- Software DMA requests
- Independent polarity control for DREQ and DACK signals
- Compressed timing option speeds transfers — up to 2.5M bytes/second
- +5 volt power supply
- N-channel silicon gate MOS technology
- 40-pin Hermetic DIP package, 44-pin PLCC package
- 9517A-5 5 MHz version for higher speed CPU compatibility

GENERAL DESCRIPTION

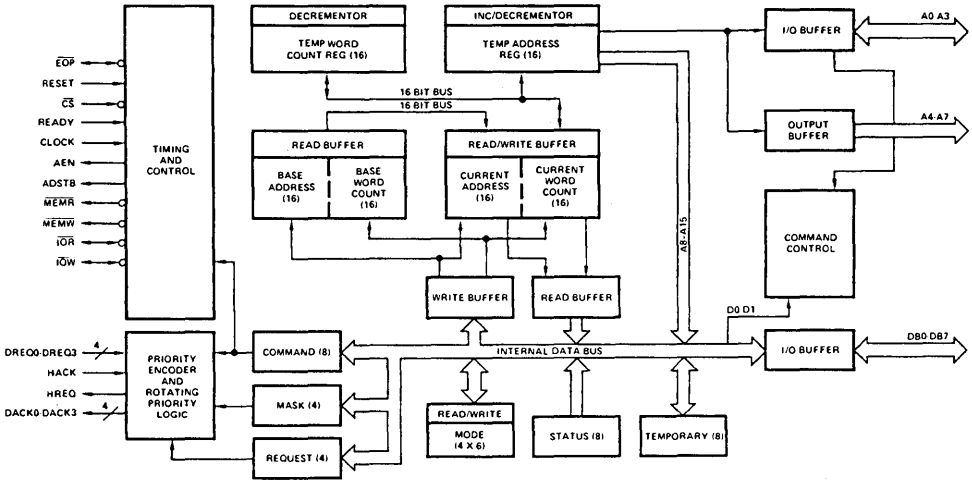
The Am9517A/8237A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information to or from the system memory. Memory-to-memory transfer capability is also provided. The Am9517A/8237A offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The Am9517A/8237A is designed to be used in conjunction with an external 8-bit address register such as the Am74LS373. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips.

The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process ($\overline{EOP}$).

Each channel has a full 64K address and word count capability. An external $\overline{EOP}$ signal can terminate a DMA or memory-to-memory transfer. This is useful for block search or compare operations using external comparators or for intelligent peripherals to abort erroneous services.

BLOCK DIAGRAM



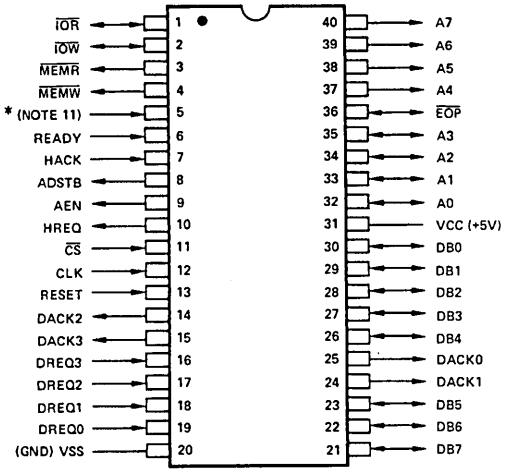
BD003250

*The 8237A is an AMD-invented device more commonly referred to as the Am9517A.

Publication # 03040 Rev. D Amendment /0
Issue Date: August 1989

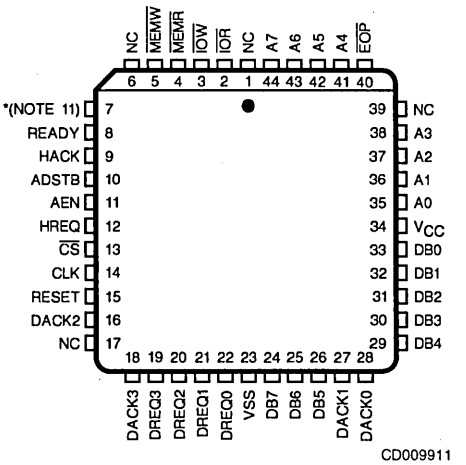
CONNECTION DIAGRAMS
Top View

DIPs



CD005072

PLCC



CD009911

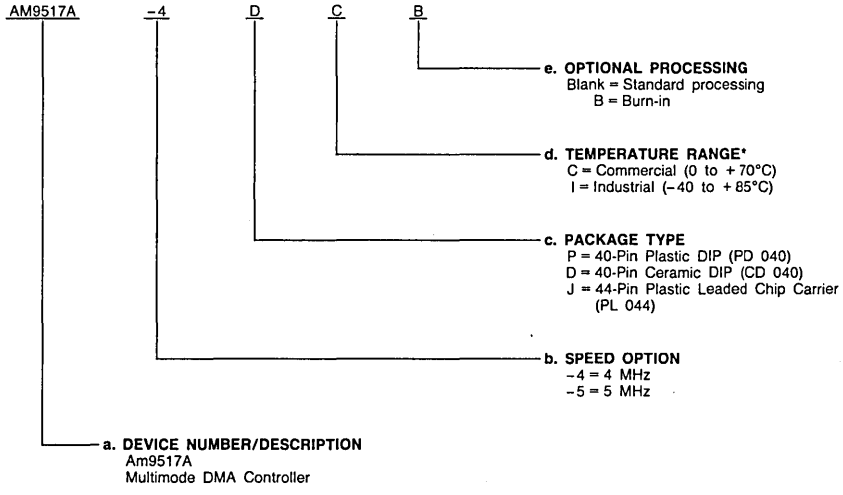
Note: Pin 1 is marked for orientation.
*See Note 11 under DC Characteristics table.

ORDERING INFORMATION

Am9517A

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9517A-4	DC, DCB, DIB, PC
AM9517A-5	DC, DCB, PC, JC

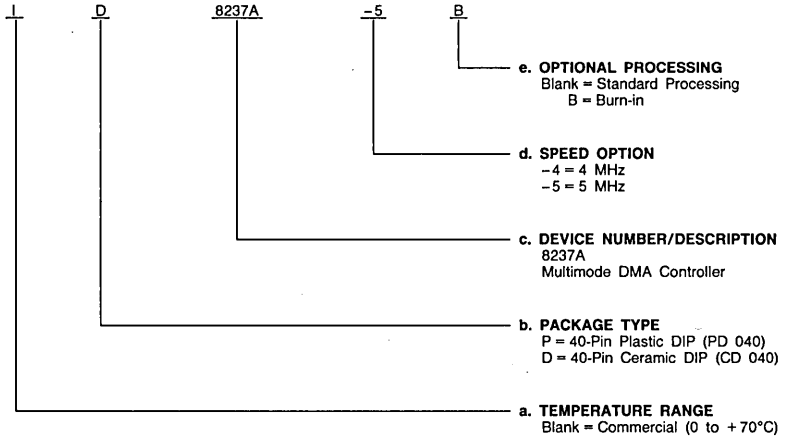
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION (continued)

8237A

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option
e. Optional Processing



Valid Combinations	
8237A-4	P, D
8237A-5	
8237A-4B	D
8237A-5B	

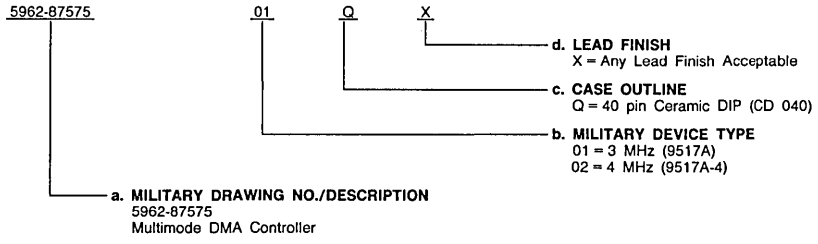
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION (continued)

Standard Military Drawing (SMD)/DESC Products

AMD standard products for Aerospace and Defense applications are available in several packages and operating ranges. Standard Military Drawing (SMD)/DESC products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for SMD/DESC products is formed by a combination of: **a. Military Drawing Part Number**
b. Device Type
c. Case Outline
d. Lead Finish



Valid Combinations	
5962-8757501	QX
5962-875702	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations.

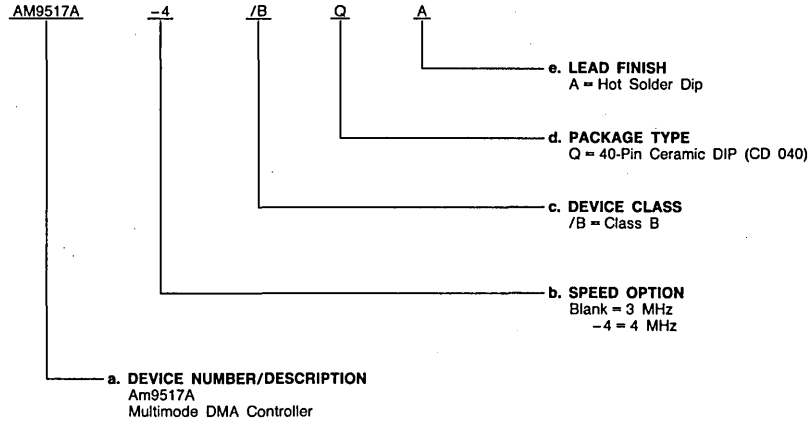
Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

ORDERING INFORMATION (continued)

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM9517A	/BQA
AM9517A-4	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

Pin No.*	Name	I/O	Description
31	V_{CC}		Power: +5 volt supply.
20	V_{SS}		Ground.
12	CLK	I	Clock Input: Clock Input controls the internal operations of the Am9517A/8237A and its rate of data transfers. The input may be driven at up to 3MHz for the standard Am9517A/8237A and up to 5 MHz for the Am9517A-5/8237A-5.
11	$\overline{CS}$	I	Chip Select: Chip Select is an active low input used to select the Am9517A/8237A as an I/O device during the Idle cycle.This allows CPU communication on the data bus.
13	RESET	I	Reset: Reset is an active high input which clears the Command, Status, Request and Temporary registers. It also clears the First/Last Flip/Flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
6	READY	I	Ready: Ready is an input used to extend the memory read and write pulses from the Am9517A/8237A to accommodate slow memories or I/O peripheral devices. Ready must not make transitions during its specified setup/hold time.
7	HACK	I	Hold Acknowledge: The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system buses.
19-16	DREQ0-DREQ3	I	DMA Request: The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In Fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.
30-26, 23-21	DB0-DB7	I/O	DATA Bus: The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the Am9517A/8237A control registers. During DMA cycles the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the Am9517A/8237A on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.
1	$\overline{IOR}$	I/O	I/O Read: I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the Am9517A/8237A to access data from a peripheral during a DMA Write transfer.
2	$\overline{IOW}$	I/O	I/O Write: I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the Am9517A/8237A. In the Active cycle, it is an output control signal used by the Am9517A/8237A to load data to the peripheral during a DMA Read transfer.
36	$\overline{EOP}$	I/O	End of Process: End of Process is an active low bidirectional open-drain signal. Information concerning the completion of DMA service is available at the bidirectional $\overline{EOP}$ pin. The Am9517A/8237A allows an external signal to terminate an active DMA service. This is accomplished by pulling the $\overline{EOP}$ input low with an external $\overline{EOP}$ signal. The Am9517A/8237A also generates a pulse when the terminal count (TC) for any channel is reached. This generates an $\overline{EOP}$ signal which is output through the $\overline{EOP}$ Line. The reception of $\overline{EOP}$, either internal or external, will cause the Am9517A/8237A to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by $\overline{EOP}$ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains unchanged. During memory-to-memory transfers, $\overline{EOP}$ will be output when the TC for channel 1 occurs. $\overline{EOP}$ should be tied high with a pull-up resistor if it is not used to prvent erroneous end of process inputs.
32-35	A0-A3	I/O	Address: The four least significant address lines are bidirectional three-state signals. In the Idle cycle, they are inputs and are used by the CPU to address the registers to be load or read. In the Active cycle, they are outputs and provide the lower 4 bits of the output address.
37-40	A4-A7	O	Address: The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during DMA service.
10	HREQ	O	Hold Request: This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes the Am9517A/8237A to issue the HRQ. After HRQ goes active, at least one clock cycle (TCY) must occur before HLDA goes active.
25, 24 14, 15	DACK0-DACK3	O	DMA Acknowledge: DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.
9	AEN	O	Address Enable. Address Enable enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable in other system bus drivers during DMA transfers. AEN is active-high.
8	ADSTB	O	Address Strobe. The active-high Address Strobe is used to strobe the upper address byte into an external latch.
3	$\overline{MEMR}$	O	Memory Read: The Memory Read signal is an active low three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.
4	$\overline{MEMW}$	O	Memory Write: The Memory Write signal is an active low three-state output used to write data to the selected memory location during a DMA Write or a a memory-to-memory transfer.

*Applies to DIPs only.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Am9517/8237A Internal Registers.

DETAILED DESCRIPTION

The Am9517A/8237A block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The Am9517A/8237A contains 344 bits of internal memory in the form of registers. The table shown above lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

The Am9517A/8237A contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the Am9517A/8237A. The Program Command Control block decodes the various commands given to the Am9517A/8237A by the microprocessor prior to servicing a DMA Request. It also decodes each channel's Mode Control word. The Priority Encoder block resolves priority contention among DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In Am9080A systems this input will usually be the $\phi 2$ TTL clock from an Am8224. However, any appropriate system clock will suffice.

DMA Operation

The Am9517A/8237A is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The Am9517A/8237A can assume seven separate states, each composed of one full clock period. State I (SI) is the inactive state. It is entered when the Am9517A/8237A has no valid DMA requests pending. While in SI, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State 0 (S0) is the first state of a DMA service. The Am9517A/8237A has requested a hold but the processor has not yet returned an acknowledge. An acknowledge from the CPU will signal that transfers may begin. S1, S2, S3, and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted before S4 by the use of the Ready line on the Am9517A/8237A.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for each complete transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23 and S24) for the write-to-memory half of the transfer. The Temporary Data register is used for intermediate storage of the memory byte.

Idle Cycle

When no channel is requesting service, the Am9517A/8237A will enter the Idle cycle and perform "SI" states. In this cycle the Am9517A/8237A will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample $\overline{CS}$, looking for an attempt by the microprocessor to write or read the internal registers of the Am9517A/8237A. When $\overline{CS}$ is LOW and HACK is LOW, the Am9517A/8237A enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0 – A3 are inputs to the device and select which registers will be read or written. The $\overline{IOR}$ and $\overline{IOW}$ lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip/flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip/flop is reset by Master Clear or Reset. A separate software command can also reset this flip/flop.

Special software commands can be executed by the Am9517A/8237A in the Program Condition. These commands are decoded as sets of addresses when both $\overline{CS}$ and $\overline{IOW}$ are active and do not make use of the data bus. Functions include Clear First/Last Flip/Flop and Master Clear.

Active Cycle

When the Am9517A/8237A is in the idle cycle and a channel requests a DMA service, the device will output a HREQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place in one of four modes:

Single Transfer Mode: In Single Transfer mode, the Am9517A/8237A will make a one-byte transfer during each HREQ/HACK handshake. When DREQ goes active, HREQ will go active. After the CPU responds by driving HACK active, a one-byte transfer will take place. Following the transfer, HREQ will go inactive, the word count will be decremented and the address will be either incremented or decremented. When the word count goes to zero, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

To perform a single transfer, DREQ must be held active only until the corresponding DACK goes active. If DREQ is held continuously active, HREQ will go inactive following each transfer and then will go active again and a new one-byte transfer will be made following each rising edge of HACK. In 8080A/Am9080A systems, this will ensure one full machine cycle of execution between DMA transfers. Details of timing between the Am9517A/8237A and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode: In Block Transfer mode, the Am9517A/8237A will continue making transfers until a TC (caused by the word count going to zero) or an external End of Process ($\overline{EOP}$) is encountered. DREQ need be held active only until DACK becomes active. An Autoinitialize will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode: In Demand Transfer mode the device will continue making transfers until a TC or external $\overline{EOP}$ is encountered or until DREQ goes inactive. Thus, the device requesting service may discontinue transfers by bringing DREQ inactive. Service may be resumed by asserting an active DREQ once again. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count may be read from the Am9517A/8237A Current Address and Current Word Count

registers. Autoinitialization will only occur following a TC or $\overline{EOP}$ at the end of service. Following Autoinitialization, an active-going DREQ edge is required to initiate a new DMA service.

Cascade Mode: This mode is used to cascade more than one Am9517A/8237A together for simple system expansion. The HREQ and HACK signals from the additional Am9517A/8237A are connected to the DREQ and DACK signals of a channel of the initial Am9517A/8237A. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel in the initial device is used only for prioritizing the additional device, it does not output any address or control signals of its own. These would conflict with the outputs of the active channel in the added device. The Am9517A/8237A will respond to DREQ with DACK but all other outputs except HREQ will be disabled.

Figure 1 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More Am9517A/8237As could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level devices forming a third level.

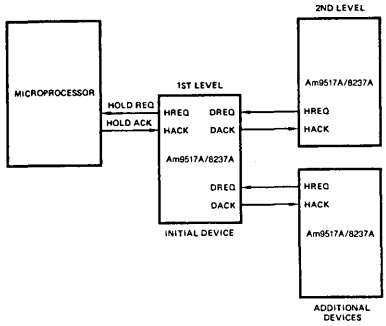


Figure 1. Cascaded Am9517A/8237As

Transfer Types

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from an I/O device to the memory by activating $\overline{IOR}$ and $\overline{MEMW}$. Read transfers move data from memory to an I/O device by activating $\overline{MEMR}$ and $\overline{IOW}$. Verify transfers are pseudo transfers; the Am9517A/8237A operates as in Read or Write transfers generating addresses, responding to $\overline{EOP}$, etc. However, the memory and I/O control lines remain inactive.

Memory-to-Memory: The Am9517A/8237A includes a block move capability that allows blocks of data to be moved from one memory address space to another. When Bit C0 in the Command register is set to a logical 1, channels 0 and 1 will operate as memory-to-memory transfer channels. Channel 0 forms the source address and channel 1 forms the destination address. The channel 1 word count is used. A memory-to-memory transfer is initiated by setting a software DMA request for channel 0. Block Transfer Mode should be used for memory-to-memory. When channel 0 is programmed for a fixed source address, a single source word may be written into a block of memory.

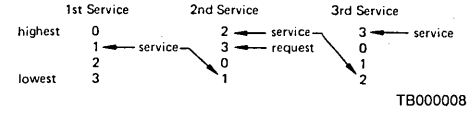
When setting up the Am9517A/8237A for memory-to-memory operation, it is suggested that both channels 0 and 1 be masked out. Further, the channel 0 word count should be initialized to the same value used in channel 1. No DACK outputs will be active during memory-to-memory transfers.

The Am9517A/8237A will respond to external $\overline{EOP}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers may be found in Timing Diagram 2.

Autoinitialize: By programming a bit in the Mode register, a channel may be set up for an Autoinitialize operation. During Autoinitialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word Count registers of that channel following $\overline{EOP}$. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not set by $\overline{EOP}$ when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to repeat its service without CPU intervention.

Priority: The Am9517A/8237A has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly. With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.



The priority encoder selects the highest priority channel requesting service on each active-going HACK edge. Once a channel is started, its operation will not be suspended if a request is received by a higher priority channel. The high priority channel will only gain control after the lower priority channel releases HREQ. When control is passed from one channel to another, the CPU will always gain bus control. This ensures generation of rising HACK edge to be used to initiate selection of the new highest-priority requesting channel.

Compressed Timing: To achieve even greater throughput where system characteristics permit, the Am9517A/8237A can compress the transfer time to two clock cycles. From Timing Diagram 3 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3 the read pulse width is made equal to the write pulse width, and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8 – A15 need updating (see Address Generation). Timing for compressed transfers is found in Timing Diagram 4.

Extended Write: For Flyby Transactions late write is normally used, as this allows sufficient time for the $\overline{IOR}$ signal to get data from the peripheral onto the bus before $\overline{MEMW}$ is activated. In some systems, performance can be improved by starting the write cycle earlier. This is especially true for memory-to-memory transactions.

Address Generation: To reduce pin count, the Am9517A/8237A multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a three-state enable. The lower order address bits are output by the Am9517A/8237A directly. Lines A0 – A7 should be connected to the address bus. Timing Diagram 1 shows the time relationships between CLK, AEN, ADSTB, DB0 – DB7 and A0 – A7.

During Block and Demand Transfer mode services which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the Am9517A/8237A executes S1 states only when updating of A8 – A15 in the latch is necessary. This means for long services that S1 states may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

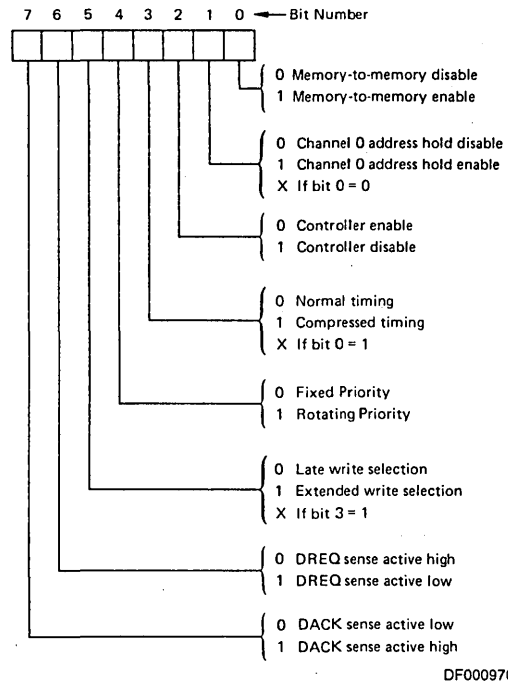
Register Description

Current Address Register: Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialization takes place only after an $\overline{EOP}$.

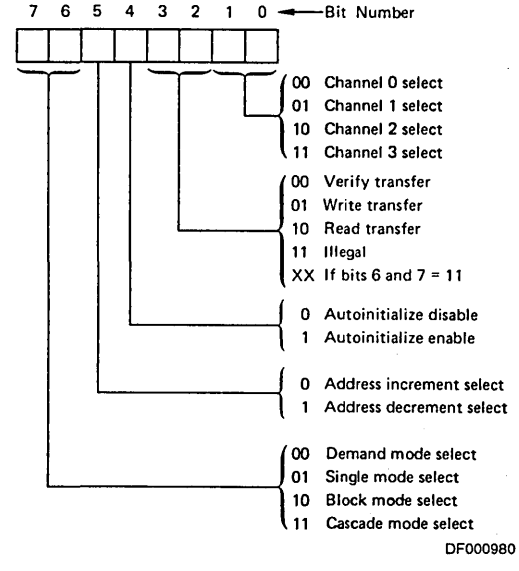
Current Word Count Register: Each channel has a 16-bit Current Word Count register. This register should be programmed with, and will return on a CPU read, a value one less than the number of words to be transferred. The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service, it may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize can occur only when an $\overline{EOP}$ occurs. Note that the contents of the Word Count register will be FFFF (hex) following on internally generated $\overline{EOP}$.

Base Address and Base Word Count Registers: Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original values of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes during DMA, programming by the microprocessor. Accordingly, writing to these registers when intermediate values are in the Current registers will overwrite the intermediate values. The Base registers cannot be read by the microprocessor.

Command Register: This 8-bit register controls the operation of the Am9517A/8237A. It is programmed by the microprocessor in the Program Condition and is cleared by Reset. The following table lists the function of the command bits. See Figure 2 for address coding.

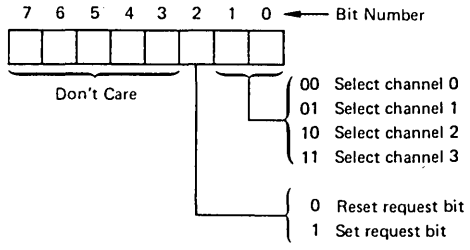


Mode Register: Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written to.



Request Register: The Am9517A/8237A can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are nonmaskable and subject to prioritization by the Priority Encoder network.

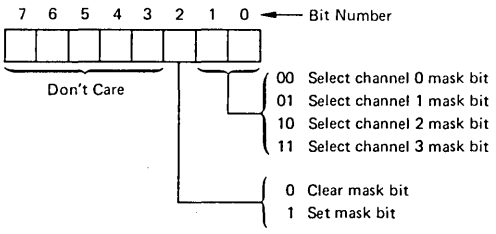
Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external EOP. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 2 for address coding.



DF000990

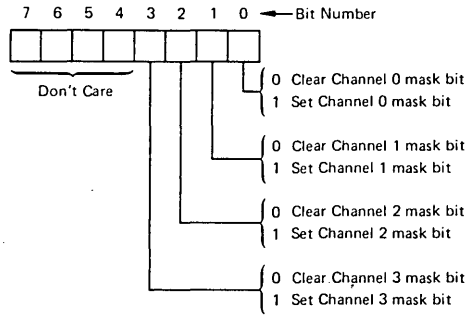
Software requests will be serviced only if the channel is in Block mode. When initiating a memory-to-memory transfer, the software request for channel 0 should be set.

Mask Register: Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an EOP if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 2 for instruction addressing.



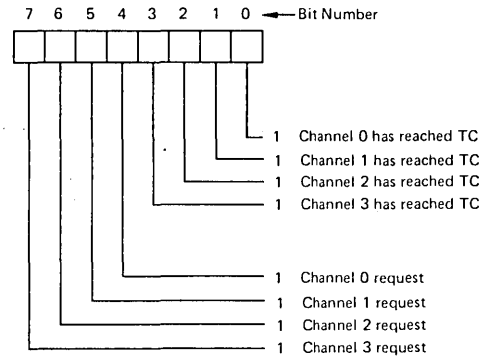
DF001000

All four bits of the Mask Register may also be written with a single command.



DF001010

Status Register: The Status registers may be read out of the Am9517A/8237A by the microprocessor. It indicates which channels have reached a terminal count and which channels have pending DMA requests. Bits 0 – 3 are set each time a TC is reached by that channel, including after each Autoinitialization. These bits are cleared by Reset and each Status Read. Bits 4 – 7 are set whenever their corresponding channel is requesting service.



DF001020

Temporary Register: The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands: There are three special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are:

Clear First/Last Flip/Flop: This command may be issued prior to writing or reading Am9517A/8237A address or word count information. This initializes the Flip/Flop to a known state so that subsequent accesses to register contents by the microprocessor will address lower and upper bytes in the correct sequence. When the Flip/Flop is cleared it addresses the lower byte and when set it addresses the upper byte.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary and Internal First/Last Flip/Flop registers are cleared and the Mask register is set. The Am9517A/8237A will enter the Idle cycle.

Clear Mask Register: This command clears the mask bits of all four channels, enabling them to accept DMA requests.

Interface Signals						
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	Operation
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 2. Register and Function Addressing

Channel	Register	Operation	Signals							Internal Flip/Flop	Data Bus DB0 – DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base & Current Address	Write	0 0	1 1	0 0	0 0	0 0	0 0	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	0 0	0 0	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	0 0	0 0	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	0 0	0 0	1 1	0 1	W0 – W7 W8 – W15
1	Base & Current Address	Write	0 0	1 1	0 0	0 0	0 0	1 1	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	0 0	1 1	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	0 0	1 1	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	0 0	1 1	1 1	0 1	W0 – W7 W8 – W15
2	Base & Current Address	Write	0 0	1 1	0 0	0 0	1 1	0 0	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	1 1	0 0	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	1 1	0 0	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	1 1	0 0	1 1	0 1	W0 – W7 W8 – W15
3	Base & Current Address	Write	0 0	1 1	0 0	0 0	1 1	1 1	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	1 1	1 1	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	1 1	1 1	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	1 1	1 1	1 1	0 1	W0 – W7 W8 – W15

Figure 3. Word Count and Address Register Command Codes

APPLICATIONS INFORMATION

Figure 4 shows a convenient method for configuring a DMA system with the Am9517A/8237A Controller and a microprocessor system. The Multimode DMA Controller issues a Hold Request to the processor whenever there is at least one valid DMA Request from a peripheral device. When the processor replies with a Hold Acknowledge signal, the Am9517A/8237A takes control of the Address Bus, the Data Bus and the Control Bus. The address for the first transfer operation comes out in two bytes – the least significant eight bits on the eight Address outputs and the most significant eight bits on the Data Bus. The contents of the Data Bus are then latched into the Am74LS373 register to complete the full 16 bits of the Address Bus. The Am74LS373 is a high-speed, low power, 8-bit, three-state register in a 20-pin package. After the initial transfer takes place, the register is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one Am9517A/8237A is used.

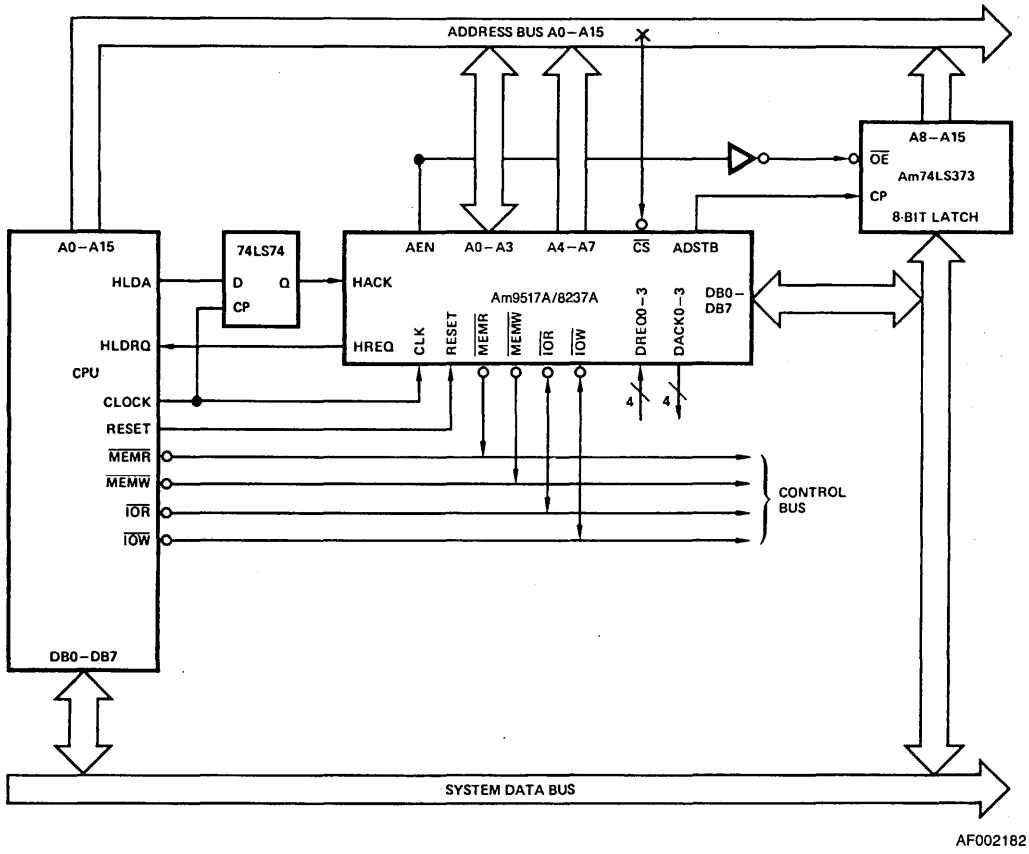


Figure 4. Basic DMA Configuration

ABSOLUTE MAXIMUM RATINGS

Storage temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0 V
All Signal Voltages with Respect to V_{SS} ..... −0.5 V to +7.0 V
Power Dissipation (Package Limitation) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to 70°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

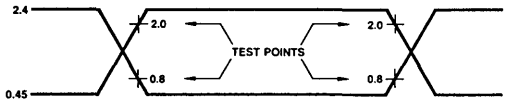
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified. (Note 1)

Parameters	Description	Test Conditions	Min	Typ	Max	Units
VOH	Output HIGH Voltage	IOH = −200 µA	2.4			Volts
		IOH = −100 µA, (HREQ Only)	3.3			
VOL	Output LOW Voltage	IOL = 3.2 mA			0.40 V	Volts
VIH	Input HIGH Voltage		2.0		VCC + 0.5	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC	−10		+10	µA
IOZ	Output Leakage Current	VCC ≤ VO ≤ VSS + 0.40	−10		+10	µA
ICC	VCC Supply Current	T_A = +25°C		65	130	mA
		T_A = 0°C		75	150	
CO	Output Capacitance (Note 14)	fc = 1.0 MHz, Inputs = 0 V		4	8	pF
CI	Input Capacitance			8	15	pF
CIO	I/O Capacitance			10	18	pF
COHREQ	Output Capicitance (HREQ)	fc = 1.0 MHz, Inputs = 0 V		18	20	pF

Notes:
1. Typical values are for T_A = 25°C, nominal supply voltage and nominal processing parameters.
2. Input timing parameters assume transition times of 20 ns or less. Waveform measurement points for both input and output signals are 2.0 V for HIGH and 0.8 V for LOW, unless otherwise noted.
3. Output loading is 1 Standard TTL gate plus 50 pF capacitance unless noted otherwise.
4. The new $\overline{IOW}$ or $\overline{MEMW}$ pulse width for normal write will be TCY-100 ns and for extended write will be 2TCY-100 ns. The net $\overline{IOR}$ or $\overline{MEMR}$ pulse width for normal read will be 2TCY-50 ns and for compressed read will be TCY-50 ns.
5. TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0 V. TDQ2 is measured at 3.3 V. The value for TDQ2 assumes an external 3.3 kΩ pull-up resistor connected from HREQ to VCC.
6. DREQ should be held active until DACK is returned.
7. DREQ and DACK signals may be active high or active low. Timing diagrams assume the active high mode.
8. Output loading on the data bus is 1 Standard TTL gate plus 15 pF for the minimum value and 1 Standard TTL gate plus 100 pF for the maximum value.
9. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 450 ns for the Am9517A-4/8237A-4, and 400 ns for the Am9517A-5/8237A-5 as recovery time between active read or write pulses.
10. Parameters are listed in alphabetical order.
11. Pin 5 is an input that should always be at a logic high level. An internal pull-up resistor will establish a logic high when the pin is left floating. Alternatively, pin 5 may be tied to VCC.
12. Signals $\overline{READ}$ and $\overline{WRITE}$ refer to $\overline{IOR}$ and $\overline{MEMW}$ respectively for peripheral-to-memory DMA operations and to $\overline{MEMR}$ and $\overline{IOW}$ respectively for memory-to-peripheral DMA operations.
13. If N wait states are added during the write-to-memory half of a memory-to-memory transfer, this parameter will increase by N (TCY).
14. All output pins except HREQ.
15. Because $\overline{EOP}$ high from clock high is load dependent, users wishing to test these parameters should use a 2k pull-up resistor and a tester with 50 pF or less load capacitance. Time constant RC = 120 ns is added to the specified number in the data sheet for testing.

SWITCHING TEST INPUT WAVEFORM



WF003310

ABSOLUTE MAXIMUM RATINGS

Storage temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0 V
All Signal Voltages with Respect to V_{SS} .. −0.5 to +7.0 V
Power Dissipation (Package Limitation) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Military (M) Devices
Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... 5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over **MILITARY** operating range (for SMD/DESC and APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{OH}	Output HIGH Voltage	I_{OH} = −200 μA, V_{CC} = 4.5 V	2.4		V
		I_{OH} = −100 μA, (HREQ Only)	3.3		
V_{OL}	Output LOW Voltage	I_{OL} = 3.2 mA, V_{CC} = 5.5 V		0.45 V	V
V_{IH}	Input HIGH Voltage	V_{CC} = 4.5 V, 5.5 V	2.2	V_{CC} + 0.5*	V
V_{IHCLK}	Input HIGH Voltage	V_{CC} = 4.5 V, 5.5 V (CLK Only)	2.35	V_{CC} + 0.5*	V
V_{IL}	Input LOW Voltage	V_{CC} = 4.5 V, 5.5 V	−0.5*	0.7	V
I_{IX}	Input Load Current	$V_{SS} \leq V_I \leq V_{CC}$, V_{CC} = 5.5 V	−10	+10	μA
I_{OZ}	Output Leakage Current	$V_{CC} \leq V_O \leq V_{SS}$ + 0.40, V_{CC} = 5.5 V	−10	+10	μA
I_{CC}	V_{CC} Supply Current	(Note 1)		150	mA
C_O †	Output Capacitance (Note 12)	fc = 1.0 MHz, Inputs = 0 V		8*	pF
C_I †	Input Capacitance			15*	pF
C_{IO} †	I/O Capacitance		18*	pF	

* Guaranteed by design.
† Not included in Group A tests.

Notes:
1. I_{CC} is measured in a dynamic condition with outputs in a worst-case state having no loads applied.
2. Input timing parameters assume transition times of 20 ns or less. Waveform measurement points for both input and output signals are 2.0 V for HIGH and 0.8 V for LOW, unless otherwise noted.
3. The new $\overline{IOW}$ or $\overline{MEMW}$ pulse width for normal write will be TCY-100 ns and for extended write will be 2TCY-100 ns. The net $\overline{IOR}$ or $\overline{MEMR}$ pulse width for normal read will be 2TCY-50 ns and for compressed read will be TCY-50 ns.
4. TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0 V. TDQ2 is measured at 3.3 V. The value for TDQ2 assumes an external 3.3 kΩ pull-up resistor connected from HREQ to V_{CC}.
5. DREQ should be held active until DACK is returned.
6. DREQ and DACK signals may be active HIGH or active LOW. Timing diagrams assume the active-HIGH mode.
7. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600 ns for the Am9517A, at least 450 ns for the Am9517A-4 as recovery time between active read or write pulses.
8. Parameters are listed in alphabetical order.
9. Pin 5 is an input that should always be at a logic-HIGH level. An internal pull-up resistor will establish a logic HIGH when the pin is left floating. Alternatively, pin 5 may be tied to V_{CC}.
10. Signals $\overline{READ}$ and $\overline{WRITE}$ refer to $\overline{IOR}$ and $\overline{MEMW}$ respectively for peripheral-to-memory DMA operations and to $\overline{MEMR}$ and $\overline{IOW}$ respectively for memory-to-peripheral DMA operations.
11. If N wait states are added during the write-to-memory half of a memory-to-memory transfer, this parameter will increase by N (TCY).
12. All output pins except HREQ.
13. Because $\overline{EOP}$ HIGH from clock HIGH is load-dependent, users wishing to test these parameters should use a 2k pull-up resistor and a tester with 50 pF or less load capacitance. Time constant R_C = 120 ns is added to the specified number in the data sheet for testing.

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating ranges unless otherwise specified
ACTIVE CYCLE (Notes 2, 3, 10, 11, and 12)

Parameters	Description	Am9517A-4/8237A-4		Am9517A-5/8237A-5		Units
		Min	Max	Min	Max	
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		225		200	ns
TAET	AEN LOW from CLK HIGH (S1) Delay Time		150		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		120		90	ns
TAFC	$\overline{READ}$ or $\overline{WRITE}$ Float from CLK HIGH		120		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		190		170	ns
TAHR	ADR from $\overline{READ}$ HIGH Hold Time	TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	30		30		ns
TAHW	ADR from WRITE HIGH Hold Time	TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time		220		170	ns
	$\overline{EOP}$ HIGH from CLK HIGH Delay Time		190		170	ns
	$\overline{EOP}$ LOW to CLK HIGH Delay Time		190		170	ns
TASM	ADR Stable from CLK HIGH		190		170	ns
TASS	DB to ADSTB LOW Set-up Time	100		100		ns
TCH	Clock High Time (Transitions $\leqslant$ 10ns)	100		80		ns
TCL	Clock Low Time (Transitions $\leqslant$ 10ns)	110		68		ns
TCY	CLK Cycle Time	250		200		ns
TDCL	CLK HIGH to $\overline{READ}$ or $\overline{WRITE}$ LOW Delay (Note 4)		200		190	ns
TDCTR	$\overline{Read}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		210		190	ns
TDCTW	$\overline{WRITE}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		150		130	ns
TDQ1	HREQ Valid from CLK HIGH Delay Time (Note 5)		120		120	ns
TDQ2			190		120	ns
TEPS	$\overline{EOP}$ LOW from CLK LOW Set-up Time	45		40		ns
TEPW	$\overline{EOP}$ Pulse Width	225		220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		190		170	ns
TFAC	$\overline{READ}$ or $\overline{WRITE}$ Active from CLK HIGH		150		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		225		200	ns
THS	HACK Valid to CLK HIGH Set-up Time	75		75		ns
TIDH	Input Data from $\overline{MEMR}$ HIGH Hold Time	0		0		ns
TIDS	Input Data to $\overline{MEMR}$ HIGH Set-up Time	190		170		ns
TODH	Output Data from $\overline{MEMW}$ HIGH Hold Time	20		10		ns
TODV	Output Data Valid to $\overline{MEMW}$ HIGH (Note 13)	125		125		ns
TQS	DREQ to CLK LOW (S1, S4) Set-up Time	0		0		ns
TRH	CLK to READY LOW Hold Time	20		20		ns
TRS	READY to CLK LOW Set-up Time	60		60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		150		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		110		90	ns
TQH	DREQ from DACK Valid Hold Time	0		0		ns
TRQHA	HREQ to HACK Delay Time	1		1		c/k
PROGRAM CONDITION (Idle Cycle) (Notes 2, 3, 10, and 11)						
TAR	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		50		ns
TAW	ADR Valid to $\overline{WRITE}$ HIGH Set-up Time	150		130		ns
TCW	CS LOW to $\overline{WRITE}$ HIGH Set-up Time	150		130		ns
TDW	Data Valid to $\overline{WRITE}$ HIGH Set-up Time	150		130		ns
TRA	ADR or CS Hold from $\overline{READ}$ HIGH	0		0		ns
TRDE	Data Access from $\overline{READ}$ LOW (Note 8)		200		140	ns
TRDF	DB Float Delay from $\overline{READ}$ HIGH	20	100	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Set-up Time	500		500		µs
TRSTS	RESET to First $\overline{IOWR}$	2TCY		2TCY		ns
TRSTW	RESET Pulse Width	300		300		ns
TRW	$\overline{READ}$ Width	250		200		ns
TWA	ADR from $\overline{WRITE}$ HIGH Hold Time	20		20		ns
TWC	CS HIGH from $\overline{WRITE}$ HIGH Hold Time	20		20		ns
TWD	Data from $\overline{WRITE}$ HIGH Hold Time	30		30		ns
TWWS	Write Width	200		160		ns

Notes: See notes under DC Characteristics table.

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for SMD/DESC and APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

ACTIVE CYCLE (Notes 2, 8, 9, and 10)

		Am9517A		Am9517A-4		
Parameters	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		300		225	ns
TAET	AEN LOW from CLK HIGH (S1) Delay Time		200		150	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		150		120	ns
TAFC	$\overline{READ}$ or $\overline{WRITE}$ Float from CLK HIGH		150		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		250		190	ns
TAHR	ADR from $\overline{READ}$ HIGH Hold Time	TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	30		30		ns
TAHW	ADR from WRITE HIGH Hold Time	TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time		280		220	ns
	$\overline{EOP}$ HIGH from CLK HIGH Delay Time		250		190	ns
	$\overline{EOP}$ LOW to CLK HIGH Delay Time		250		190	ns
TASM	ADR Stable from CLK HIGH		250		190	ns
TASS	DB to ADSTB LOW Setup Time	100		100		ns
TCH	Clock High Time (Transitions $\leqslant$ 10ns)	120		100		ns
TCL	Clock Low Time (Transitions $\leqslant$ 10ns)	150		110		ns
TCY	CLK Cycle Time	320		250		ns
TDCL	CLK HIGH to $\overline{READ}$ or $\overline{WRITE}$ LOW Delay (Note 3)		270		200	ns
TDCTR	$\overline{Read}$ HIGH from CLK HIGH (S4) Delay Time (Note 3)		270		210	ns
TDCTW	$\overline{WRITE}$ HIGH from CLK HIGH (S4) Delay Time (Note 3)		200		150	ns
TDQ1	HREQ Valid from CLK HIGH Delay Time (Note 4)		160		120	ns
TDQ2			2TCY + 250		2TCY + 190	ns
TEPS	$\overline{EOP}$ LOW from CLK LOW Setup Time	60		45		ns
TEPW	$\overline{EOP}$ Pulse Width	300		225		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		250		190	ns
TFAC	$\overline{READ}$ or $\overline{WRITE}$ Active from CLK HIGH		200		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		300		225	ns
THS	HACK Valid to CLK HIGH Setup Time	100		75		ns
TIDH	Input Data from $\overline{MEMR}$ HIGH Hold Time	0		0		ns
TIDS	Input Data to $\overline{MEMR}$ HIGH Setup Time	250		190		ns
TODH	Output Data from $\overline{MEMW}$ HIGH Hold Time	20		20		ns
TODV	Output Data Valid to $\overline{MEMW}$ HIGH (Note 11)	200		125		ns
TQS	DREQ to CLK LOW (S1, S4) Setup Time	0		0		ns
TRH	CLK to READY LOW Hold Time	20		20		ns
TRS	READY to CLK LOW Setup Time	100		60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		200		150	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		140		110	ns
TQH	DREQ from DACK Valid Hold Time	0		0		ns
TRQHA	HREQ to HACK Delay Time	1		1		clk

AC Device Test Conditions: V_{CC} = 4.5 V, 5.5 V
V_{IL} = 0.45 V, V_{IH} = 2.4 V
V_{OL} = 0.8 V, V_{OH} = 2.0 V
I_{OL} = 3.2 mA, I_{OH} = 200 μA
CL = 100 pF ±20 pF

Notes: See notes following DC Characteristics.

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)

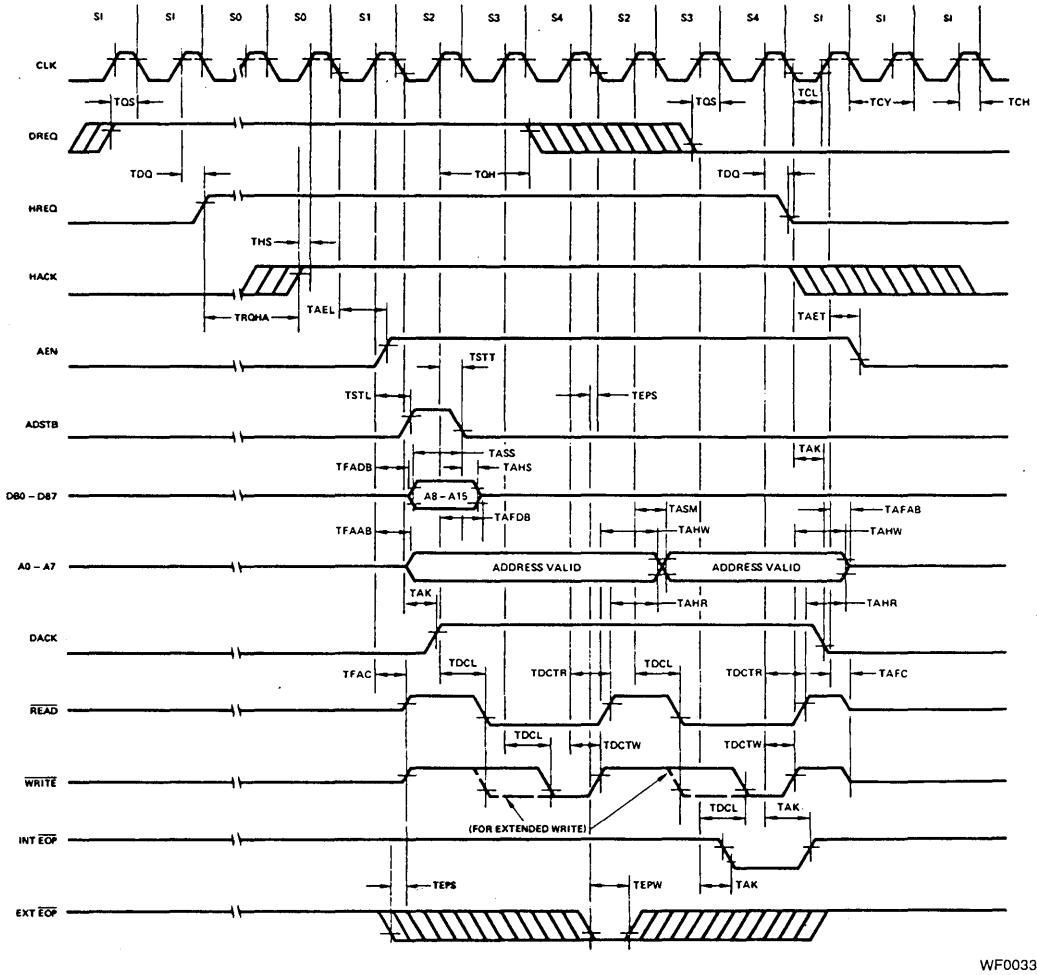
PROGRAM CONDITION (Idle Cycle) (Notes 2, 3, 10, and 11)

Parameters	Description	Am9517A		Am9517A-4		Unit
		Min.	Max.	Min.	Max.	
TAR	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		50		ns
TAW	ADR Valid to $\overline{WRITE}$ HIGH Setup Time	200		150		ns
TCW	CS LOW to $\overline{WRITE}$ HIGH Setup Time	200		150		ns
TDW	Data Valid to $\overline{WRITE}$ HIGH Setup Time	200		150		ns
TRA	ADR or CS Hold from $\overline{READ}$ HIGH	0		0		ns
TRDE	Data Access from $\overline{READ}$ LOW (Note 8)		300		200	ns
TRDF	DB Float Delay from $\overline{READ}$ HIGH	20	150	20	100	ns
TRSTD	Power Supply HIGH to RESET LOW Setup Time	500		500		μs
TRSTS	RESET to First $\overline{IOWR}$	2TCY		2TCY		ns
TRSTW	RESET Pulse Width	300		300		ns
TRW	$\overline{READ}$ Width	300		250		ns
TWA	ADR from $\overline{WRITE}$ HIGH Hold Time	20		20		ns
TWC	CS HIGH from $\overline{WRITE}$ HIGH Hold Time	20		20		ns
TWD	Data from $\overline{WRITE}$ HIGH Hold Time	30		30		ns
TWWS	Write Width	200		200		ns
TAD	Data Access from ADR Valid $\overline{CS}$ LOW (TAD = TAR + TRDE)		300		300	ns

Notes: See notes following DC Characteristics.

SWITCHING WAVEFORMS

Timing Diagram 1. Active Cycle Timing Diagram

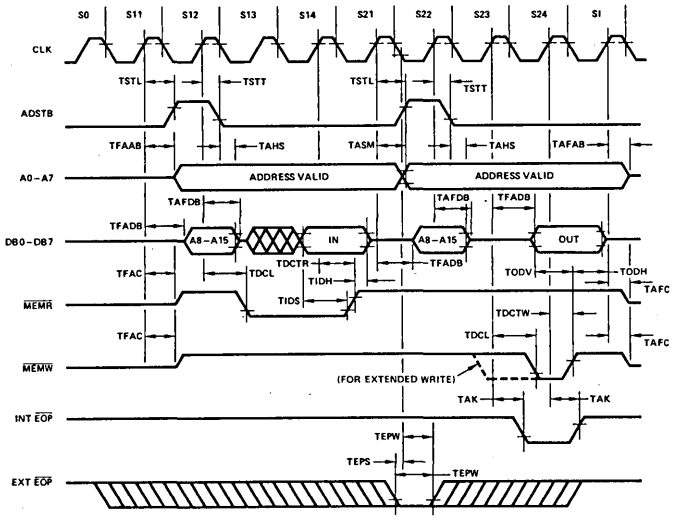


WF003300

Note: $\overline{EOP}$ must precede AEN in single transfer mode.

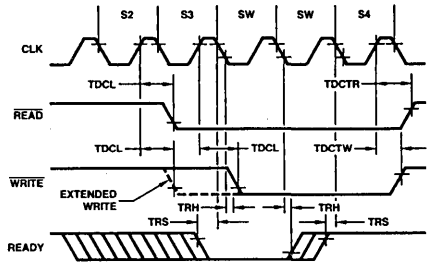
SWITCHING WAVEFORMS (continued)

Timing Diagram 2. Memory-to-Memory



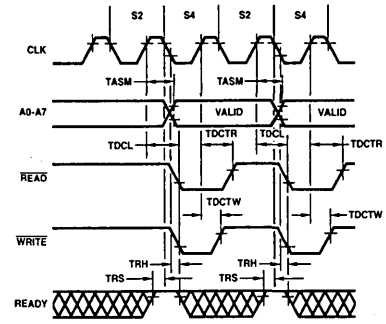
WF003320

Timing Diagram 3. Ready Timing



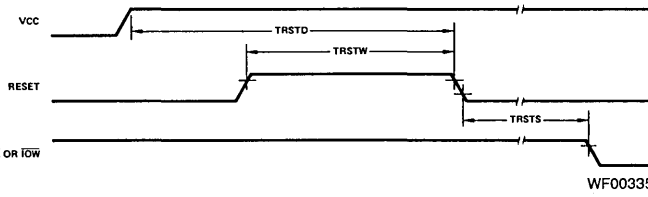
WF003330

Timing Diagram 4. Compressed Timing



WF003340

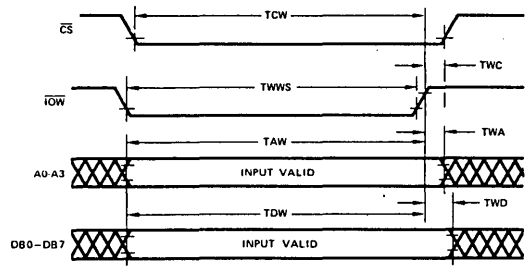
Timing Diagram 5. Reset Timing



WF003350

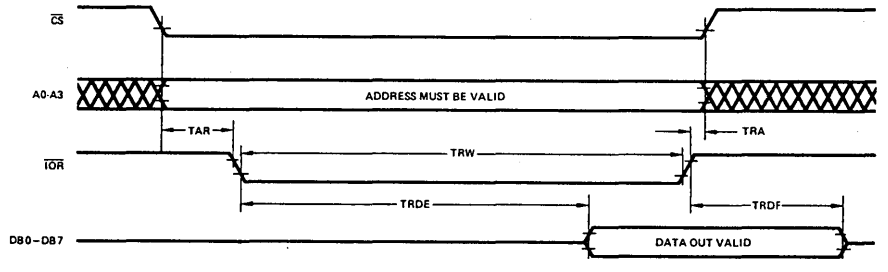
SWITCHING WAVEFORMS (continued)

Timing Diagram 6. Program Condition Write Timing



WF003360

Timing Diagram 7. Program Condition Read Cycle



WF003370

80286 High Performance Memory Interface (True Zero Wait State versus Page-Interleaved)

by Linda Bishop

PREFACE

This application note will detail the approach to high performance, yet cost-effective, page interleaved memory to support a 16-MHz 80286-based system.

Cost performance comparisons are provided for traditional zero-wait-state memory systems for both 12-MHz and 16-MHz processor speeds, and for a page-interleaving memory system.

The end result is a memory design that allows an overall faster and lower cost system to be realized.

The discussion of high performance 80286-based PC-AT-compatible systems inevitably focuses on the number of clock cycles required to perform the I/O and main memory accesses in the system. In other words, the time-critical "wait states." Wait states are necessary to match the performance of subsystems to the performance of the microprocessor.

System performance indicates the speed that the microprocessor system performs a given task or set of instructions. Memory or peripheral performance indicates the speed that the microprocessor can access or store (read or write) an item of information into the memory or peripheral. To accommodate slower memories or peripherals, which cannot keep up with the 80286 microprocessor, wait states can be inserted into the 80286 bus cycle. Of course a slower clock frequency for the 80286 can be used as well, thereby lengthening the 80286 bus cycle.

When reading from a memory, for example, the microprocessor allows a fixed amount of time for the memory to respond with valid data after an address is output. This is the access time required by the microprocessor for full-speed operation and is determined by the microprocessor clock rate. If the memory cannot respond in time, a wait state must be inserted or an error will result. The access time of the memory must be less than the access time required by the microprocessor, including any wait states.

These wait states extend the time required to perform individual bus operations and therefore directly influence 80286 system performance. For a 16-MHz 80286, each wait state adds an additional 62 ns to the minimum bus cycle time of 125 ns. The key to high performance then is to eliminate the wait states.

Several operations are time-critical in the PC-AT system where wait states need to be examined: system board DRAM access, system board ROM access, expansion memory access, and bus I/O operations.

System ROM contains various system service routines. ROM memory access occurs more frequently during power up and less frequently during normal operations. Bus I/O operations are also relatively infrequent.

The most common operation performed by the 80286 is a memory access to DRAM on the system mother board. DRAM memory is used for program storage and for storage of data and information used in processing, hence the most important for no-wait-state performance. This article will focus on DRAM access.

Two of the principle considerations in designing DRAM memory subsystems for the 80286 are the effect of memory performance on overall system performance and the trade-off between system performance and memory cost.

Coupling a high-performance 80286 with low-performance memory makes little sense; a 16-MHz system with one wait state is only 5–6% faster than a 12-MHz 80286 system with zero wait states, and is more costly. To realize the high performance potential of the 80286, it must be coupled to relatively fast memory.

However, fast memory devices are more expensive than slower memory devices. In an 80286 system supporting up to 16 megabytes of physical addressable memory, fast memories result in a very costly design.

Designers must strike a balance between system performance and system cost, and the choice of memory subsystems is the balancing factor.

MEMORY INTERFACE TECHNIQUES

DRAM Memories

Memory subsystems performance is determined by both subsystem design and by the speed of the memory devices. Two memory interface techniques will be explored here: standard address strobe memory and page-interleaved memory. A performance versus cost trade-off of these will be examined in detail as well.

Before beginning this, however, it is helpful to first take a brief and general look at DRAM memories. In a typical microprocessor system, memory is 16 bits wide and from 512K to several million bits deep. Memory of this size can best be implemented using "By one" architecture to minimize the number of lines, input and output capacitances, pin count, board area, and cost.

A 256K DRAM has nine address lines requiring multiplexing to produce the 18 address bits needed to access all 256K cell locations. To address any single location requires multiplexing of two 9-bit address fields

(A1–A19). At the beginning of every memory cycle, two clocks are used to strobe these address bits in. RAS is initiated on the lower 9 address bits to address a row. CAS is asserted on the upper 9 address bits to specify a column. The intersection of the row and column determines the bit to be accessed in either a read or write operation.

To implement 1 megabyte of physical memory, A0 on the 80286 is used to differentiate between two sets of banks, each bank containing 16 256K × 1 DRAMs. There are 32 256K × 1 parts required to provide 1 megabyte of memory storage (see Figure 1).

A20–A23 on the 80286 provide the signals for up to 16 megabytes of memory as illustrated in Figure 2. If 256K × 1 parts are used, 512 will be required for the full

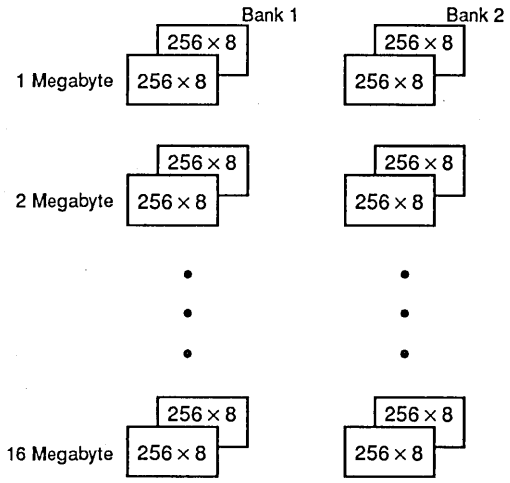


Figure 1

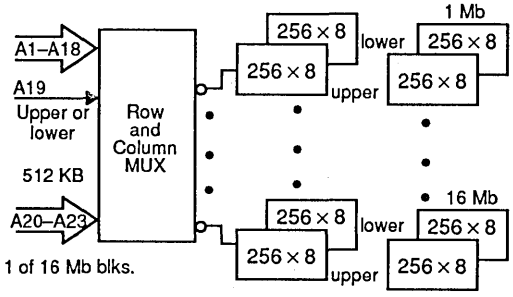


Figure 2

16 megabytes of memory. With memory this large, however, 1 megabit DRAMs are more economical.

Because DRAM memories are dynamic they require refreshing. A DRAM memory cell is made of a single transistor and storage capacitor. The RAM is based on information stored as charge on the capacitor accessed through the gated transistor. This charge degrades with time and must be refreshed to retain the data. This is accomplished by a refresh operation that cycles through the 256 row address locations every 4 ms or a refresh rate of 15.6 µs. Most DRAMs have several refresh mechanisms available to system designers.

System Timing Analysis

The first high-performance memory interface technique we will look at is a standard interface using a companion DRAM controller. Most controllers can drive up to a 1-megabyte array and can directly drive up to 88 DRAMs. The DRAM controller acts as the address controller between any processor and dynamic memory. It handles address multiplexing, refresh cycles, counting, and address strobe timing. In other words, it handles all required data and refresh needed by the DRAM memory. To implement DRAM control and refresh circuitry using standard TTL devices typically requires 8–12 chips. A DRAM controller performs these functions in a single IC.

A typical zero-wait-state application is shown in Figure 3.

To achieve zero-wait-state operation, the controller interfaces synchronously to the processor. The DRAM controller's clock input must be connected to the 80286 clock input (clock signal is provided by the 82284 clock generator). The controller's address inputs connect directly to the 80286's address outputs. Since the memory requires that an address be present for an entire cycle, the address must be stored in a latch external to the microprocessor. A1–A19 addresses are latched internally in the DRAM controller. An external latch is used for the address line (A0) which is not used to address the RAM directly, but is used along with the BHE signal to gate the upper and lower memory byte. A0 = 0 accesses the lower memory bank, A0 = 1 accesses the upper memory bank.

For a read operation A0 and BHE are ignored since the microprocessor will look only at the half of the data bus that it is interested in. For a write cycle, however, the WRITE to the RAM must be gated by these signals.

To perform a byte transfer (8 bits) to an even address, the CPU transfers information over the lower half of the data bus (D0–D7). A0 enables the lower memory bank to participate in the transfer while BHE disables the bank on the upper half of the data bus. To perform a word transfer (16 bits), A0 and BHE are both held low, enabling both memory banks. A1–A19 select the appropriate byte from each bank. These byte and word transfers are illustrated in Figure 4.

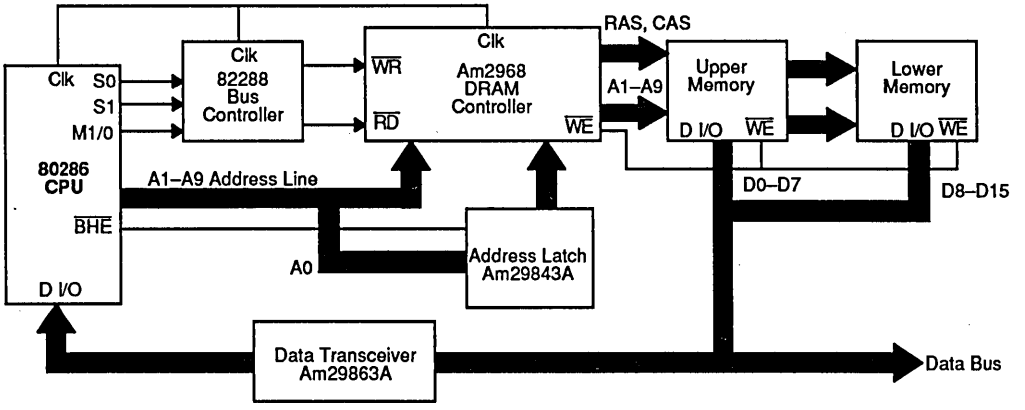


Figure 3

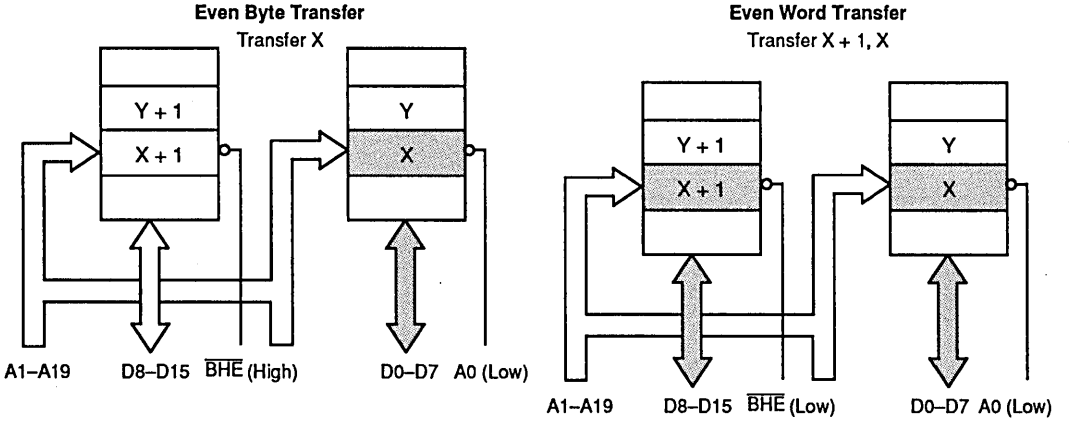


Figure 4

Figure 5 illustrates the zero-wait-state timing for a back-to-back memory cycle: a read from Memory Bank 1 and a write to Memory Bank 2. Since the cycles are to different memory banks, the RAS 1 for the second cycle is overlapped with the RAS 0 precharge from the first cycle to allow a no-wait-state response. If the second cycle were to the same memory bank, a wait state would need to be inserted to allow for RAS precharge period.

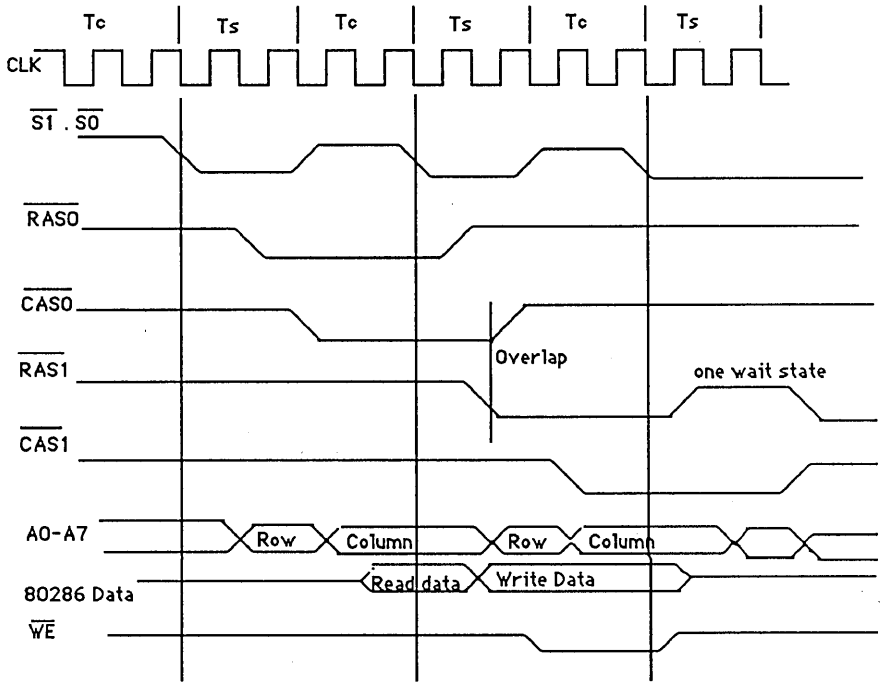
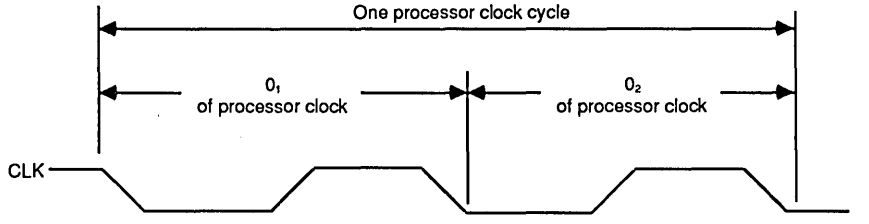


Figure 5

In order to more fully understand this timing diagram and the discussion to follow, here is a brief explanation of the 80286 clock. The 80286 uses a double frequency system clock (CLK) to control bus timing. The CPU internally divides the system CLK by two to provide the internal processor clock. In other words, each processor clock is composed of two system clocks called Phase 1 and Phase 2.



The processor clock can have several bus states; Send Status (Ts), Perform Command (Tc), and Idle (Ti).

Ts is signaled by S1 and S0 indicating a read or write operation (S1 High/S0 Low is a read). The Tc cycle is when the memory or I/O device is expected to respond by either transferring read data to the CPU or accepting write data. Tc cycles are repeated as many times as necessary to assure sufficient time for the memory or I/O device to respond. Each additional Tc cycle is one wait state.

Let's return now to our discussion of timing and discuss the DRAM access time necessary to operate a zero-wait-state system.

To operate the 80286-16 MHz at zero wait state requires DRAM access times of 60 ns. This is determined by measuring the memory access cycle of RAS going low at the controller to data valid at the CPU pins. In a standard memory read cycle, the time available to generate the memory access cycle plus allow for the RAS to data access time of the DRAM, plus return the data to the processor is equal to the number of clock cycles. In other words:

(A)	Memory Access time
+	+
(B)	RAS active delay from controller to allow RAM RAS to data access
+	+
(C)	Transceiver delay to return data to the CPU
+	+
(D)	CPU data setup time
=	=
(E)	Number of clk cycles · Clock cycle time
or	{A = E – B – C – D}

The calculation of maximum allowable RAS access time for a 16-MHz system is shown below:

RAS active to Valid Data for a Read cycle

Timing	Max RAS access time for 16-MHz system	
(E)	Three 80286 clock cycles (max)	93 ns
–(B)	–RAS active delay (max) (Am2968 controller/ 50 pf load)	–18 ns
–(C)	–Transceiver delay (max) (Am29863A)	–8 ns
–(D)	–80286 read data setup time (min)	–5 ns
= (A) =	Maximum address access time	62 ns

The corresponding CAS access time is:

CAS active to Valid Data for a Read cycle

Timing	Max CAS access time for 16-MHz system
Two 80286 clock cycles (max)	62 ns
–CAS active delay (max) (Am2968 controller)	–17 ns
–Transceiver delay (max) (Am29863A)	–8 ns
–80286 read data setup time (min)	–5 ns
= Maximum address access time	32 ns

The table below shows the access times for a 60-ns DRAM and a memory controller with zero wait state.

Timing Parameter	60-ns DRAM	16-MHz 80286 with memory controller zero wait state
RAS access Time	60 ns	62 ns
CAS access Time	30 ns	32 ns

Thus, 60-ns DRAMs are required for 16-MHz zero wait state making the system very expensive. Using the same analysis, a 12-MHz zero-wait-state system requires 80-ns DRAMs.

We now turn our attention to the second memory interface technique using a Page-Interleaved memory controller. Several companies manufacture page-interleaved controllers including Chips & Technology, VTI, and Faraday. The controller provides an interleaved memory subsystem design with page mode operation, supporting up to 8 MB of on-board DRAM. With additional circuitry, up to 16 MB of memory can be supported. (For this, the CPU address line A23 is clocked in a D flip-flop, whose outputs are ORed with the memory controller CAS line to generate two additional CAS signals for another two blocks of memory.)

With an interleaved memory subsystem, memory devices are grouped into banks so that every sequential memory fetch comes from a different memory bank allowing full use of the 80286's address pipelining. This allows an early address from the 80286 into the next memory bank while the address from the current memory cycle is still valid in another bank. The interleaving produces increased access times and allows considerably slower DRAMs to be used while still providing performance equal to that of faster DRAMs using a standard memory interface. A system using interleave logic is shown in Figure 6.

Typical two-way interleaving schemes use two banks of DRAMs with even word addresses on one bank and odd word addresses on the other. For sequential memory accesses, the RAS precharge time of one bank overlaps the RAS active time of the other bank. Figure 7A shows a memory interleaving scheme using Banks 0 and 1. The RAS signals are interleaved so that the RAS precharge time (Tp) of one bank is used as the RAS active time of the other bank. This requires sequential accesses to be alternating between the banks. With a solely interleaved system, back-to-back cycles to the same memory bank (as with the controller application) will require a wait state. This equates to a small decrease in system performance over a zero-wait-state system.

Paged mode on the other hand does allow back-to-back accesses to occur with zero wait states. Page mode allows the DRAM to perform successive operations very rapidly. A Row Address Strobe (RAS) is not needed to reach individual bits of data; instead the RAS is locked in first, then successive strobes of the CAS clock enter sequential column addresses. In the page mode the RAS clock is held active for a single row address, and the CAS clock is cycled while new column addresses are supplied for each bit. As long as the microprocessor accesses sequential bits in the row, no wait states are needed. This way of operation eliminates the time normally needed to strobe a new row address. Figure 7B shows paged mode DRAM operation.

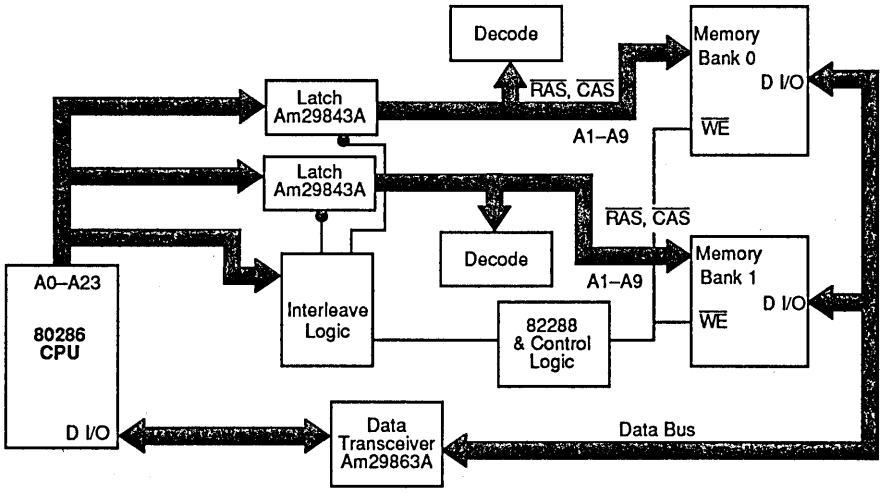


Figure 6

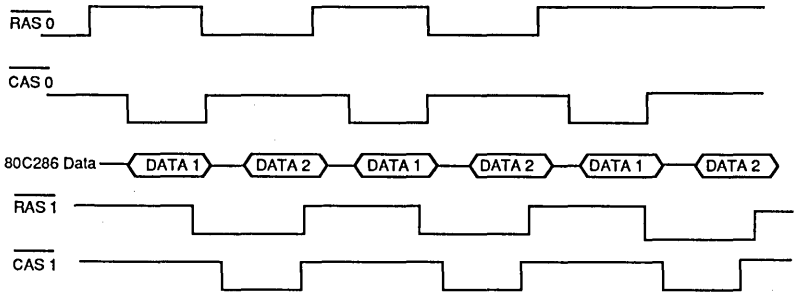


Figure 7A DRAM Interleaved Operation

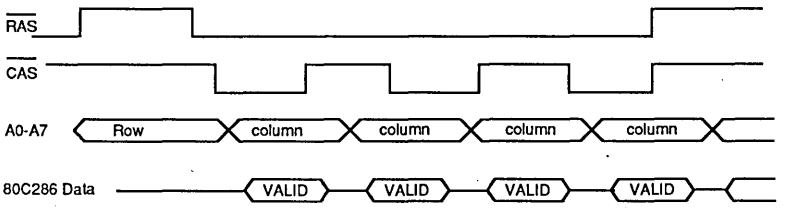


Figure 7B DRAM Page Mode Operation

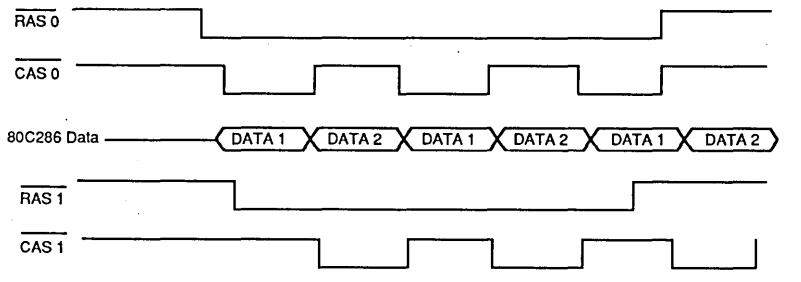


Figure 7C DRAM Page/Interleaved Operation

Figure 7

Combining these two features of interleaving and page mode memory results in a substantial increase in maximum DRAM access time allowed. Figure 7C shows a two-way page-interleaved scheme. It is possible to make zero-wait-state accesses between the two banks by overlapping CAS precharge time of one bank with CAS active time of the other bank. The RAS lines can be held active until the RAS active time-out when a RAS precharge is required.

The DRAM access time is now determined by the CAS active delay, which is typically equal to or less than half of the RAS access time. The maximum allowable CAS access time for a 16-MHz page-interleaved system is 51 ns, as calculated below warranting a 100-ns DRAM. (A 12-MHz zero-wait-state system must use 100-ns DRAMs.)

Timing	Max CAS access time for 16-MHz system
Three 80286 clock cycles (max)	93 ns
–Memory controller CAS delay (max)	–19 ns
–Memory bus to data bus delay (max)	–18 ns
–80286 read data setup time (min)	–5 ns
= Maximum address access time	51 ns

Thus, a 16-MHz 80286 system can utilize 100 ns DRAMs to obtain zero-wait-state operation, which will provide a significant (60%) performance improvement, at no memory cost increase over a 12-MHz zero-wait-state system using a conventional memory organization technique.

PERFORMANCE ANALYSIS

We have concluded, thus far, that in a PC design with 1 megabyte or more of RAM, a 16-MHz page interleaved system is faster than a 12-MHz zero-wait-state system. We have also concluded that a 16-MHz page-interleaved system requires a significantly less expensive memory subsystem than either a 12-MHz zero wait state or 16-MHz zero-wait-state system.

Let's take a look now at the performance versus cost differences between these two memory interface techniques. Several benchmarks have been run to demonstrate the performance differences between a true zero-wait-state system and one incorporating page-interleaved memory.

The benchmarks that were run are all industry standard benchmarks run by many manufacturers and magazines. Dhrystone 1.1 runs under DOS 3.31. Norton SI™ (System Info) is a product of Peter Norton Computing Inc., reporting performance relative to an IBM PC-XT™. MIPS
(Millions of Instructions Per Second) was written by PC Labs of PC Magazine reporting several different CPU tests. Landmark Speed from Landmark Software Corp. reports two numbers. The first is a rating similar to the Norton SI relative to the PC-XT. The second is a speed (MHz) rating that rates performance relative to an original 6-MHz PC-AT. (See Chart of Benchmark Analysis.)

BENCHMARK ANALYSIS CHART

	286-12 0WS	286-16 0WS	286-16 paged	12 vs 16 0WS	0WS vs Paged 286-16	
PC Labs 4.1						
128K NOP Loop	1.52	2	2	24%	0%	
Do nothing	1.77	2.4	2.4	26%	0%	
Integer add	1.70	2.3	1.7	26%	26%	
Integer multiply	1.58	2.2	1.7	28%	23%	
String sort and move	1.58	2.5	2	36%	20%	
Prime number sieve	1.86	2.5	2	26%	20%	
88/86 instruction set	1.88	2.5	2.3	25%	8%	
286 instruction set	1.87	2.5	2.3	24%	8%	
Norton SI		18	18		0%	
Landmark Speed 2	17.93	20.8	20.1	14%	3%	
Dhrystone 1.1		3333		2941		12%
Total (Average)				25%	9%	

The results indicate an average performance degradation of 9% when using the page-interleaved interface versus the true zero wait state. This is not significant since the performance degradation of one wait state versus zero wait state is about 19%.

The advantage to using paged interleaving, as we have shown, is the ability to use slower, less expensive memory parts. The page-interleaved system uses 100-ns DRAMs at 16 MHz, while the zero-wait-state system requires 60-ns DRAMs at 16 MHz.

PRICE ANALYSIS

Let's look at two scenarios, one using 1 megabyte of DRAM (for DOS) and the other using 4 megabytes (for OS/2™). Four megabytes is necessary to run multiple applications in OS/2.

DRAM pricing is based on typical prices available from distribution for "volume purchases."

100 ns 256K DRAM	$ 6.00
60 ns 256K DRAM	$16.00
Page mode memory controller	$20.00

A 1-megabyte DOS system would use 36 256K DRAMs (32 plus 4 parity). For a 16-MHz zero-wait-state system using 60-ns DRAMs, this amounts to $576 in memory parts. A page-interleaved system would use $216 in 100-ns DRAMs plus a $20 memory controller for a total of $236. Thus, a zero-wait-state system gives you a 9% performance increase for a 244% price increase in memory costs, or a $38 per percent performance increase. In addition, the memory cost for a 16-MHz page-interleaved system at $236 is very close to that of a 12-MHz zero-wait-state system at $216 while offering a substantial performance increase.

A 4 megabyte zero-wait-state system would use $2300 in memory while a page-interleaved system would use $884 in memory and controller. Thus, the zero wait state system costs $157 for every 1% performance increase.

The price increase in a zero-wait-state system is not warranted by the low 9% performance increase as illustrated graphically in Figure 8.

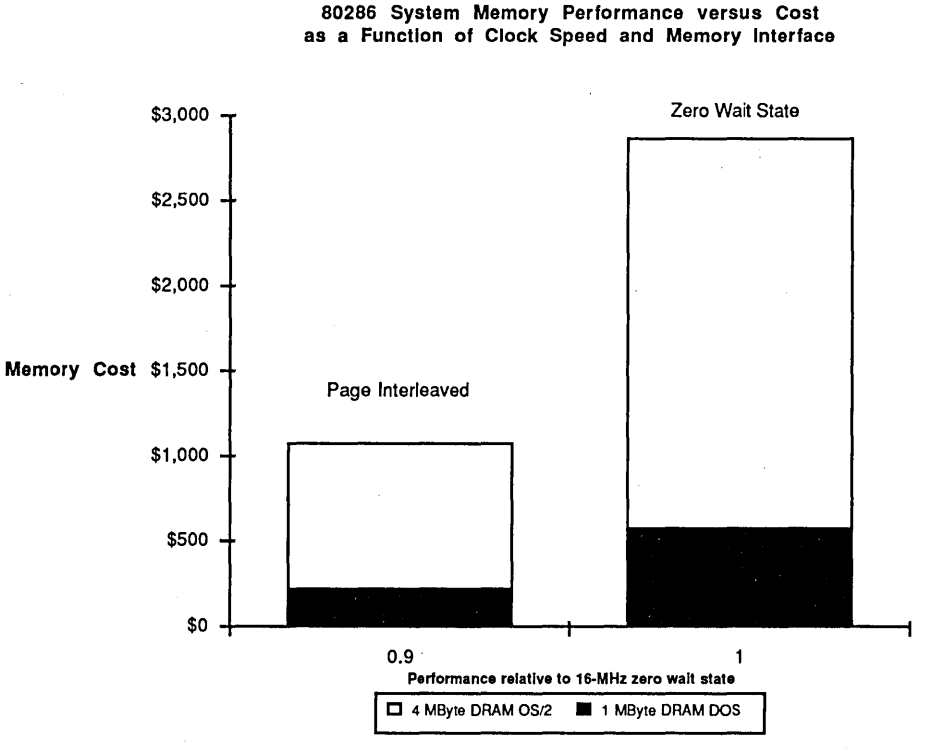


Figure 8

SYSTEM DESIGN

Lastly, let's consider a system design implementing a high-performance chip set of the PC-AT control logic, including a page-interleaved memory controller. The chip set components include a System Controller, a Page-interleaved Memory Controller, a Data/Address Buffer, and a Peripheral Controller. Some chip sets include the address buffer and the memory controller on the same chip.

Using a chip set has several advantages: it reduces chip count, increases flexibility, and improves operating speed. Most chip sets replace 40–60% of the components on a typical AT motherboard. Many chip sets also offer a migration path to 20- and 25-MHz systems.

The Peripheral Controller contains the Direct Memory Access (DMA) controller (82C37A) (used in time-critical applications allowing data to be exchanged directly between the disk drives and the computer's memory without having to pass through the microprocessor), and Interrupt Controller (82C59A). The System Controller contains all the clock generation logic of an AT-compatible system. It replaces an 82C284 Clock Generator, an 82C288 Bus Controller, and wait state and refresh controls. The Data/Address Buffer provides the buffering and latching between the local CPU address bus and the peripheral address bus (29843). It also provides buffering between the local CPU data bus and the memory data bus (29863). Parity bit generation and error detection logic also reside on this chip. The Memory Controller, as previously explained, provides interleaved memory design with page mode operation.

A block diagram of a typical system is shown in Figure 9 using standard components; the chip set equivalent parts are highlighted.

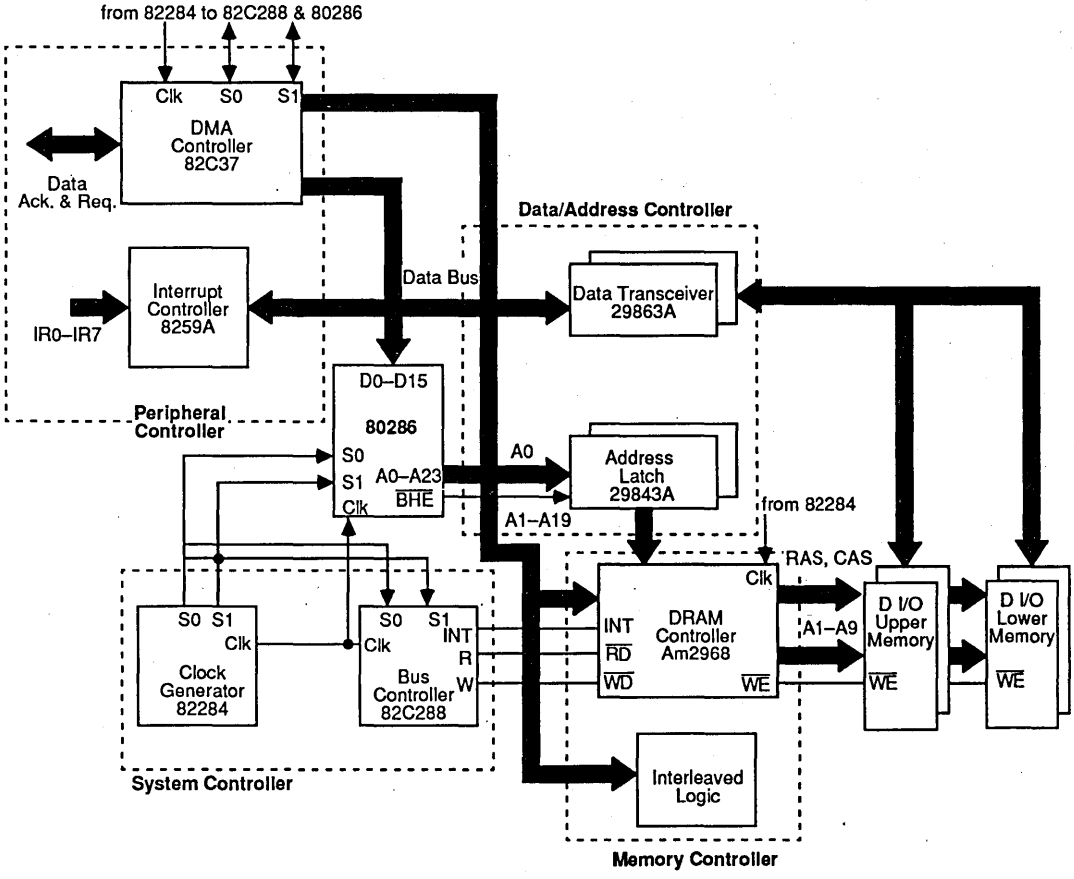


Figure 9

80C287 Coprocessor/Software Accelerator Performance Benchmarks

by Linda Bishop

INTRODUCTION

The math coprocessor/software accelerator performing floating-point arithmetic helps your 80286-based computer do more work in less time. A computer with a high-speed math coprocessor installed can perform floating-point calculations up to 1000% faster than it can without a coprocessor.

Coprocessing is becoming more important. More applications demand more processing power to execute complex mathematical tasks. Business spreadsheets and financial models; statistical and accounting packages; and engineering applications, such as computer-aided design (CAD), require high computational speed from your computer. Hence 80287 usage is increasing, and using faster 80287 coprocessors fulfills these computational needs better.

Using an AMD 80C287 math coprocessor is like using a second CPU specifically for those number crunching tasks. A math coprocessor works hand-in-hand with the CPU to speed up your computer's ability to perform calculations. When your computer performs certain types of math calculations, it triggers execution by the 80C287 rather than the CPU. The 80C287, in turn, performs the calculations more rapidly and delivers the answer to the CPU. Sometimes the need for a faster processor (CPU) can be eliminated by adding a high-speed coprocessor.

This application note discusses the performance advantage of using a math coprocessor in a PC system, showing several benchmarks as examples.

Considerations for incorporating a 16.67 MHz AMD 80C287 or 80EC287 in new system designs is discussed.

An AMD daughterboard design is introduced to allow an existing PC system of any speed to benefit from the high-speed AMD 16.67 MHz 80C287 and 80EC287 devices.

Software applications and development tools for coprocessors are also discussed. Attached is an appendix of 137 software packages that take advantage of a math coprocessor.

As time and technologies advance, the demand for system performance is shared with system reliability demands. System reliability ranks on par with performance. Therefore, reduced component heat dissipation becomes important especially for laptop PCs where the additional criteria of battery life is paramount. With the increased use of graphics in applications and operating environments, the need for faster numeric calculations has increased. Incorporating a CMOS math coprocessor into an 80286-based system fulfills both requirements by greatly enhancing computational speed and offering lower power/heat dissipation than an NMOS version.

An 80286 or 80C286 CPU, although powerful, was not designed to handle complex math operations rapidly. Enter the Math Coprocessor. A system will perform floating-point, transcendental, and integer calculations significantly faster and with much greater precision if a math coprocessor is linked to it. A math coprocessor works hand-in-hand with the CPU to speed up a computer's ability to perform calculations. The coprocessor extends the register and instruction set of the microprocessor system for existing data types and adds new data types as well. It extends the microprocessor data types to process 32-, 64-, 80-bit floating-point, 32- and 64-bit integer, and 18-digit binary coded decimal (BCD) numbers.

The more numbers an application juggles, the longer it takes to perform calculations. When software, written to use a coprocessor, performs mathematical operations, the coprocessor rather than the CPU is engaged. The coprocessor often performs the calculations as much as 1000% faster than the CPU. Coprocessors also save programming time, because trigonometric, logarithmic, and exponential functions are built into the coprocessor's instruction set.

THE AMD CMOS COPROCESSOR/ SOFTWARE ACCELERATORS

The AMD CMOS 80C287 math coprocessor/software accelerator is offered in two versions. The first, the 80C287, is directly compatible with the Intel NMOS 80287 coprocessor, where an extensive PC-installed base exists for their use. AMD's 80C287 is offered as a 6-,8-,10-MHz device and as a high-speed 16 MHz . The AMD 80C287 consumes 26% less power than the Intel 80287, reducing the heat of the device and total system and thereby potentially improving reliability.

The second version, the 80EC287, is an enhanced low-power version with a sleep feature, perfect for battery-powered laptop PCs. With the sleep feature, the coprocessor clock runs only when an instruction is executing, automatically shutting itself off when the instruction bus is idle and thus saving power consumption and valuable battery life. This feature is completely transparent to the user and system designer. The AMD 80EC287 is functionally and pin equivalent to the AMD 80C287. The 80EC287 operates in speed versions of 12.5 and 16.67 MHz.

Publication #	Rev.	Amendment	Issue Date:
12640	A	/0	July 1989

BENCHMARK PERFORMANCE

Figures 1 through 3 illustrate two different speed 80287 and 80C287 coprocessors showing improvements over systems with no coprocessor while executing mathematical operations. Floating-point calculation times decrease dramatically when a coprocessor is added. The faster the coprocessor, the better the performance. Eight-MHz and 16-MHz coprocessor speed versions are compared.

Figure 1 shows the *PC Magazine* Benchmark Series Release 4, floating-point benchmark program. This benchmark measures coprocessor speed by looping through a series of floating-point calculations, including multiplication, division, exponentiation, and logarithmic and trigonometric functions. From the chart one can see that a 16-MHz 80C287 results in a 1000% speed increase over no coprocessor.

Figure 2 compares coprocessor speeds using the *PC Tech Journal* HLBENCH benchmark "HLFLOAT." This benchmark performs a Fast Fourier Transform (FFT) on data it generates internally. By way of background, a Discrete Fourier Transform (DFT) takes a set of data points (sampled from a signal at equal intervals) and by a series of mathematical operations, generates the signal's frequency spectrum. The frequency spectrum is then "low-pass" filtered; all components above the eleventh harmonic are deleted. Finally the HLFLOAT reconstructs the time domain signal by running a reverse FFT. The result, a low-pass filtered version of the original signal, is compared to the original.

As one can see from Figure 2, using a 16-MHz coprocessor results in an 860% speed increase over no coprocessor.

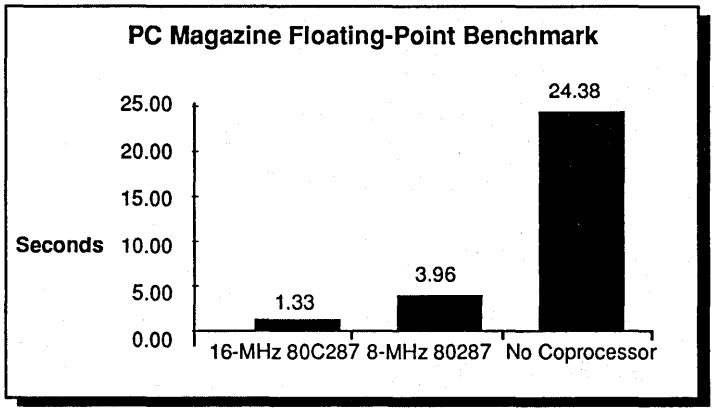


Figure 1

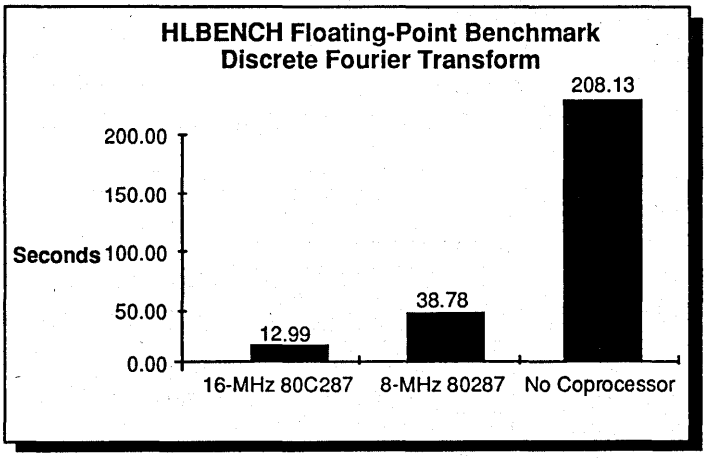


Figure 2

Figure 3 shows the Whetstone test. This benchmark is a traditional test of a systems ability to perform floating-point calculations. Results are indicated in thousand Whetstones per second. Again, the 16-MHz coprocessor shows a 720% performance increase over no coprocessor.

SYSTEM CONFIGURATION

The 80C287 is an extension of the 80286 microprocessor, sharing the same memory and data buses. Certain signals allow the 80C287 to receive instructions and data from the CPU. Application software must be written to utilize a coprocessor's numeric instruction set. However, all communication between the 80286 and 80C287 is transparent to applications software. Whenever a numeric instruction is executed, the CPU automatically looks for and controls the 80C287. Once the 80C287 has been activated, it can process data in parallel with, and independent of, the host CPU.

The 80C287 and 80EC287 work at one-third the frequency of the incoming clock. Thus, to utilize the 80C287 at its highest speed of 16.67 MHz, the 80C287 should be configured to run off a clock of 50 MHz.

THE INSTALLED BASE

Existing 80286-based systems, however, have not been designed with the high-speed dedicated clock necessary to run the 80C287 at 16 MHz. In most of the installed 80286 computer base, the coprocessor socket is configured to be driven by the CPU clock at speeds of 6, 8, or 10 MHz, depending on the particular PC.

These systems can benefit from the high-speed AMD 80C287–16, however, by using an AMD-developed daughterboard. The daughterboard, equal in size to the 80C287 40-pin DIP footprint, is comprised of an 80C287 socket and a 50-MHz oscillator mounted underneath the 80C287 device, as shown in Figure 4.

The board can plug directly into any 80287 motherboard socket. All of the motherboard signals pass unchanged to the 80C287 with the exception of the clock input. The clock is driven directly from the 50-MHz on-board oscillator. In this way the 80C287 will operate at full speed regardless of the clock speed selected by the motherboard jumper.

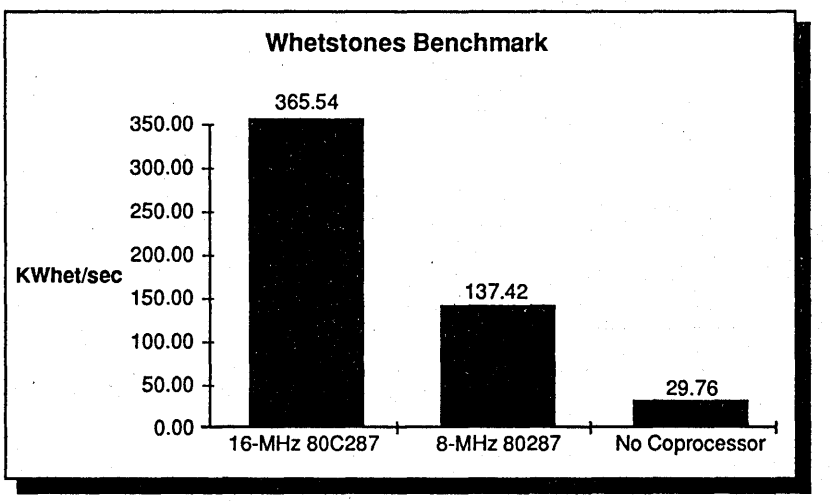


Figure 3

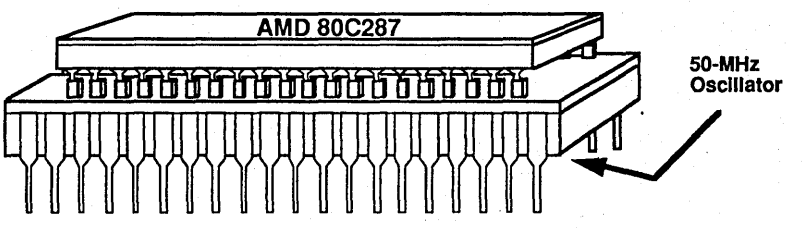


Figure 4

SOFTWARE APPLICATIONS AND DEVELOPMENT TOOLS

Hundreds of application programs have been written to take advantage of the coprocessor's speed and precision, including business, engineering, graphics, and statistical packages.

Compilers and assemblers benefit from a coprocessor as well. High-level languages are greatly simplified with a coprocessor. Instructions such as sine, square root, hyperbolic tangent, and log are built into the library routine of the coprocessor. Most assembly language programs have the option of writing code reflecting coprocessor instructions, or by linking in a developer's math library or one supplied by compilers.

Most high-level languages link an emulation library into any program containing floating-point instructions or data. On start-up, code is generated to check for the presence of a coprocessor. If one is detected, it is used. If a coprocessor is not detected, the emulation library is used. In this way, programs written to take advantage of a coprocessor can run on systems without one.

See also *Comparing AMD's CMOS 80C287 and 80EC287 Math Coprocessors to the Intel NMOS 80287 and CMOS 80C287A Application Note*, order #12190A, available from your local AMD sales office or call (800) 538-8450.

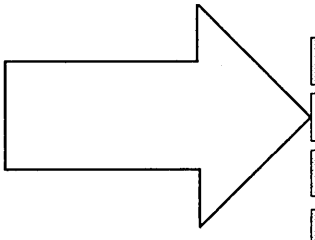
APPENDIX

Company	Application
Programming Tools	
Alsys, Inc.	Ada® Compiler
Borland International	Turbo Pascal, Turbo Assembler, Turbo C
CACI Products	Simscript II.5
Computer Innovations, Inc.	C86PLUS
C Ware Corporation	DC88
FORTH, Inc.	polyFORTH II
Gold Hill Computers	Golden Common LISP, Goldworks
IBM	TopView™, Graphical Kernal System, Macro Assembler, RDT Resident Debugging Tool, Plotting System
Integral Quality, Inc.	IQLISP, IQCLISP
Laboratory Microsystems	PC/Forth, UR/Forth
Lahey Computer Systems, Inc.	F77L-EM/16, F77L-EM/32, Lahey Personal Fortran
Lattice, Inc.	C Compiler
Logitech	Modula-2
Mark Williams Co.	Let's C
Micro Focus, Inc.	CO-Math
Microsoft	FORTRAN, Pascal, C, Quick BASIC
Microtech Research, Inc.	ASM186™, Paragon PP/86, MCC86
Morgan Computing Company, Inc.	Professional BASIC
Pecan Software	Assembler Modula-2, Fortran 77, UCSD Pascal, Power System
RR software, Inc.	Janus Ada (Pharlap/Xenix), Systems Pack
Ryan McFarland Corp.	RM/FORTRAN, RM COBOL
STSC, Inc.	APL *PLUS/PC
SuperSoft, Inc.	FORTRAN, C
TeleSoft	TeleSoft-Ada, Pascal
True Basic, Inc.	True BASIC
Unisource Software Corp.	VENIX/86
WATCOM	WATCOM APL, WATCOM BASIC, WATFOR-77, WATCOM COBOL, WATCOM Pascal
Whitesmiths, Ltd.	C
Financial Analysis and Planning	
Alcar Group, Inc.	The Value Planner, The Merger Planner, The Strategy Planner
Black River Systems Corp.	MACRO*WORLD FORECASTER, MACRO*WORLD INVESTOR
Chase Manhattan Bank	Chase Duration, Cap Floor Calculator, BRMS
Dac Software, Inc.	DacEasy Payroll, DacEasy Accounting
Intex Solutions, Inc.	Forecast!
MECA Ventures, Inc.	Managing Your Money
Sterling Wentworth	Step1, PlanMan, Next$, Business Plan

Company	Application
Business Graphics	
Advanced Graphics Software	SlideWrite Plus
Bitstream	Fontware
Computer Support Corp.	Diagraph, Arts and Letters, Picture Perfect
Custom Applications, Inc.	Freedom of Press
LaserGo, Inc.	GoScript
Lotus Development Corp.	Graphwriter II, Freelance Plus
New England Software	Graph-in-the-box
Softcraft, Inc.	Font Effects, SpinFont, WYSIfonts!
Zenographics	Autumn, Mirage, Pixy
Statistical	
Anderson-Bell	Abstat
Aptech Systems	Gauss
BMDP Statistical Software, Inc.	BMDP Statistical Software
Mikros, Inc.	ESP
P-Stat, Inc.	P-Stat
SAS Institute, Inc.	PC SAS System
SPSS, Inc.	SPSS/PC
Strategy Plus	Exec*U*Stat
STSC, Inc.	STATGRAPHICS
SYSTAT, Inc.	SYSTAT
Analysis	
IMPULSE Engineering	Fortran Addenda
Microcompatibles	PLOTMATIC, OMNIPLOT, GRAFMATIC
Wiley Professional Software	Statlib.gl, Statlib.tsf, Scientific Subroutine Library, Mathematical Function Library
DataBase Management and Analysis	
Ashton-Tate	dBASE™
Borland International	Reflex, Paradox®
DataEase International, Inc.	DataEase
Economica	DX
Fox Software	FoxBase+
Information Builders, Inc.	PC/FOCUS
Mainstay Software Corp.	Mainstay
Microrim	CLOUT, DB Graphics, R:base for DOS
Migent Software, Inc.	Emerald Bay
Nantucket Corp.	Clipper
Oracle Corp.	Professional Oracle
Revelations Technology	Advanced Revelation
Computer Aided Design	
AutoDesk	AutoCAD, Autosketch
CadKey	CadKey
MathSOFT	MathCAD
MegaCadd	MegaSHADE, MegaDRAFT, MegaMODEL
Robo Systems	RoboCAD-PC
Universal Integraphix Corp.	3-D Graphixx CADD
Versacad Corp.	VersaCAD Design

Company	Application
Integrated Software	
Ashton-Tate	Framework
Informix Software	SmartWare
Lemain, Inc.	Corporate MBA
Lotus Development Corp.	Symphony
Microsoft	Microsoft Works
Migent Software, Inc.	Ability Plus
Mosaic Marketing, Inc.	INTEGRATED 7
Software Group	Enable
Mathematical Modeling	
BMDP Statistical Software, Inc.	EQF
Borland International	Eureka: The Solver
Eastern Software Products, Inc.	LP88, BLP88, TSA88
General Optimization	What's Best
Pacific Crest Software	Point FIVE
Universal Technical Systems, Inc.	TK Solver Plus
WEFA Group	Aremos, PC Model
Wolfram Research, Inc.	Mathematica
Scientific Engineering	
Alligator Transforms	Fourier Perspectives, Prime Factor FFT
Asyst Software Technologies, Inc.	Asystant, Asyst
Celestial Software	IMAGES-3D
Compact Software	Super-Compact PC, AutoArt PC
Das Consulting	DAST100
IMPULSE Engineering	Steam 85
MacNeal-Schwendler Corp.	MSC/cal, MSC/mod, MSC/pal 2
Mathworks, Inc.	Mathlab
Synoptics	Semper 6 Plus
Wescom	Wescom
Spreadsheets	
Borland International	Quattro
Computer Associates International	SuperCalc
Dac Software, Inc.	Lucid 3-D
Daybreak Technologies	Silk
Frontline Systems	3-2-1 Blastoff
Javelin Software Corp.	Javelin
Lotus Development Corp.	Lotus 1-2-3™
Microsoft	Multiplan, Excel
Mosaic Marketing, Inc.	The TWIN
Paperback Software International	VP Planner Plus
Software Publishing Corp.	PFS: Professional Plan
Unisource Software Corp.	Tactician (Unix)

Advanced
Micro
Devices



CHAPTER 2
SYSTEM SUPPORT PERIPHERALS

Am29C325

CMOS 32-Bit Floating-Point Processor

FINAL

DISTINCTIVE CHARACTERISTICS

- Single VLSI device performs high-speed single-precision floating-point arithmetic
  Floating-point addition, subtraction, and multiplication in a single clock cycle
  Internal architecture supports sum-of-products, Newton-Raphson division
- 32-bit, three-bus flow-through architecture
  Programmable I/O allows interface to 32- and 16-bit systems
- IEEE and DEC formats
  Performs conversions between formats
  Performs integer ↔ floating-point conversions
- Input and output registers may be made transparent independently
- Pin and functionally compatible with the bipolar Am29325
- The Am29C325 uses less than one-quarter the power of the Am29325
- 145 PGA requires no heatsink

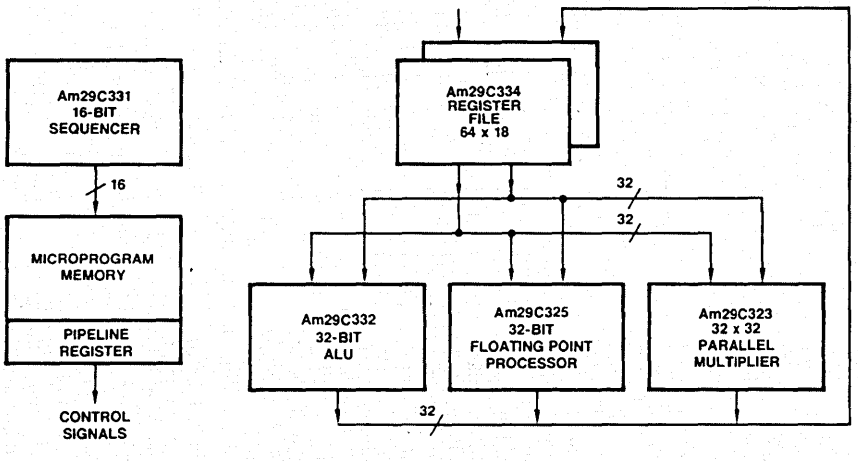
GENERAL DESCRIPTION

The Am29C325 is a high-speed floating-point processor unit. It performs 32-bit single-precision floating-point addition, subtraction, and multiplication operations in a single VLSI circuit, using the format specified by the IEEE floating-point standard, 754. The DEC single-precision floating-point format is also supported. Operations for conversion between 32-bit integer format and floating-point format are available, as are operations for converting between the IEEE and DEC floating-point formats. Any instruction can be performed in a single clock cycle. Six flags — invalid operation, inexact result, zero, not-a-number, overflow, and underflow — monitor the status of operations.

The Am29C325 has a three-bus, 32-bit architecture, with two input buses and one output bus. This configuration provides high I/O bandwidth, allows access to all buses, and affords a high degree of flexibility when connecting this device in a system. All buses are registered, with each register having a clock enable. Input and output registers may be made transparent independently. Two other I/O configurations, a 32-bit, two-bus architecture and a 16-bit, three-bus architecture, are user-selectable, easing interface with a wide variety of systems. Thirty-two-bit internal feedforward datapaths support accumulation operations, including sum-of-products and Newton-Raphson division.

Fabricated using Advanced Micro Devices' 1.2 micron CMOS process, the Am29C325 is powered by a single 5-volt supply. The device is housed in a 145-lead pin-grid-array package.

Am29C300 FAMILY HIGH-PERFORMANCE SYSTEM BLOCK DIAGRAM



AF004651

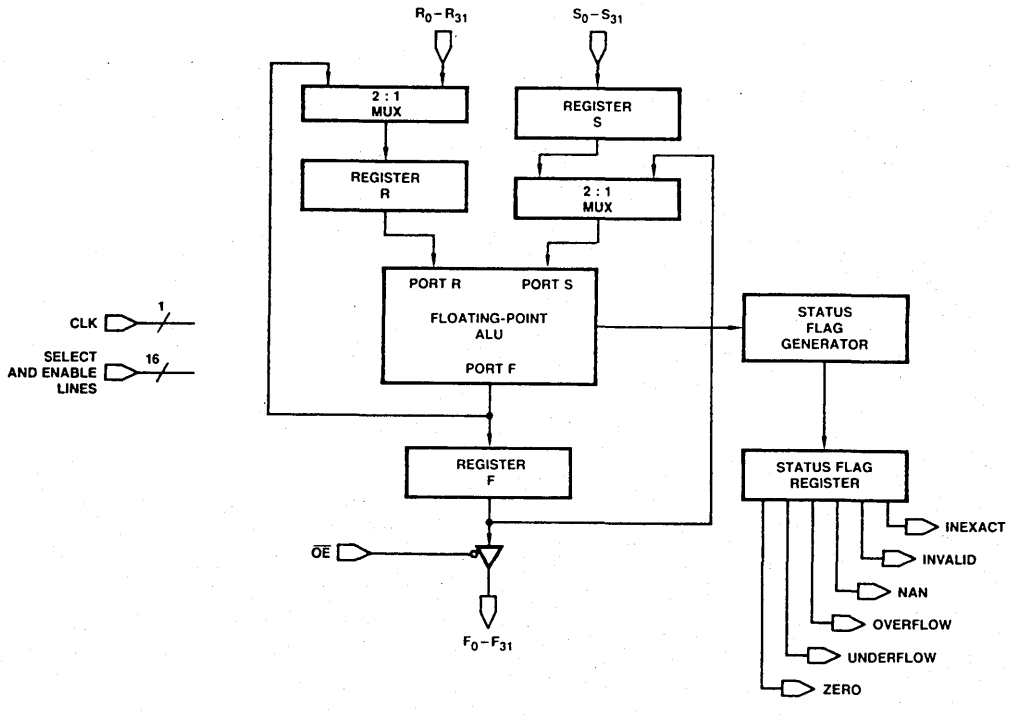
Publication #	Rev.	Amendment
07783	C	/0

Issue Date: March 1989

RELATED AMD PRODUCTS

Part No.	Description
Am29027	Am29K Arithmetic Accelerator
Am29116	High-Performance Bipolar 16-Bit Microprocessor
Am29C116	High-Performance CMOS 16-Bit Microprocessor
Am29CPL141	CMOS 64 x 16 EPROM Field Programmable Controller
Am29CPL142	CMOS 128 x 16 EPROM Field Programmable Controller
Am29CPL144	CMOS 512 x 16 EPROM Field Programmable Controller
Am29CPL151	CMOS 64 x 16 EPROM Field Programmable Controller — Slim DIP
Am29CPL152	CMOS 128 x 16 EPROM Field Programmable Controller — Slim DIP
Am29CPL154	CMOS 512 x 16 EPROM Field Programmable Controller — Slim DIP
Am29C323	CMOS 32-Bit Parallel Multiplier
Am29C327	CMOS Double-Precision Floating-Point Processor
Am29331	16-Bit Microprogram Sequencer
Am29C331	CMOS 16-Bit Microprogram Sequencer
Am29332	32-Bit Extended Function ALU
Am29C332	CMOS 32-Bit Extended Function ALU
Am29334	64 x 18 Four-Port, Dual-Access Register File
Am29C334	CMOS 64 x 18 Four-Port, Dual-Access Register File

BLOCK DIAGRAM



BD007080

CONNECTION DIAGRAM
Top View

PGA

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R
1	INEX	I2	I1	$\overline{ENF}$	I4	OBUS	$\overline{OE}$	VCC	CLK	R31	R30	R25	R24	R21	R20
2	INVA	NAN	I0	I/$\overline{D}$	FT0	FT1	VCC	VCC	RND0	RND1	R27	R28	R23	R22	R17
3	F29	ZERO	GNDO	$\overline{ENR}$	$\overline{ENS}$	16/$\overline{32}$	VCC	VCC	VCC	R29	R26	GND	GND	R19	R18
4	F30	F31	GNDO	*									R15	R16	R13
5	F23	OVFL	UNFL										R14	R11	R12
6	F26	F27	F28										R9	R10	R7
7	F21	F24	F25										R8	R5	R6
8	F22	F19	VCCO										R3	R4	R1
9	F17	F20	VCCO										R0	I3	R2
10	F18	F15	F16										S28	S31	S30
11	F13	F14	F11										S27	S26	S29
12	F12	F9	F10										VCC	S25	S24
13	F7	F6	GNDO	GNDO	GNDO	GNDO	GND	GND	GND	S8	S13	S14	VCC	S22	S23
14	F8	F3	F2	GNDO	F0	S1	S2	GND	S4	S9	S10	S15	S18	S21	S20
15	F5	F4	F1	GNDO	P/$\overline{AFF}$	S0	S3	S5	S7	S6	S11	S12	S17	S16	S19

CD010492

Key: 16/$\overline{32}$ = S16/$\overline{32}$
I/$\overline{D}$ = IEEE/$\overline{DEC}$
INEX = INEXACT
INVA = INVALID
OBUS = ONEBUS
OVFL = OVERFLOW
P/$\overline{AFF}$ = PROJ/$\overline{AFF}$
UNFL = UNDERFLOW

*D4 is an alignment pin (not connected internally).

PIN DESIGNATIONS

(Sorted by Pin No.)

PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME
A-1	Inexact	**C-7**	F_{25}	**H-13**	GND	**N-10**	S_{28}
A-2	Invalid	**C-8**	V_{CCO}	**H-14**	GND	**N-11**	S_{27}
A-3	F_{29}	**C-9**	V_{CCO}	**H-15**	S_5	**N-12**	V_{CC}
A-4	F_{30}	**C-10**	F_{16}	**J-1**	CLK	**N-13**	V_{CC}
A-5	F_{23}	**C-11**	F_{11}	**J-2**	RND_0	**N-14**	S_{18}
A-6	F_{26}	**C-12**	F_{10}	**J-3**	V_{CC}	**N-15**	S_{17}
A-7	F_{21}	**C-13**	GNDO	**J-13**	GND	**P-1**	R_{21}
A-8	F_{22}	**C-14**	F_2	**J-14**	S_4	**P-2**	R_{22}
A-9	F_{17}	**C-15**	F_1	**J-15**	S_7	**P-3**	R_{19}
A-10	F_{18}	**D-1**	$\overline{ENF}$	**K-1**	R_{31}	**P-4**	R_{16}
A-11	F_{13}	**D-2**	IEEE/$\overline{DEC}$	**K-2**	RND_1	**P-5**	R_{11}
A-12	F_{12}	**D-3**	$\overline{ENR}$	**K-3**	R_{29}	**P-6**	R_{10}
A-13	F_7	**D-13**	GNDO	**K-13**	S_8	**P-7**	R_5
A-14	F_8	**D-14**	GNDO	**K-14**	S_9	**P-8**	R_4
A-15	F_5	**D-15**	GNDO	**K-15**	S_6	**P-9**	I_3
B-1	I_2	**E-1**	I_4	**L-1**	R_{30}	**P-10**	S_{31}
B-2	NAN	**E-2**	FT_0	**L-2**	R_{27}	**P-11**	S_{26}
B-3	ZERO	**E-3**	$\overline{ENS}$	**L-3**	R_{26}	**P-12**	S_{25}
B-4	F_{31}	**E-13**	GNDO	**L-13**	S_{13}	**P-13**	S_{22}
B-5	OVERFLOW	**E-14**	F_0	**L-14**	S_{10}	**P-14**	S_{21}
B-6	F_{27}	**E-15**	PROJ/$\overline{AFF}$	**L-15**	S_{11}	**P-15**	S_{16}
B-7	F_{24}	**F-1**	ONEBUS	**M-1**	R_{25}	**R-1**	R_{20}
B-8	F_{19}	**F-2**	FT_1	**M-2**	R_{28}	**R-2**	R_{17}
B-9	F_{20}	**F-3**	S16/$\overline{32}$	**M-3**	GND	**R-3**	R_{18}
B-10	F_{15}	**F-13**	GNDO	**M-13**	S_{14}	**R-4**	R_{13}
B-11	F_{14}	**F-14**	S_1	**M-14**	S_{15}	**R-5**	R_{12}
B-12	F_9	**F-15**	S_0	**M-15**	S_{12}	**R-6**	R_7
B-13	F_6	**G-1**	$\overline{OE}$	**N-1**	R_{24}	**R-7**	R_6
B-14	F_3	**G-2**	V_{CC}	**N-2**	R_{23}	**R-8**	R_1
B-15	F_4	**G-3**	V_{CC}	**N-3**	GND	**R-9**	R_2
C-1	I_1	**G-13**	GND	**N-4**	R_{15}	**R-10**	S_{30}
C-2	I_0	**G-14**	S_2	**N-5**	R_{14}	**R-11**	S_{29}
C-3	GNDO	**G-15**	S_3	**N-6**	R_9	**R-12**	S_{24}
C-4	GNDO	**H-1**	V_{CC}	**N-7**	R_8	**R-13**	S_{23}
C-5	UNDERFLOW	**H-2**	V_{CC}	**N-8**	R_3	**R-14**	S_{20}
C-6	F_{28}	**H-3**	V_{CC}	**N-9**	R_0	**R-15**	S_{19}

Note: Pin number D4 = Alignment Pin
V_{CCO} and GNDO are power and ground pins for the output buffers.
V_{CC} and GND are power and ground pins for the rest of the logic.

PIN DESIGNATIONS (Cont'd.)

(Sorted by Pin Name)

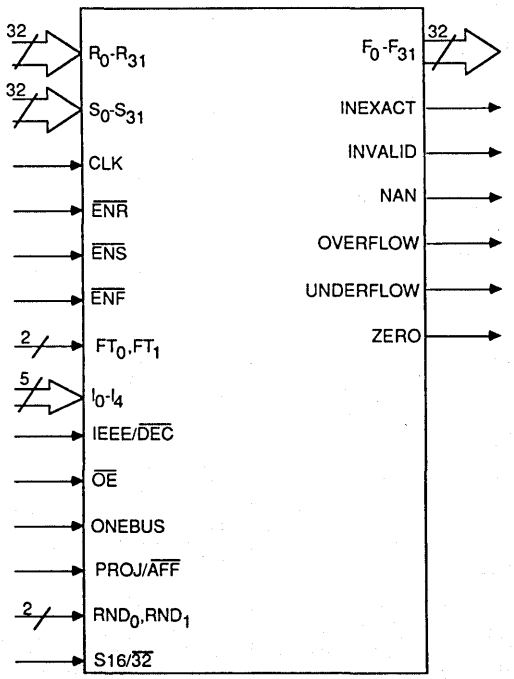
PIN NO.	PIN NAME	PIN NO.	PIN NAME.	PIN NO.	PIN NAME	PIN NO.	PIN NAME.
J-1	CLK	E-2	FT_0	R-6	R_7	K-14	S_9
D-1	$\overline{ENF}$	F-2	FT_1	N-7	R_8	L-14	S_{10}
D-3	$\overline{ENR}$	G-13	GND	N-6	R_9	L-15	S_{11}
E-3	$\overline{ENS}$	H-13	GND	P-6	R_{10}	M-15	S_{12}
E-14	F_0	H-14	GND	P-5	R_{11}	L-13	S_{13}
C-15	F_1	J-13	GND	R-5	R_{12}	M-13	S_{14}
C-14	F_2	M-3	GND	R-4	R_{13}	M-14	S_{15}
B-14	F_3	N-3	GND	N-5	R_{14}	P-15	S_{16}
B-15	F_4	C-3	GNDO	N-4	R_{15}	F-3	$S16/\overline{32}$
A-15	F_5	C-4	GNDO	P-4	R_{16}	N-15	S_{17}
B-13	F_6	C-13	GNDO	R-2	R_{17}	N-14	S_{18}
A-13	F_7	D-13	GNDO	R-3	R_{18}	R-15	S_{19}
A-14	F_8	D-14	GNDO	P-3	R_{19}	R-14	S_{20}
B-12	F_9	D-15	GNDO	R-1	R_{20}	P-14	S_{21}
C-12	F_{10}	E-13	GNDO	P-1	R_{21}	P-13	S_{22}
C-11	F_{11}	F-13	GNDO	P-2	R_{22}	R-13	S_{23}
A-12	F_{12}	C-2	I_0	N-2	R_{23}	R-12	S_{24}
A-11	F_{13}	C-1	I_1	N-1	R_{24}	P-12	S_{25}
B-11	F_{14}	B-1	I_2	M-1	R_{25}	P-11	S_{26}
B-10	F_{15}	P-9	I_3	L-3	R_{26}	N-11	S_{27}
C-10	F_{16}	E-1	I_4	L-2	R_{27}	N-10	S_{28}
A-9	F_{17}	D-2	IEEE/$\overline{DEC}$	M-2	R_{28}	R-11	S_{29}
A-10	F_{18}	A-1	INEXACT	K-3	R_{29}	R-10	S_{30}
B-8	F_{19}	A-2	INVALID	L-1	R_{30}	P-10	S_{31}
B-9	F_{20}	B-2	NAN	K-1	R_{31}	C-5	UNDERFLOW
A-7	F_{21}	G-1	$\overline{OE}$	J-2	RND_0	G-2	V_{CC}
A-8	F_{22}	F-1	ONEBUS	K-2	RND_1	G-3	V_{CC}
A-5	F_{23}	B-5	OVERFLOW	F-15	S_0	H-1	V_{CC}
B-7	F_{24}	E-15	PROJ/$\overline{AFF}$	F-14	S_1	H-2	V_{CC}
C-7	F_{25}	N-9	R_0	G-14	S_2	H-3	V_{CC}
A-6	F_{26}	R-8	R_1	G-15	S_3	J-3	V_{CC}
B-6	F_{27}	R-9	R_2	J-14	S_4	N-12	V_{CC}
C-6	F_{28}	N-8	R_3	H-15	S_5	N-13	V_{CC}
A-3	F_{29}	P-8	R_4	K-15	S_6	C-8	V_{CCO}
A-4	F_{30}	P-7	R_5	J-15	S_7	C-9	V_{CCO}
B-4	F_{31}	R-7	R_6	K-13	S_8	B-3	ZERO

Note: Pin number D4 = Alignment Pin

V_{CCO} and GNDO are power and ground pins for the output buffers.

V_{CC} and GND are power and ground pins for the rest of the logic.

LOGIC SYMBOL

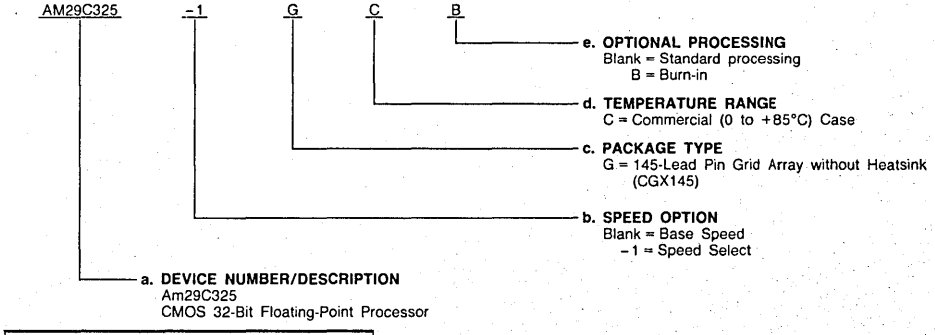


LS002920

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM29C325	GC, GCB
AM29C325-1	

Valid Combinations

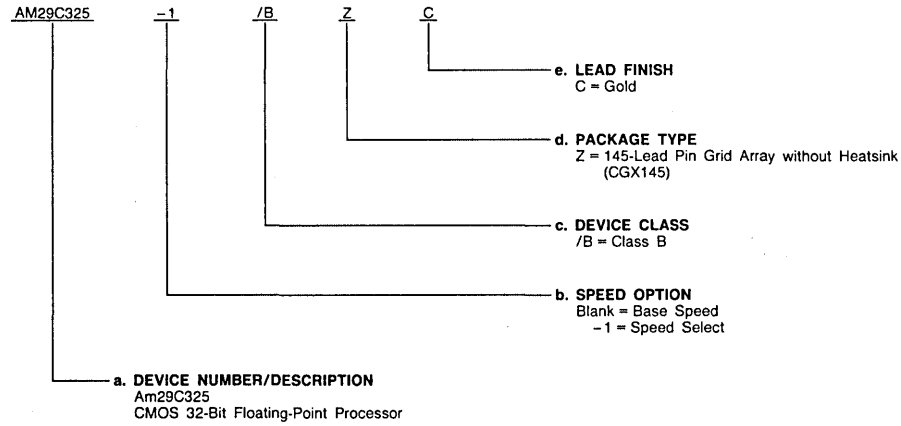
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

2

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM29C325	/BZC
AM29C325-1	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

CLK Clock (Input)
For the internal registers.

$\overline{ENF}$ Register F Clock Enable (Input; Active LOW)
When $\overline{ENF}$ is LOW, register F is clocked on the LOW-to-HIGH transition of CLK. When $\overline{ENF}$ is HIGH, register F retains the previous contents.

$\overline{ENR}$ Register R Clock Enable (Input; Active LOW)
When $\overline{ENR}$ is LOW, register R is clocked on the LOW-to-HIGH transition of CLK. When $\overline{ENR}$ is HIGH, register R retains the previous contents.

$\overline{ENS}$ Register S Clock Enable (Input; Active LOW)
When $\overline{ENS}$ is LOW, register S is clocked on the LOW-to-HIGH transition of CLK. When $\overline{ENS}$ is HIGH, register S retains the previous contents.

$F_0 - F_{31}$ F Operand Bus (Output)
F_0 is the least significant bit.

FT_0 Input Register Feedthrough Control (Input; Active HIGH)
When FT_0 is HIGH, registers R and S are transparent.

FT_1 Output Register Feedthrough Control (Input; Active HIGH)
When FT_1 is HIGH, register F and the status flag register are transparent.

$I_0 - I_2$ Operation Select Lines (Input)
Used to select the operation to be performed by the ALU. See Table 1 for a list of operations and the corresponding codes.

I_3 ALU S Port Input Select (Input)
A LOW on I_3 selects register S as the input to the ALU S port. A HIGH on I_3 selects register F as the input to the ALU S port.

I_4 Register R Input Select (Input)
A LOW on I_4 selects $R_0 - R_{31}$ as the input to register R. A HIGH selects the ALU F port as the input to register R.

IEEE/$\overline{DEC}$ IEEE/$\overline{DEC}$ Mode Select (Input)
When IEEE/$\overline{DEC}$ is HIGH, IEEE mode is selected. When IEEE/$\overline{DEC}$ is LOW, DEC mode is selected.

INEXACT Inexact Result Flag (Output; Active HIGH)
A HIGH indicates that the final result of the last operation was not infinitely precise, due to rounding.

INVALID Invalid Operation Flag (Output; Active HIGH)
A HIGH indicates that the last operation performed was invalid; e.g., ∞ times 0.

NAN Not-a-Number Flag (Output; Active HIGH)
A HIGH indicates that the final result produced by the last operation is not to be interpreted as a number. The output in such cases is either an IEEE Not-a-Number (NAN) or a DEC-reserved operand.

$\overline{OE}$ Output Enable (Input; Active LOW)
When $\overline{OE}$ is LOW, the contents of register F are placed on $F_0 - F_{31}$. When $\overline{OE}$ is HIGH, $F_0 - F_{31}$ assume a high-impedance state.

ONEBUS Input Bus Configuration Control (Input)
A LOW on ONEBUS configures the input bus circuitry for two-input bus operation. A HIGH on ONEBUS configures the input bus circuitry for single-input bus operation.

OVERFLOW Overflow Flag (Output; Active HIGH)
A HIGH indicates that the last operation produced a final result that overflowed the floating-point format.

PROJ/$\overline{AFF}$ Projective/$\overline{Affine}$ Mode Select (Input)
Choice of projective or affine mode determines the way in which infinities are handled in IEEE mode. A LOW on PROJ/$\overline{AFF}$ selects affine mode; a HIGH selects projective mode.

$R_0 - R_{31}$ R Operand Bus (Input)
R_0 is the least significant bit.

RND_0, RND_1 Rounding Mode Selects (Input)
RND_0 and RND_1 select one of four rounding modes. See Table 5 for a list of rounding modes and the corresponding control codes.

$S_0 - S_{31}$ S Operand Bus (Input)
S_0 is the least significant bit.

S16/$\overline{32}$ 16- or $\overline{32}$-Bit I/O Mode Select (Input)
A LOW on S16/$\overline{32}$ selects the 32-bit I/O mode; a HIGH selects the 16-bit I/O mode. In 32-bit mode, input and output buses are 32 bits wide. In 16-bit mode, input and output buses are 16 bits wide, with the least and most significant portions of the 32-bit input and output words being placed on the buses during the HIGH and LOW portions of CLK, respectively.

UNDERFLOW Underflow Flag (Output; Active HIGH)
A HIGH indicates that the last operation produced a rounded result that underflowed the floating-point format.

ZERO Zero Flag (Output; Active HIGH)
A HIGH indicates that the last operation produced a final result of zero.

Definition of Terms

Affine Mode

One of two modes affecting the handling of operations on infinities — see the **Operations with Infinities** section under **Operations in IEEE Mode.**

Biased Exponent

The true exponent of a floating-point number, plus a constant. For IEEE floating-point numbers, the constant is 127; for DEC floating-point numbers, the constant is 128. See also **True Exponent.**

Bus

Data input or output channel for the floating-point processor.

DEC-Reserved Operand

A DEC floating-point number that is interpreted as a symbol and has no numeric value. A DEC-reserved operand has a sign of 1 and a biased exponent of 0.

Destination Format

The format of the final result produced by the floating-point ALU. The destination format can be IEEE floating point, DEC floating point, or integer.

Final Result

The result produced by the floating-point ALU.

Fraction

The 23 least-significant bits of the mantissa.

Infinitely Precise Result

The result that would be obtained from an operation if both exponent range and precision were unbounded.

Input Operands

The value or values on which an operation is performed. For example, the addition 2 + 3 = 5 has input operands 2 and 3.

Mantissa

The portion of a floating-point number containing the number's significant bits. For the floating-point number 1.101×2^{-3}, the mantissa is 1.101.

NAN (Not-a-Number)

An IEEE floating-point number that is interpreted as a symbol and has no numeric value. A NAN has a biased exponent of 255_{10} and a non-zero fraction.

Port

Data input or output channel for the floating-point ALU.

Projective Mode

One of two modes affecting the handling of operations on infinities — see the **Operations with Infinities** section under **Operation in IEEE Mode.**

Rounded Result

The result produced by rounding the infinitely precise result to fit the destination format.

True Exponent (or Exponent)

Number representing the power of two by which a floating-point number's mantissa is to be multiplied. For the floating-point number 1.101×2^{-3}, the true exponent is −3.

FUNCTIONAL DESCRIPTION

Architecture

The Am29C325 comprises a high-speed, floating-point ALU, a status flag generator, and a 32-bit data path.

Floating-Point ALU

The floating-point ALU performs 32-bit floating-point operations. It also performs floating-point-to-integer conversions, integer-to-floating-point floating-point conversions, and conversions between the IEEE and DEC formats. The ALU has two 32-bit input ports, R and S, and a 32-bit output port, F.

Conceptually, the process performed by the ALU can be divided into three stages (see Figure 1). The operation stage performs the arithmetic operation selected by the user; the output of this section is referred to as the infinitely precise result of the operation. The rounding stage rounds the infinitely precise result to fit in the destination format; the output of this stage is called the rounded result. The last stage checks for exceptional conditions. If no exceptional condition is found, the rounded result is passed through this stage. If some exceptional condition is found (e.g., overflow, underflow, or an invalid operation), this section may replace the rounded result with another output, such as +∞, −∞, a NAN, or a DEC-reserved operand. The output of this last stage appears on port F and is called the final result.

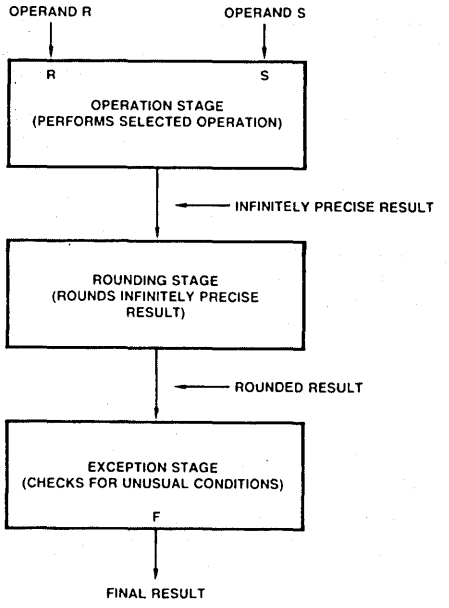


Figure 1. Conceptual Model of the Process Performed by the Floating-Point ALU

The ALU performs one of eight operations; the operation to be performed is selected by placing the appropriate control code on lines I_0–I_2. Table 1 gives the control codes corresponding to each of the eight operations.

The floating-point addition operation (R PLUS S) adds the floating-point numbers on ports R and S and places the floating-point result on port F. In IEEE mode (IEEE/$\overline{\text{DEC}}$ = HIGH) the addition is performed in IEEE floating-point format; in DEC mode (IEEE/$\overline{\text{DEC}}$ = LOW) the addition is performed in DEC format.

The floating-point subtraction operation (R MINUS S) subtracts the floating-point number on port S from the floating-point number on port R and places the floating-point result on port F. In IEEE mode (IEEE/$\overline{\text{DEC}}$ = HIGH) the subtraction is performed in IEEE floating-point point format; in DEC mode (IEEE/$\overline{\text{DEC}}$ = LOW) the subtraction is performed in DEC format.

The floating-point multiplication operation (R TIMES S) multiplies the floating-point numbers on ports R and S and places the floating-point result on port F. In IEEE mode (IEEE/$\overline{\text{DEC}}$ = HIGH) the multiplication is performed in IEEE floating-point format; in DEC mode (IEEE/$\overline{\text{DEC}}$ = LOW) the multiplication is performed in DEC format.

The floating-point constant subtraction (2 MINUS S) operation subtracts the floating-point value on port S from 2 and places the result on port F. The operand on port R is not used in this operation; its value will not affect the operation in any way. In IEEE mode (IEEE/$\overline{\text{DEC}}$ = HIGH) the operation is performed in IEEE floating-point format; in DEC mode (IEEE/$\overline{\text{DEC}}$ = LOW) the operation is performed in DEC format. This operation is

used to support Newton-Raphson floating-point division; a description of its use appears in **Appendix C.**

The integer-to-floating-point conversion (INT-TO-FP) operation takes a 32-bit, two's complement integer on port R and places the equivalent floating-point value on port F. The operand on port S is not used in this operation; its value will not affect the operation in any way. In IEEE mode (IEEE/$\overline{DEC}$ = HIGH) the result is delivered in IEEE format; in DEC mode (IEEE/$\overline{DEC}$ = LOW) the result is delivered in DEC format.

TABLE 1. ALU OPERATION SELECT

I_2	I_1	I_0	Operation	Output Equation
0	0	0	Floating-point addition (R PLUS S)	F = R + S
0	0	1	Floating-point subtraction (R MINUS S)	F = R - S
0	1	0	Floating-point multiplication (R TIMES S)	F = R * S
0	1	1	Floating-point constant subtraction (2 MINUS S)	F = 2 - S
1	0	0	Integer-to-floating-point conversion (INT-TO-FP)	F (floating-point) = R (integer)
1	0	1	Floating-point-to-integer conversion (FP-TO-INT)	F (integer) = R (floating-point)
1	1	0	IEEE-TO-DEC format conversion (IEEE-TO-DEC)	F (DEC format) = R (IEEE format)
1	1	1	DEC-TO-IEEE format conversion (DEC-TO-IEEE)	F (IEEE format) = R (DEC format)

The floating-point-to-integer conversion (FP-TO-INT) operation takes a floating-point number on port R and places the equivalent 32-bit, two's complement integer value on port F. The operand on port S is not used in this operation; its value will not affect the operation in any way. In IEEE mode (IEEE/$\overline{DEC}$ = HIGH) the operand on port R is interpreted using the IEEE floating-point format; in DEC mode (IEEE/$\overline{DEC}$ = LOW) it is interpreted using the DEC floating-point format.

The IEEE-to-DEC conversion operation (IEEE-TO-DEC) takes an IEEE-format floating-point number on port R and places the equivalent DEC-format floating-point number on port F. The operand on port S is not used in this operation; its value will not affect the operation in any way. The operation can be performed in either IEEE mode (IEEE/$\overline{DEC}$ = HIGH) or DEC mode (IEEE/$\overline{DEC}$ = LOW).

The DEC-to-IEEE conversion operation (DEC-TO-IEEE) takes a DEC-format floating-point number on port R and places the equivalent IEEE-floating-point number on port F. The operand on port S is not used in this operation; its value will not affect the operation in any way. The operation can be performed in either IEEE mode (IEEE/$\overline{DEC}$ = HIGH) or DEC mode (IEEE/$\overline{DEC}$ = LOW).

Status Flag Generator

The status flag generator controls the state of six flags that report the status of floating-point ALU operations. The flags indicate when an operation is invalid (e.g., ∞ times 0) or when an operation has produced an overflow, an underflow, a non-numerical result (e.g., a NAN- or DEC-reserved operand), an inexact result, or a result of zero. The flags represent the status of the most recently performed operation. Flag status is stored in the flag status register on the LOW-to-HIGH transition of CLK. When the output register feedthrough control FT_1 is HIGH, the flag status register is made transparent.

Data Path

The 32-bit data path consists of the R and S input buses; the F output bus; data registers R, S, and F; the register R input multiplexer; and the ALU port S input multiplexer.

Input operands enter the floating-point processor through the 32-bit R and S input buses, R_0–R_{31} and S_0–S_{31}. Results of operations appear on the 32-bit F bus, F_0–F_{31}. The F bus assumes a high-impedance state when output enable $\overline{OE}$ is HIGH.

The R and S registers store input operands; the F register stores the final result of the floating-point ALU operation. Each register has an independent clock enable ($\overline{ENR}$, $\overline{ENS}$, and $\overline{ENF}$). When a register's clock enable is LOW, the register stores the data on its input at the LOW-to-HIGH transition of CLK; when the clock enable is HIGH, the register retains its current data. All data registers are fully edge-triggered — both the input data and the register enable need only meet modest setup and hold time requirements. Registers R and S can be made transparent by setting FT_0, the input register feedthrough control, HIGH. Register F can be made transparent by setting FT_1, the output register feedthrough control, HIGH.

The register R input multiplexer selects either the R input bus or the floating-point ALU's F port as the input to register R. Selection is controlled by I_4 — a LOW selects the R input bus; a HIGH selects the ALU F port. The ALU port S input multiplexer selects either register S or register F as the input to the floating-point ALU's S port. Selection is controlled by I_3 — a LOW selects register S; a HIGH selects register F.

Data selected by I_3 and I_4 is described in Table 2. When registers R and S are transparent (FT_0 = HIGH), multiplexer select I_4 must be kept LOW, so that the register R input multiplexer selects R_0–R_{31}. When register F is transparent (FT_1 = HIGH), multiplexer select I_3 must be kept LOW, so that the ALU port S input multiplexer selects register S.

TABLE 2. MUX SELECT

I_3	Data selected for floating-point ALU S port
0	Register S
1	Register F
I_4	**Data selected for register R input**
0	R bus
1	Floating-point ALU port F

TABLE 3. I/O MODE SELECTION

S16/$\overline{32}$	ONEBUS	I/O Mode
0	0	32-bit, two-input-bus mode
0	1	32-bit, single-input-bus mode*
1	0	16-bit, two-input-bus mode*
1	1	Illegal I/O mode selection value

*FT_0 must be held LOW in this mode (see text).

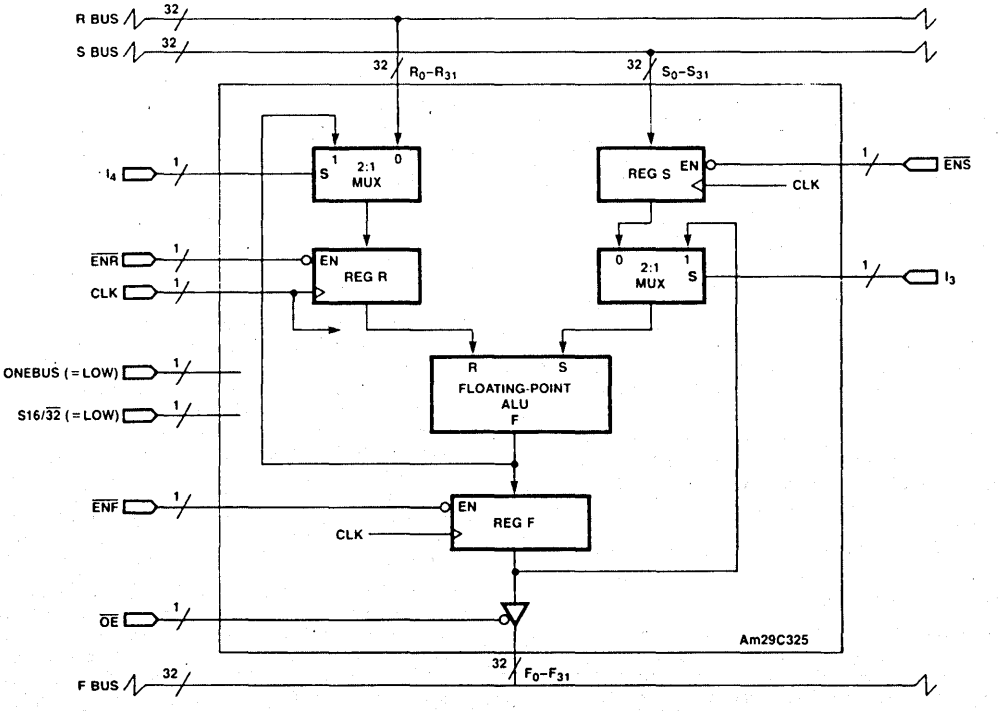
I/O Modes

The Am29C325 datapath can be configured in one of three I/O modes: a 32-bit, two-input bus mode; a 32-bit, single-input bus mode; and a 16-bit, two-input bus mode. These modes affect only the manner in which data is delivered to and taken from the Am29C325; operation of the floating-point ALU is not altered. The I/O mode is selected with the ONEBUS and S16/$\overline{32}$ controls. Table 3 lists the control codes needed to invoke each I/O mode.

32-Bit, Two-Input Bus Mode

In this I/O mode, the R and S buses are configured as independent 32-bit input buses, and the F bus is configured as a 32-bit output bus. Figure 2 is a functional block diagram of the Am29C325 in this I/O mode.

R and S operands are taken from their respective input buses and clocked into the R and S registers on the LOW-to-HIGH transition of CLK. Register F is also clocked on the LOW-to-HIGH transition of CLK. Figure 5a depicts typical I/O timing in this mode.



BD007051

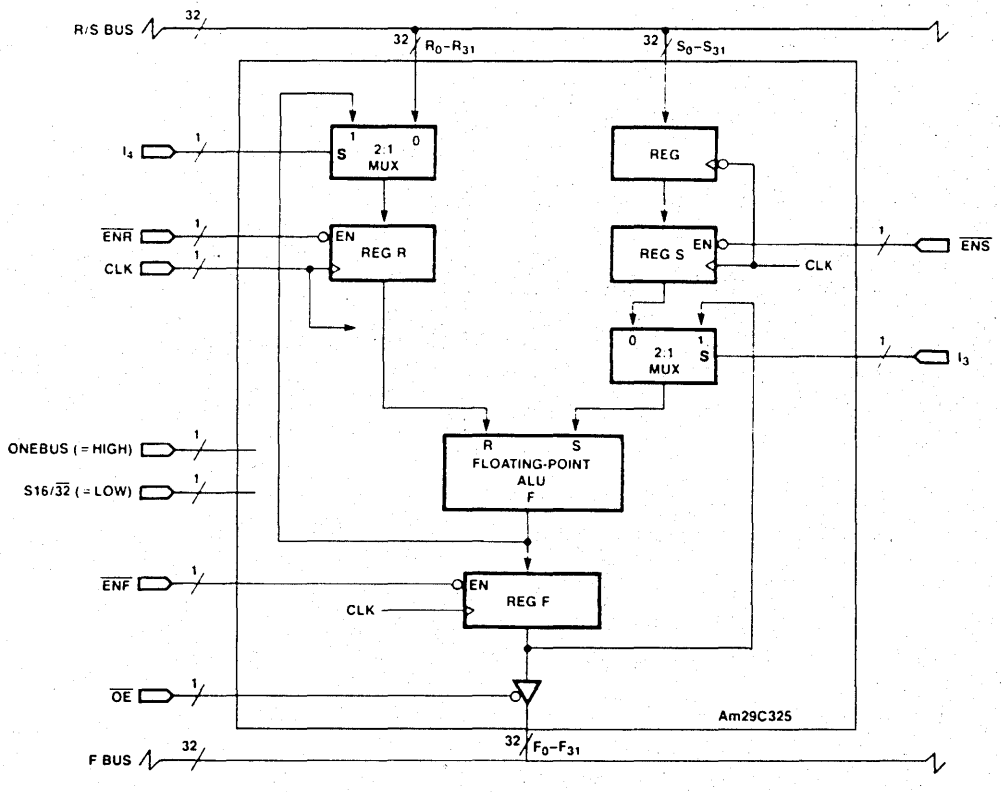
Figure 2. Functional Block Diagram for the 32-Bit, Two-Input Bus Mode

32-Bit, Single-Input Bus Mode

In this I/O mode, the R and S buses are connected to a single 32-bit multiplexed input data bus; the F bus is configured as an independent 32-bit output bus. Figure 3 is a functional block diagram of the Am29C325 in this I/O mode. Note that both the R and S bus lines must be wired to the input bus.

R and S operands are multiplexed onto the input bus by the host system. The S operand is clocked from the input bus into a temporary holding register on the HIGH-to-LOW transition of CLK and is transferred to register S on the LOW-to-HIGH transition of CLK. The R operand is clocked from the input bus into register R on the LOW-to-HIGH transition of CLK. Register F is clocked on the LOW-to-HIGH transition of CLK. Figure 5b depicts typical I/O timing in this mode.

When placed in this I/O mode, the data path will not function properly if the R and S registers are made transparent. Therefore, input register feedthrough control FT_0 must be held LOW in this mode.



BD007062

Figure 3. Functional Block Diagram for the 32-Bit, Single-Input Bus Mode

16-Bit, Two-Input Bus Mode

In this I/O mode, the R and S buses are configured as independent 16-bit input buses, and the F bus is configured as a 16-bit output bus. Figure 4 is a functional block diagram of the Am29C325 in this I/O mode. Note that the 16 least significant bits (LSBs) and 16 most significant bits (MSBs) of the R, S, and F buses must be wired to their respective system buses in parallel.

Thirty-two-bit operands are passed along the 16-bit data buses by time-multiplexing the 16 LSBs and 16 MSBs of each 32-bit word. For the R input bus, the host system multiplexes the 16 LSBs and 16 MSBs of the R operand onto the 16-bit R bus. The 16 LSBs of the R operand are stored in a temporary holding register on the HIGH-to-LOW transition of CLK. The 16 MSBs are clocked into register R on the LOW-to-HIGH transition of CLK; at the same time, the 16 LSBs are transferred from the temporary holding register to register R. Transfer of data from the S input bus to the S register takes place in a similar fashion. Register F is clocked on the LOW-to-HIGH transition of CLK. Circuitry internal to the Am29C325 multiplexes data from register F onto the 16-bit output bus by enabling the 16 LSBs of the F output bus when CLK is HIGH and enabling the 16 MSBs of the F output bus when CLK is LOW. Figure 5c depicts typical I/O timing in this mode.

When placed in this I/O mode, the data path will not function properly if the R and S registers are made transparent. Therefore, input register feedthrough control FT_0 must be held LOW in this mode. Caution must also be taken in controlling the register R input multiplexer control line, I_4, in this I/O mode. I_4 should be changed only when CLK is HIGH, in addition to meeting the setup and hold time requirements given in the **Switching Characteristics** section.

Operation in IEEE Mode

When input signal IEEE/$\overline{\text{DEC}}$ is HIGH, the IEEE mode of operation is selected. In this mode the Am29C325 uses the floating-point format set forth in the IEEE Proposed Standard for Binary Floating-Point Arithmetic, P754. In addition, the IEEE mode complies with most other aspects of single-precision floating-point operation outlined in the proposed standard — differences are discussed in **Appendix A.**

IEEE Floating-Point Format

The IEEE single-precision floating-point word is 32 bits wide and is arranged in the format shown in Figure 6. The floating-point word is divided into three fields: a single-bit sign, an 8-bit biased exponent, and a 23-bit fraction.

The sign bit indicates the sign of the floating-point number's value. Non-negative values have a sign of 0; negative values, a sign of 1. The value zero may have either sign.

The biased exponent is an 8-bit unsigned integer field representing a multiplicative factor of some power of two. The bias value is 127. If, for example, the multiplicative factor for a floating-point number is to be 2^a, the value of the biased exponent would be a +127; "a" is called the true exponent.

The fraction is a 23-bit unsigned fraction field containing the 23 LSBs of the floating-point number's 24-bit mantissa. The weight of fraction's MSB is 2^{-1}; the weight of the LSB is 2^{-23}.

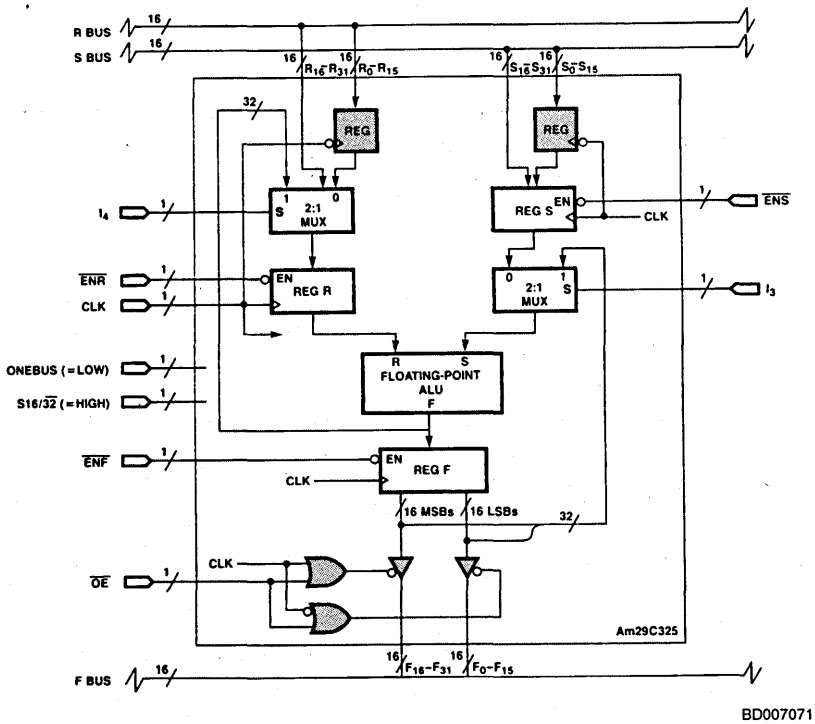


Figure 4. Functional Block Diagram for the 16-Bit, Two-Input Bus Mode

A floating-point number is evaluated or interpreted per the following conventions:

let s = sign bit
e = biased exponent
f = fraction

if e = 0 and f = 0...value = $(-1)^{s}*(0)$ (+ 0, −0)
if e = 0 and f ≠ 0...value = denormalized number
if 0 < e < 255...value = $(-1)^{s}*(2^{e-127})*(1.f)$
(normalized number)
if e = 255 and f = 0...value = $(-1)^{s}*(\infty)$ (+ ∞, −∞)
if e = 255 and f ≠ 0...value = not-a-number (NAN)

Zero: The value zero can have either a positive or negative sign. Rules for determining the sign of a zero produced by an operation are given in the **Sign Bit** section.

Denormalized Number: A denormalized number represents a quantity with magnitude less than 2^{-126} but greater than zero.

Normalized Number: A normalized number represents a quantity with magnitude greater than or equal to 2^{-126} but less than 2^{128}.

Example 1:

The number + 3.5 can be represented in floating-point format as follows:

$$+3.5 = 11.1_2 \times 2^0 = 1.11_2 \times 2^1$$

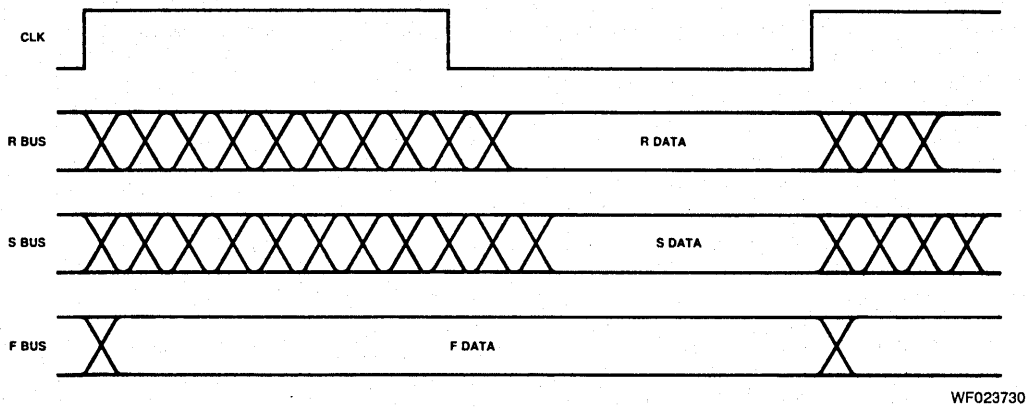
sign = 0

biased exponent = $1_{10} + 127_{10} = 128_{10} = 10000000_2$

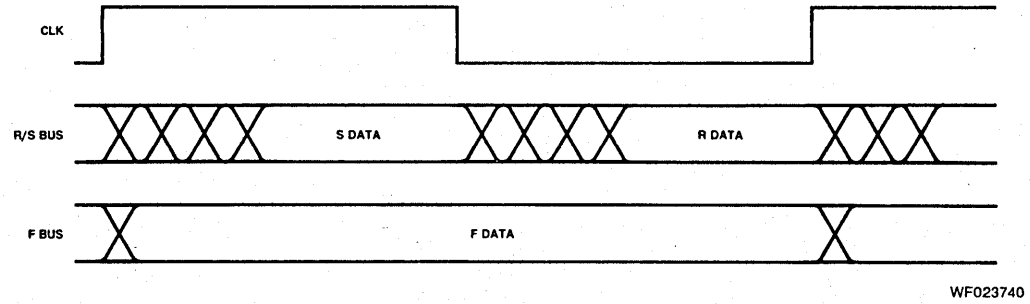
fraction = $11000000000000000000000_2$
(the leading 1 is implied in the format)

Concatenating these fields produces the floating-point word 40600000_{16}.

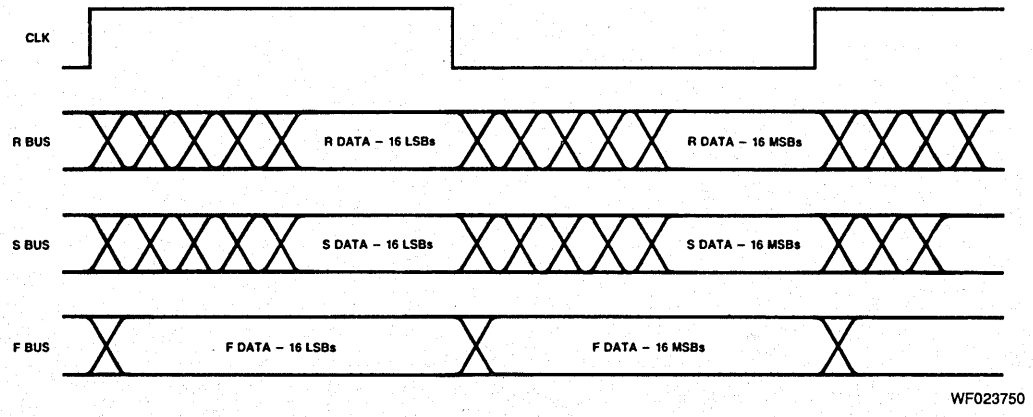
2



a) 32-Bit, Two-Input-Bus Mode

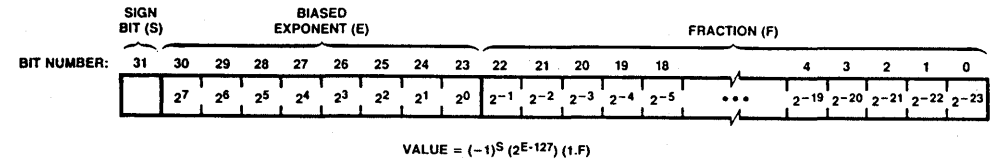


b) 32-Bit, Single-Input-Bus Mode



c) 16-Bit, Two-Input-Bus Mode

Figure 5. Typical Bus Timing for the I/O Modes with FT_0 = LOW, FT_1 = LOW



TB000640

Figure 6. IEEE Mode Single-Precision Floating-Point Format

Example 2:

The number −11.375 can be represented in floating-point format as follows:

$$-11.375 = -1011.011_2 \times 2^0$$
$$= -1.011011_2 \times 2^3$$

sign = 1

biased exponent = $3_{10} + 127_{10} = 130_{10}$
$= 10000010_2$

fraction = $01101100000000000000000_2$
(the leading 1 is implied in the format)

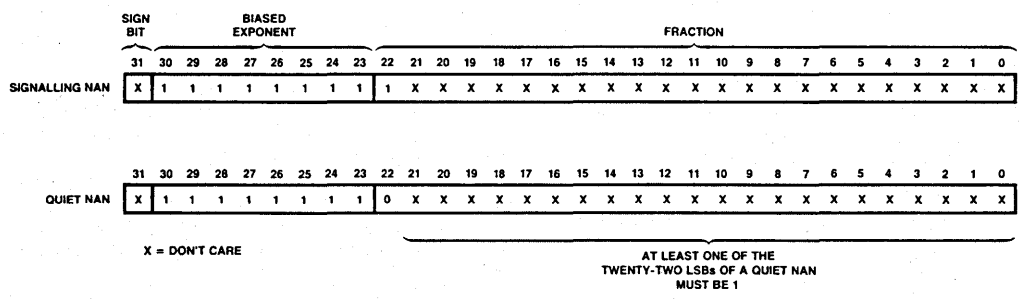
Concatenating these fields produces the floating-point word $C1360000_{16}$.

Infinity: Infinity can have either a positive or negative sign. The way in which infinities are interpreted is determined by the state of the projective/affine mode select, PROJ/$\overline{\text{AFF}}$.

Not-a-Number: A not-a-number, or NAN, does not represent a numeric value but is interpreted as a signal or symbol. NANs are used to indicate invalid operations and as a means of passing process status information through a series of calculations. NANs arise in two ways: 1) they can be generated by the Am29C325 to indicate that an invalid operation has taken place (e.g., ∞ x 0), or 2) be provided by the user as an input operand. There are two types of NANs, signalling and quiet (see Figure 7 for formats).

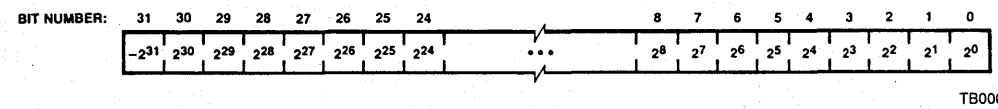
IEEE Mode Integer Format

Integer numbers are represented as 32-bit, two's complement words (Figure 8 depicts the integer format). The integer word can represent a range of integer values from -2^{31} to $2^{31} - 1$.



TB000650

Figure 7. Signalling and Quiet NAN Formats



TB000660

Figure 8. 32-Bit Integer Format

Operations

All eight floating-point ALU operations discussed in the **Functional Description** section can be performed in IEEE mode. Various exceptional aspects of the R PLUS S, R MINUS S, R TIMES S, 2 MINUS S, INT-TO-FP, and FP-TO-INT operations for this mode are described below. The IEEE-TO-DEC and DEC-TO-IEEE operations are discussed separately in the **IEEE-TO-DEC AND DEC-TO-IEEE Operations** section.

Operations with NANs: NANs arise in two ways: 1) they can be generated by the Am29C325 to indicate that an invalid operation has taken place (e.g., ∞ x 0), or 2) be provided by the user as an input operand. There are two types of NANs, signalling and quiet (see Figure 7 for formats).

Signalling NANs set the invalid operation flag when they appear as an input operand to an operation. They are useful for indicating uninitialized variables, or for implementing user-

designed extensions to the operations provided. The ALU never produces a signalling NAN as the final result of an operation.

Quiet NANs are generated for invalid operations. When they appear as an input operand, they are passed through most operations without setting the invalid flag, the floating-point-to-integer conversion operation being the exception.

The sign of any input operand NAN is ignored. All quiet NANs produced as the final result of an operation have a sign of 0.

When a NAN appears as an input operand, the final result of the operation is a quiet NAN that is created by taking the input NAN and forcing bit 22 LOW and bit 21 HIGH. If an operation has two NANs as input operands, the resulting quiet NAN is created using the NAN on the R port.

When a quiet NAN is produced as the final result of an invalid operation whose input operand or operands are not NANs, the resulting NAN will always have the value $7FA00000_{16}$.

The NAN flag will be HIGH whenever an operation produces a NAN as a final result.

Example 1:

Suppose the floating-point addition operation is performed with the following input operands:

R port: $3F800000_{16}$ $(1.0*2^0)$
S port: $7FC12345_{16}$ (signalling NAN)

Result: The signalling NAN on the S port is converted to a quiet NAN by forcing bit 22 LOW and bit 21 HIGH. The operation's final result will be $7FA12345_{16}$. Since one of the two input operands is a signalling NAN, the invalid flag will be HIGH; the NAN flag will also be HIGH.

Example 2:

Suppose the floating-point multiplication operation is performed with the following input operands:

R port: $FFF11111_{16}$ (signalling NAN)
S port: $7FC22222_{16}$ (quiet NAN)

Result: Since both input operands are NANs, the NAN on the R port is chosen for output. In addition to forcing bit 22 LOW, the sign bit (bit 31) is set LOW (bit 21 is already HIGH, and need not be changed). The operation's final result will be $7FB11111_{16}$. Since one of the two input operands is a signalling NAN, the invalid flag is HIGH; the NAN flag will also be HIGH.

Example 3:

Suppose the floating-point subtraction operation is performed with the following input operands:

R port: $FF800001_{16}$ (quiet NAN)
S port: $7F800000_{16}$ $(+\infty)$

Result: To create the final result, the quiet NANs sign bit (bit 31) is forced LOW and bit 21 is forced HIGH (bit 22 is already LOW, and need not be changed). The final result will be $7FA00001_{16}$. The NAN flag will be HIGH.

Operations with Denormalized Numbers: The proposed IEEE standard incorporates denormalized numbers to allow a means of gradual underflow for operations that produce non-zero results too small to be expressed as a normalized floating-point number. The Am29C325 does not support gradual underflow. If a floating-point operation produces a non-zero rounded result that is not large enough to be expressed as a normalized floating-point number, the final result will be a zero of the same sign; the inexact, underflow, and zero flags will be HIGH. If an input operand is a denormalized number, the floating-point ALU will assume that operand to be a zero of the same sign.

Operations Producing Overflows: If an operation has a finite input operand or operands and if the operation produces a rounded result that is too large to fit in the destination format, the operation is said to have overflowed.

A floating-point overflow occurs if an R PLUS S, R MINUS S, R TIMES S, or 2 MINUS S operation with finite input operand(s) produces a result which, after rounding, has a magnitude greater than or equal to 2^{128}. Positive or negative infinity will appear as the final result if the rounded result is positive or negative, respectively, and the overflow and inexact flags will be HIGH.

Integer overflow occurs when the floating-point-to-integer conversion operation attempts to convert a number which, after rounding, is greater than $2^{31} - 1$ or less than -2^{31}. The final result will be quiet NAN $7FA00000_{16}$, and the invalid operation and NAN flags will be HIGH. Note that the overflow and inexact flags remain LOW for integer overflow.

Operations Producing Underflows: If an operation produces a floating-point rounded result having a magnitude too small to be expressed as a normalized floating-point number but greater than zero, that operation is said to have underflowed. Underflow occurs when an R PLUS S, R MINUS S, or R TIMES S operation produces a result which, after rounding, has a magnitude in the range:

$$0 < \text{magnitude} < 2^{-126}.$$

In such cases, the final result will be +0 (00000000_{16}) if the rounded result is non-negative and −0 (80000000_{16}) if the rounded result is negative. The underflow, inexact, and zero flags will be HIGH.

Underflow does not occur if the destination format is integer. If the infinitely precise result of a floating-point-to-integer conversion has a magnitude greater than 0 and less than 1 but the rounded result is 0, the underflow flag remains LOW.

Operations with Infinities: In most cases, positive and negative infinity are valid inputs for the R PLUS S, R MINUS S, R TIMES S, and 2 MINUS S operations. Those cases for which infinities are not valid inputs for these operations are listed in Table 4.

Infinities in IEEE mode can be handled either as projective or affine. The projective mode is selected when PROJ/$\overline{\text{AFF}}$ is HIGH; the affine mode is selected when PROJ/$\overline{\text{AFF}}$ is LOW. The only differences between the modes that are relevant to Am29C325 operation occur during the addition and subtraction of infinities:

Operation	Affine Mode	Projective Mode
$(+\infty) + (+\infty)$	Output $+\infty$	Output $7FA00000_{16}$ (quiet NAN), set invalid and NAN flags
$(-\infty) + (-\infty)$	Output $-\infty$	Output $7FA00000_{16}$ (quiet NAN), set invalid and NAN flags
$(+\infty) - (-\infty)$	Output $+\infty$	Output $7FA00000_{16}$ (quiet NAN), set invalid and NAN flags
$(-\infty) - (+\infty)$	Output $-\infty$	Output $7FA00000_{16}$ (quiet NAN), set invalid and NAN flags

If an R PLUS S, R MINUS S, or 2 MINUS S operation has infinity as an input operand or operands, the final result, if valid, is presumed to be exact. For example, adding $+\infty$ and 2.0 will produce a final result of $+\infty$; since the result is considered exact, the inexact flag remains LOW.

Invalid Operations: If an input operand is invalid for the operation to be performed, that operation is considered invalid. When an invalid operation is performed, the floating-point ALU produces a quiet NAN as the final result and the invalid operation flag goes HIGH. Table 4 lists the cases for which the invalid flag is HIGH in IEEE mode and the final results produced for these operations.

TABLE 4. IEEE MODE INVALID OPERATIONS

Operation	Input Operand	Final Result
R PLUS S	$(+\infty)+(-\infty)$ or $(-\infty)+(+\infty)$	$7FA00000_{16}$ (quiet NAN)
R PLUS S	$(+\infty)+(+\infty)$ or $(-\infty)+(-\infty)$ (Note 1)	$7FA00000_{16}$ (quiet NAN)
R MINUS S	$(+\infty)-(+\infty)$ or $(-\infty)-(-\infty)$	$7FA00000_{16}$ (quiet NAN)
R MINUS S	$(+\infty)-(-\infty)$ or $(-\infty)-(+\infty)$ (Note 1)	$7FA00000_{16}$ (quiet NAN)
R TIMES S	$(+0)*(+\infty)$ or $(+0)*(-\infty)$ or $(-0)*(+\infty)$ or $(-0)*(-\infty)$	$7FA00000_{16}$ (quiet NAN)
R PLUS S R MINUS S R TIMES S	R or S is a signalling NAN	(Note 2)
2 MINUS S	S is a signalling NAN	(Note 2)
FP-TO-INT	R is a signalling or quiet NAN	(Note 2)
FP-TO-INT	$R > 2^{31}-1$ or $R < -(2^{31})$	$7FA00000_{16}$ (quiet NAN)

Notes: 1. These cases are invalid in projective mode only.
2. Results for these operations are described in the **Operations with NANs** section.

The Sign Bit

For most floating-point operations, the sign bit of the final result is unambiguous; i.e., there is only one sign bit value that yields a numerically correct result. Operations that produce an infinitely precise result of zero, however, present a problem, as the IEEE floating-point format allows for representation of both +0 and −0. The following rules can be used to determine the signs of zero produced in such cases.

R PLUS S: The operations $+x+(-x)$ and $-x+(+x)$ produce a final result of zero; the sign of the zero is dependent on the rounding mode:

Rounding Mode	Sign of Final Result
Round to nearest	0
Round toward $-\infty$	1
Round toward $+\infty$	0
Round toward 0	0

Operations $+0+(-0)$ and $-0+(+0)$ produce a result of 0, with the sign of the result determined by the table above.

The operation $+0+(+0)$ produces a final result of +0; the operation $-0+(-0)$ produces a final result of −0.

R MINUS S: The operations $+x-(+x)$ and $-x-(-x)$ produce a final result of zero; the sign of the zero is dependent on the rounding mode:

Rounding Mode	Sign of Result
Round to nearest	0
Round toward $-\infty$	1
Round toward $+\infty$	0
Round toward 0	0

Operations $+0-(+0)$ and $-0-(-0)$ produce a result of 0, with the sign of the result determined by the table above.

The operation $+0-(-0)$ produces a final result of +0; the operation $-0-(+0)$ produces a final result of −0.

R TIMES S: The sign of any multiplication result other than a NAN is the exclusive OR of the signs of the input operands. Therefore, if x is non-negative,
+0 times +x produces a final result of +0,
+0 times −x produces a final result of −0,
−0 times +x produces a final result of −0,
−0 times −x produces a final result of +0.

2 MINUS S: If S equals 2, the final result is −0 for the round toward $-\infty$ mode and +0 for all other rounding modes.

Rounding

Rounding is performed whenever an operation produces an infinitely precise result that cannot be represented exactly in the destination format. For example, suppose a floating-point operation produces the infinitely precise result:

$$1.10101010101010101010101\backslash 01 \times 2^{3}.$$

In this example, the fraction portion of the mantissa has 25 bits; the IEEE floating-point format can accommodate only 23. The backslash (\) in the mantissa represents the boundary between the first 23 bits of the fraction and any remaining bits. Rounding is the process by which this result is approximated by a representation that fits the destination format.

There are four rounding modes in IEEE mode: 1) round to nearest, 2) round toward $+\infty$, 3) round toward $-\infty$, and 4) round toward 0. The rounding mode is chosen using the rounding mode select lines, RND_0 and RND_1. Table 5 lists the select states needed to obtain the desired rounding mode.

TABLE 5. ROUNDING MODE SELECT

RND_1	RND_0	Rounding Mode
0	0	Round to nearest
0	1	Round toward $-\infty$
1	0	Round toward $+\infty$
1	1	Round toward 0

Round to Nearest: In this rounding mode the infinitely precise result of an operation is rounded to the closest representation that fits in the destination format. If the infinitely precise result is exactly halfway between two representations, it is rounded to the representation having an LSB of zero. Rounding is performed both for floating-point and integer destination formats.

Figure 9 illustrates four examples of the round-to-nearest process for operations having a floating-point destination format. The infinitely precise result of an operation is represented by an "X" on the number line; the black dots on the number line indicate those values that can be represented exactly in the floating-point format.

Example 1:

In Figure 9(a), the infinitely precise result of an operation is:

$2^{20} + 2^{-4} + 2^{-5} = 1.00000000000000000000000\backslash 11 \times 2^{20}$

The result is rounded to the closest representable floating-point value,

$2^{20} + 2^{-3} = 1.00000000000000000000001 \times 2^{20}$

Example 2:

In Figure 9(b), the infinitely precise result of an operation is:

$2^{20} - 2^{-4} + 2^{-8} =$
$1.11111111111111111111111\backslash 0001 \times 2^{19}$

This result is rounded to the closest representable floating-point value,

$2^{20} - 2^{-4} = 1.11111111111111111111111 \times 2^{19}$

Example 3:

In Figure 9(c), the infinitely precise result of an operation is:

$-(2^{20} + 2^{-3} + 2^{-4})$
$= -1.00000000000000000000001\backslash 1 \times 2^{20}$

This result is exactly halfway between two representable floating-point values. Accordingly, it is rounded to the closest representation with an LSB of zero, or

$-(2^{20} + 2*2^{-3}) = -1.00000000000000000000010 \times 2^{20}$

Example 4:

In Figure 9(d), the infinitely precise result of an operation is:

$2^{20} + 3*2^{-3} = 1.00000000000000000000011 \times 2^{20}$

This result can be represented exactly in the floating-point format, and is left unaltered by the rounding process.

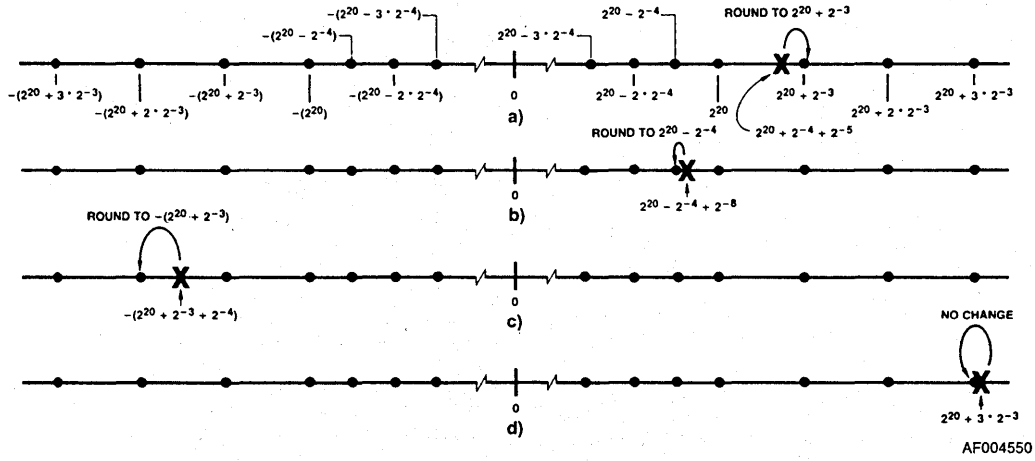


Figure 9. Floating-Point Rounding Examples for Round-to-Nearest Mode

Figure 10 illustrates four examples of the round-to-nearest process for operations having an integer destination format. The infinitely precise result of an operation is represented by an "X" on the number line; the black dots on the number line indicate those values that can be represented exactly in the integer format.

Example 1:

In Figure 10(a), the infinitely precise result of an operation is:

$2^{10} - 2^{-2}$ = 00...001111111111.11

The result is rounded to the closest representable integer value,

2^{10} = 00...010000000000

Example 2:

In Figure 10(b), the infinitely precise result of an operation is:

$2^{10} + 2^{0} + 2^{-3}$ = 00...010000000001.001

This result is rounded to the closest representable integer value,

$2^{10} + 2^{0}$ = 00...010000000001

Example 3:

In Figure 10(c), the infinitely precise result of an operation is:

$-(2^{10} + 2^{0} + 2^{-1})$ = −11...101111111110.1

This result is exactly halfway between two representable integer values. Accordingly, it is rounded to the closest representation with an LSB of zero, or

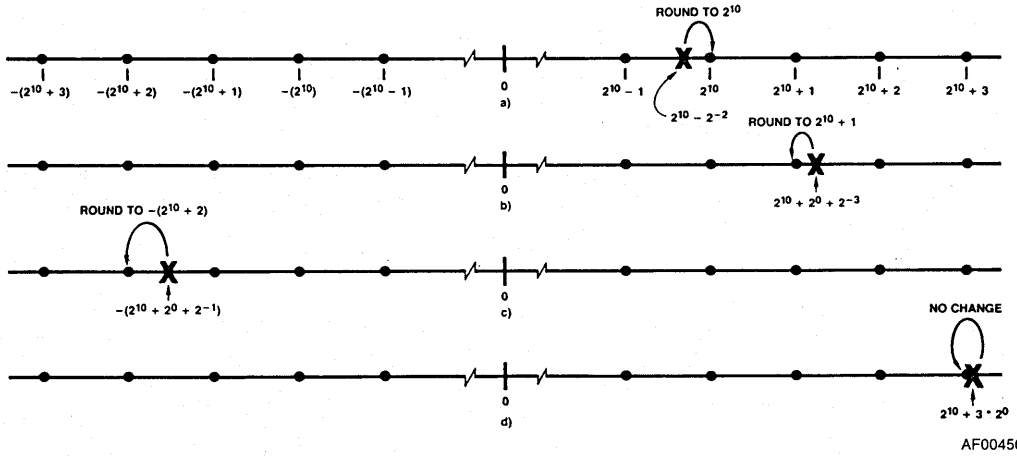
$-(2^{10} + 2*2^{0})$ = 11...101111111110

Example 4:

In Figure 10(d), the infinitely precise result of an operation is:

$2^{10} + 3*2^{0}$ = 00...010000000011

This result can be represented exactly in the integer format and is left unaltered by the rounding process.



AF004560

Figure 10. Integer Rounding Examples for Round-to-Nearest Mode

Round Toward $-\infty$: In this rounding mode the result of an operation is rounded to the closest representation that is less than or equal to the infinitely precise result and which fits the destination format. Rounding is performed both for floating-point and integer destination formats.

Figure 11 illustrates four examples of the round toward $-\infty$ process for operations having a floating-point destination format. The infinitely precise result of an operation is represented by an "X" on the number line; the black dots on the number line indicate those values that can be represented exactly in the floating-point format.

Example 1:

In Figure 11(a), the infinitely precise result of an operation is:

$2^{20} + 2^{-4} + 2^{-5} = 1.00000000000000000000000\backslash 11 \times 2^{20}$

This result cannot be represented exactly in floating-point format and is rounded to the next-smaller floating-point representation:

$2^{20} = 1.00000000000000000000000 \times 2^{20}$

Example 2:

In Figure 11(b), the infinitely precise result of an operation is:

$2^{20} - 2^{-4} + 2^{-8} =$
$1.11111111111111111111111\backslash 0001 \times 2^{19}$

This result cannot be represented exactly in floating-point format and is rounded to the next-smaller floating point representation:

$2^{20} - 2^{-4} = 1.11111111111111111111111 \times 2^{19}$

Example 3:

In Figure 11(c), the infinitely precise result of an operation is:

$-(2^{20} + 2^{-3} + 2^{-4}) =$
$-1.00000000000000000000001\backslash 1 \times 2^{20}$

This result cannot be represented exactly in floating-point format and is rounded to the next-smaller floating-point representation.

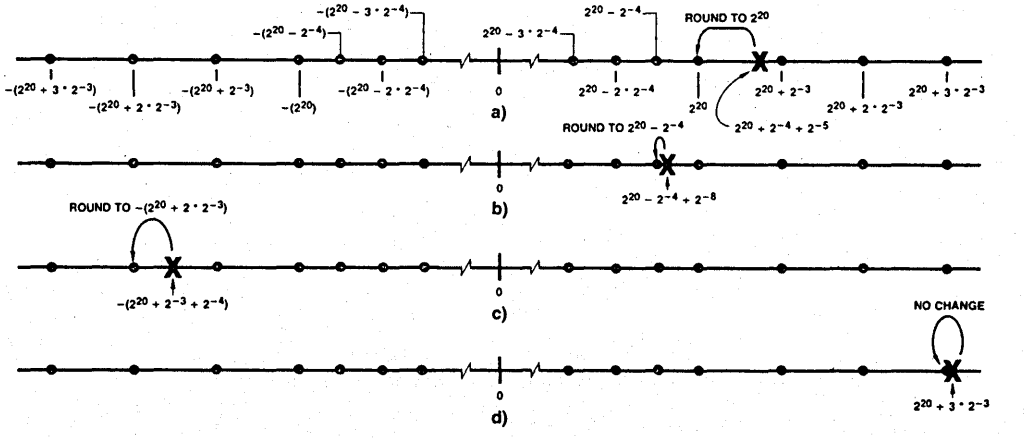
$-(2^{20} + 2*2^{-3}) = -1.00000000000000000000010 \times 2^{20}$

Example 4:

In Figure 11(d), the infinitely precise result of an operation is:

$2^{20} + 3*2^{-3} = 1.00000000000000000000011 \times 2^{20}$

This result can be represented exactly in the floating-point format and is left unaltered by the rounding process.



AF004510

Figure 11. Floating-Point Rounding Examples for Round Toward $-\infty$ Mode

Figure 12 illustrates four examples of the round toward $-\infty$ process for operations having an integer destination format. The infinitely precise result of an operation is represented by an "X" on the number line; the black dots on the number line indicate those values that can be exactly represented in the integer format.

Example 1:

In Figure 12(a), the infinitely precise result of an operation is:

$2^{10} - 2^{-2}$ = 00...001111111111.11

The result is rounded to the next-smaller representable integer value,

$2^{10} - 2^{0}$ = 00...001111111111

Example 2:

In Figure 12(b), the infinitely precise result of an operation is:

$2^{10} + 2^{0} + 2^{-3}$ = 00...010000000001.001

This result is rounded to the next-smaller representable integer value,

$2^{10} + 2^{0}$ = 00...010000000001

Example 3:

In Figure 12(c), the infinitely precise result of an operation is:

$-(2^{10} + 2^{0} + 2^{-1})$ = 11...101111111110.1

This result is rounded to the next-smaller representable integer value:

$-(2^{10} + 2*2^{0})$ = 11...101111111110

Example 4:

In Figure 12(d), the infinitely precise result of an operation is:

$2^{10} + 3*2^{0}$ = 00...010000000011

This result can be represented exactly in the integer format and is unaltered by the rounding process.

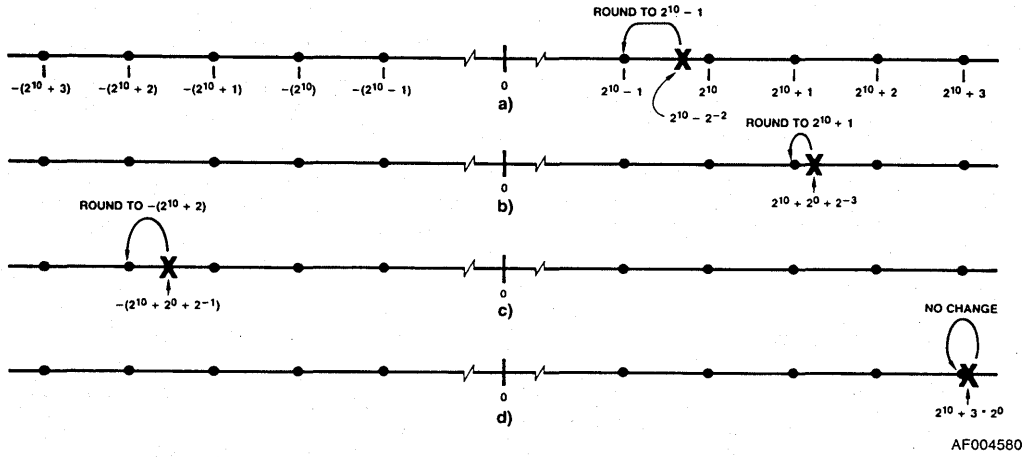


Figure 12. Integer Rounding Examples for Round Toward $-\infty$ Mode

Round Toward +∞: In this rounding mode the result of an operation is rounded to the closest representation that is greater than or equal to the infinitely precise result and which fits the destination format. Rounding is performed both for floating-point and integer destination formats.

Figure 13 illustrates four examples of the round toward +∞ process for operations having a floating-point destination format. The infinitely precise result of an operation is represented by an "X" on the number line; the black dots on the number line indicate those values that can be represented exactly in the floating-point format.

Example 1:

In Figure 13(a), the infinitely precise result of an operation is:

$2^{20} + 2^{-4} + 2^{-5} = 1.00000000000000000000000\backslash 11 \times 2^{20}$

This result cannot be represented exactly in floating-point format and is rounded to the next-larger floating-point representation:

$2^{20} + 2^{-3} = 1.00000000000000000000001 \times 2^{20}$

Example 2:

In Figure 13(b), the infinitely precise result of an operation is:

$2^{20} - 2^{-4} + 2^{-8} =$
$1.11111111111111111111111\backslash 0001 \times 2^{19}$

This result cannot be represented exactly in floating-point format and is rounded to the next-larger floating point representation:

$2^{20} = 1.00000000000000000000000 \times 2^{20}$

Example 3:

In Figure 13(c), the infinitely precise result of an operation is:

$-(2^{20} + 2^{-3} + 2^{-4}) =$
$-1.00000000000000000000001\backslash 1 \times 2^{20}$

This result cannot be represented exactly in floating-point format and is rounded to the next-larger floating-point representation.

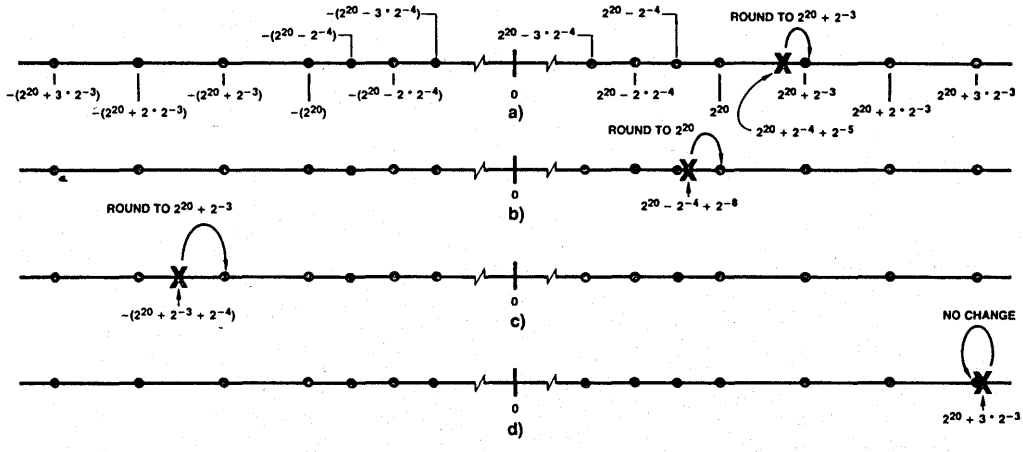
$-(2^{20} + 2^{-3}) = -1.00000000000000000000001 \times 2^{20}$

Example 4:

In Figure 13(d), the infinitely precise result of an operation is:

$2^{20} + 3*2^{-3} = 1.00000000000000000000011 \times 2^{20}$.

This result can be represented exactly in the floating-point format — no rounding takes place.



AF004590

Figure 13. Floating-Point Rounding Examples for Round Toward +∞ Mode

Figure 14 illustrates four examples of the round toward $+\infty$ process for having an integer destination format. The infinitely precise result of an operation is represented by an "X" on the number line; the black dots on the number line indicate those values that can be exactly represented in the integer format.

Example 1:

In Figure 14(a), the infinitely precise result of an operation is:

$2^{10} - 2^{-2} = 00...001111111111.11$

The result is rounded to the next-larger representable integer value,

$2^{10} = 00...010000000000$

Example 2:

In Figure 14(b), the infinitely precise result of an operation is:

$2^{10} + 2^{0} + 2^{-3} = 00...010000000001.001$

This result is rounded to the next-larger representable integer value,

$2^{10} + 2*2^{0} = 00...010000000010$

Example 3:

In Figure 14(c), the infinitely precise result of an operation is:

$-(2^{10} + 2^{0} + 2^{-1}) = 11.101111111110.1$

This result is rounded to the next-larger representable integer value:

$-(2^{10} + 2^{0}) = 11...1011111111110$

Example 4:

In Figure 14(d), the infinitely precise result of an operation is:

$2^{10} + 3*2^{0} = 00...010000000011$

This result can be represented exactly in the integer format — no rounding takes place.

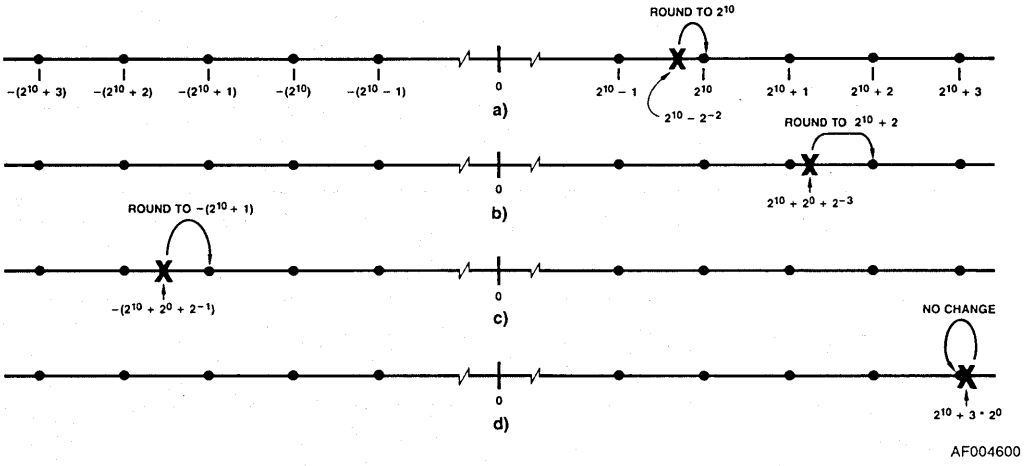


Figure 14. Integer Rounding Examples for Round Toward $+\infty$ Mode

Round Toward 0: In this rounding mode the result of an operation is rounded to the closest representation whose magnitude is less than or equal to the infinitely precise result and which fits the destination format. Rounding is performed both for floating-point and integer destination formats.

Figure 15 illustrates four examples of the round toward 0 process for operations having a floating-point destination format. The infinitely precise result of an operation is represented by an "X" on the number line; the black dots on the number line indicate those values that can be represented exactly in the floating-point format.

Example 1:

In Figure 15(a), the infinitely precise result of an operation is:

$2^{20} + 2^{-4} + 2^{-5} =$
$1.00000000000000000000000\backslash 11 \times 2^{20}$

This result cannot be represented exactly in floating-point format and is rounded to:

$2^{20} = 1.00000000000000000000000 \times 2^{20}$

Example 2:

In Figure 15(b), the infinitely precise result of an operation is:

$2^{20} - 2^{-4} + 2^{-8} =$
$1.11111111111111111111111\backslash 001 \times 2^{19}$

This result cannot be represented exactly in floating-point format and is rounded to:

$2^{20} - 2^{-4} = 1.11111111111111111111111 \times 2^{19}$

Example 3:

In Figure 15(c), the infinitely precise result of an operation is:

$-(2^{20} + 2^{-3} + 2^{-4}) =$
$-1.00000000000000000000001\backslash 1 \times 2^{20}$

This result cannot be represented exactly in floating-point format and is rounded to:

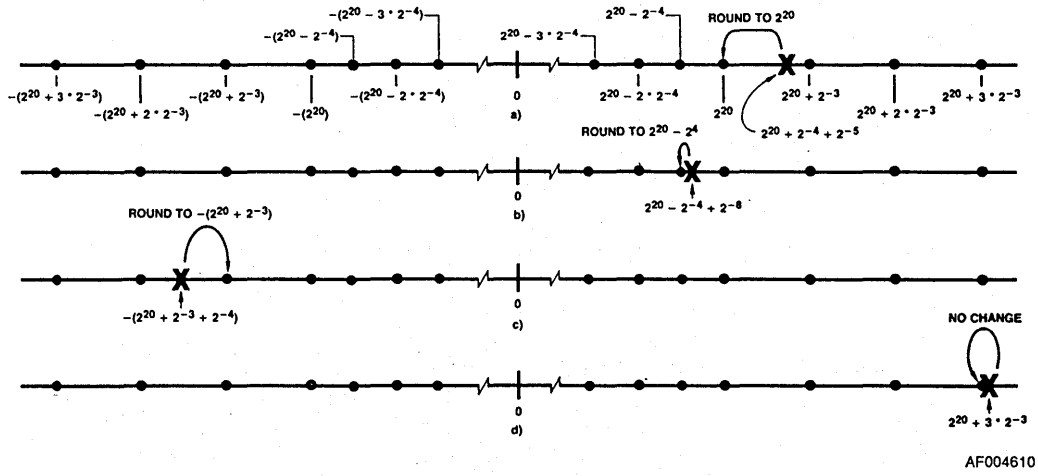
$-(2^{20} + 2^{-3}) = -1.00000000000000000000001 \times 2^{20}$

Example 4:

In Figure 15(d), the infinitely precise result of an operation is:

$2^{20} + 3*2^{-3} = 1.00000000000000000000011 \times 2^{20}$

This result can be represented exactly in the floating-point format and is unaffected by the rounding process.



AF004610

Figure 15. Floating-Point Rounding Examples for Round Toward 0 Mode

Figure 16 illustrates four examples of the round toward 0 process for operations having an integer destination format. The infinitely precise result of an operation is represented by an "X" on the number line; the black dots on the number line indicate those values that can be exactly represented in the integer format.

Example 1:

In Figure 16(a), the infinitely precise result of an operation is:

$2^{10} - 2^{-2} = 00...001111111111.11$

The result is rounded to:

$2^{10} - 2^{0} = 00...001111111111$

Example 2:

In Figure 16(b), the infinitely precise result of an operation is:

$2^{10} + 2^{0} + 2^{-3} = 00...010000000001.001$

The result is rounded to:

$2^{10} + 2^{0} = 00...010000000001$

Example 3:

In Figure 16(c), the infinitely precise result of an operation is:

$-(2^{10} + 2^{0} + 2^{-1}) = 11...101111111110.1$

The result is rounded to:

$-(2^{10} + 2^{0}) = 11...101111111111$

Example 4:

In Figure 16(d), the infinitely precise result of an operation is:

$2^{10} + 3*2^{0} = 00...010000000011$

This result can be represented exactly in the integer format and is unaffected by the rounding process.

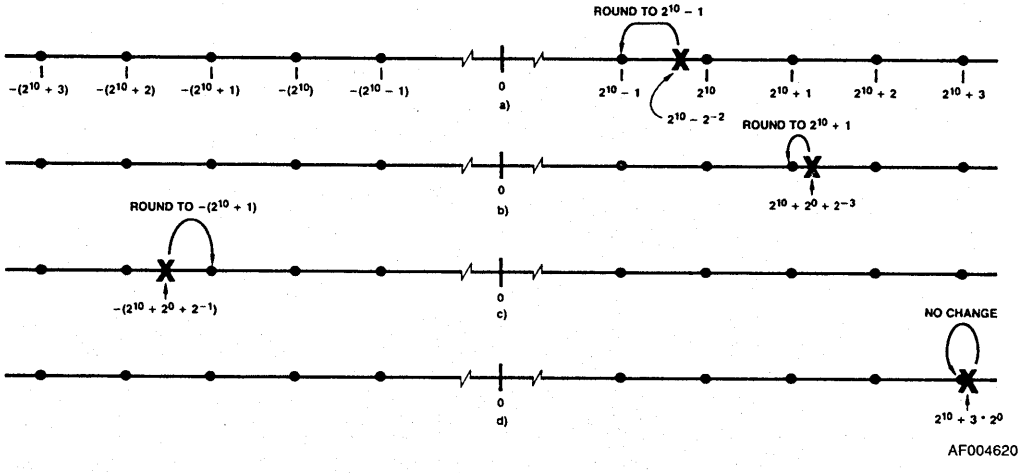


Figure 16. Integer Rounding Examples for Round Toward 0 Mode

Flag Operation

The Am29C325 generates six status flags to monitor floating-point processor operation. The following is a summary of flag conventions in IEEE mode:

Invalid Operation Flag: The invalid operation flag is HIGH when an input operand is invalid for the operation to be performed. Table 4 lists the cases for which the invalid operation flag is HIGH in IEEE mode and the corresponding final result. In cases where the invalid operation flag is HIGH, the overflow, underflow, zero, and inexact flags are LOW; the NAN flag will be HIGH.

Overflow Flag: The overflow flag is HIGH if an R PLUS S, R MINUS S, R TIMES S, or 2 MINUS S operation with finite input operand(s) produces a result which, after rounding, has a magnitude greater than or equal to 2^{128}. The final result will be $+\infty$ or $-\infty$.

Underflow Flag: The underflow flag is HIGH if an R PLUS S, R MINUS S, or R TIMES S operation produces a result which, after rounding, has a magnitude in the range:
$0 <$ magnitude $< 2^{-126}$.

The final result will be +0 (00000000_{16}) if the rounded result is non-negative and −0 (80000000_{16}) if the rounded result is negative.

Inexact Flag: The inexact flag is HIGH if the final result of an R PLUS S, R MINUS S, R TIMES S, 2 MINUS S, INT-TO-FP, or FP-TO-INT operation is not equal to the infinitely precise result. Note that if the underflow or overflow flag is HIGH, the inexact flag will also be HIGH.

Zero Flag: The zero flag is HIGH if the final result of an operation is zero. For operations producing an IEEE floating-point number, the flag accompanies outputs +0 (00000000_{16}) and −0 (80000000_{16}). For operations producing an integer, the flag accompanies the output 0 (00000000_{16}).

NAN Flag: The NAN flag is HIGH if an R PLUS S, R MINUS S, R TIMES S, 2 MINUS S, or FP-TO-INT operation produces a NAN as a final result.

Operation in DEC Mode

When input signal IEEE/$\overline{\text{DEC}}$ is LOW, the DEC mode of operation is selected. In this mode the Am29C325 uses the single-precision floating-point format (floating F) set forth in

Digital Equipment Corporation's VAX Architecture Manual. In addition, the DEC mode complies with most other aspects of single-precision floating-point operation outlined in the manual — differences are discussed in **Appendix B**.

DEC Floating-Point Format

The DEC single-precision floating-point word is 32 bits wide and is arranged in the format shown in Figure 17. The floating-point word is divided into three fields: a single-bit sign, an 8-bit biased exponent, and a 23-bit fraction.

The sign bit indicates the sign of the floating-point number's value. Non-negative values have a sign of 0, negative values a sign of 1.

The biased exponent is an 8-bit unsigned integer field representing a multiplicative factor of some power of two. The bias value is 128. If, for example, the multiplicative factor for a floating-point number is to be 2^a, the value of the biased exponent would be a + 128; "a" is called the true exponent.

The fraction is a 23-bit unsigned fractional field containing the 23 LSBs of the floating-point number's 24-bit mantissa. The weight of this field's MSB is 2^{-2}; the weight of the LSB is 2^{-24}.

A floating-point number is evaluated or interpreted per the following conventions:
let s = sign bit
e = biased exponent
f = fraction

if e = 0 and s = 0...value = 0
if e = 0 and s = 1...value = DEC-reserved operand
if $0 < e \leqslant 255$...value = $(-1)^{s}*(2^{e-128})*(.1f)$
(normalized number)

Zero: The value zero always has a sign of zero.

DEC-Reserved Operand: A DEC-reserved operand does not represent a numeric value but is interpreted as a signal or symbol. DEC-reserved operands are used to indicate invalid operations and operations whose results have overflowed the destination format. They may also be used to pass symbolic information from one calculation to another.

Normalized Number: A normalized number represents a quantity with magnitude greater than or equal to 2^{-128} but less than 2^{127}.

Example 1:

The number + 3.5 can be represented in floating-point format as follows:

$$+3.5 = 11.1_2 \times 2^0 = .111_2 \times 2^2$$

sign = 0

biased exponent = $2_{10} + 128_{10} = 130_{10} = 10000010_2$

fraction = $11000000000000000000000_2$
(the leading 1 is implied in the format)

Concatenating these fields produces the floating-point word 41600000_{16}.

Example 2:

The number −11.375 can be represented in floating-point format as follows:

$$-11.375 = -1011.011_2 \times 2^0 = -.1011011_2 \times 2^4$$

sign = 1

biased exponent = $4_{10} + 128_{10} = 132_{10} = 10000100_2$

fraction = $01101100000000000000000_2$
(the leading 1 is implied in the format)

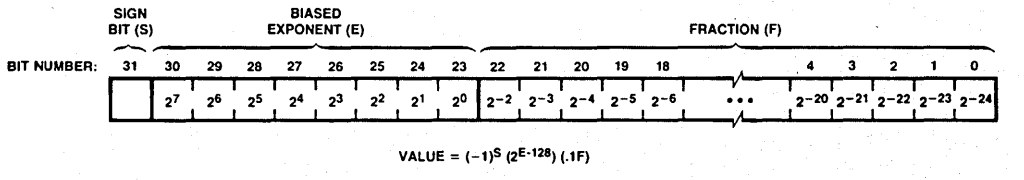
Concatenating these fields produces the floating-point word $C2360000_{16}$.

DEC Mode Integer Format

DEC mode integer format is identical to that of the IEEE mode. Integer numbers are represented as 32-bit, two's complement words (Figure 8 depicts the integer format). The integer word can represent a range of integer values from -2^{31} to $2^{31} - 1$.

Operations

All eight floating-point ALU operations discussed in the **General Description** section can be performed in DEC mode.



TB000671

Figure 17. DEC-Mode Floating-Point Format

Various exceptional aspects of the R PLUS S, R MINUS S, R TIMES S, 2 MINUS S, INT-TO-FP, and FP-TO-INT operations for this mode are described below. The IEEE-TO-DEC and DEC-TO-IEEE operations are discussed separately in the **IEEE-TO-DEC and DEC-TO-IEEE Operations** section.

Operations with DEC-Reserved Operands: DEC-reserved operands arise in two ways: 1) they can be generated by the Am29325 to indicate that an invalid operation or floating-point overflow has taken place, or 2) be provided by the user as an input operand.

When a DEC-reserved operand appears as an input operand, the final result of the operation is the same DEC-reserved operand. If an operation has two DEC-reserved operands as inputs, the DEC-reserved operand on the R port becomes the final result.

The NAN flag will be HIGH whenever an operation produces a DEC-reserved operand as a final result.

Example 1:

Suppose the floating-point addition operation is performed with the following input operands:

R port: 40800000_{16} $(0.1*2^1)$
S port: 80012345_{16} (DEC-reserved operand)

Result: This operation produces the DEC-reserved operand on the S port, 80012345_{16}, as the final result. The NAN flag will be HIGH.

Example 2:

Suppose the floating-point multiplication operation is performed with the following input operands:

R port: 80765432_{16} (DEC-reserved operand)
S port: 80000001_{16} (DEC-reserved operand)

Result: Since both input operands are DEC-reserved operands, the operand on the R port, 80765432_{16}, is the final result of the operation. The NAN flag will be HIGH.

Operations Producing Overflows: If an operation produces a rounded result that is too large to fit in the the destination format, that operation is said to have overflowed.

A floating-point overflow occurs if an R PLUS S, R MINUS S, R TIMES S, or 2 MINUS S operation with finite input operand(s) produces a result which, after rounding, has a magnitude greater than or equal to 2^{127}. The final result in such cases will be DEC-reserved operand 80000000_{16}; the overflow, inexact, and NAN flags will be HIGH.

Integer overflow occurs when the "floating-point-to-integer" conversion operation attempts to convert to integer a floating-point number which, after rounding, is greater than $2^{31} - 1$ or less than -2^{31}. The final result in such cases will be DEC-reserved operand 80000000_{16}; the invalid operation flag will be HIGH. Note that the overflow and inexact flags remain LOW for integer overflow.

Operations Producing Underflows: If an operation produces a floating-point result which, after rounding, has a magnitude too small to be expressed as a normalized floating-point number but greater than 0, that operation is said to have underflowed. Underflow occurs when an R PLUS S, R MINUS S, or R TIMES S operation produces a result which, after rounding, has the magnitude:

$0 < \text{magnitude} < 2^{-128}$.

The final result in such cases will be 0 (00000000_{16}). The underflow, inexact, and zero flags will be HIGH.

Underflow does not occur if the destination format is integer. If the infinitely precise result of a floating-point-to-integer conversion has a magnitude greater than 0 and less than 1 but the rounded result is 0, the underflow flag remains LOW.

Invalid Operations: If an input operand is invalid for the operation to be performed, that operation is considered invalid. There is only one invalid operation in DEC mode: performing a floating-point-to-integer conversion on a value too large to be converted to an integer. In this case, the final result will be DEC-reserved operand 80000000_{16}, and the invalid operation and NAN flags will be HIGH.

Sign Bit

For all operations producing a DEC floating-point result, the sign bit of the final result is unambiguous; i.e., there is only one sign bit value that yields a numerically correct result.

Rounding

There are four rounding modes for DEC operation: 1) round to nearest, 2) round toward $+\infty$, 3) round toward $-\infty$, and 4) round toward 0. The round toward $+\infty$, round toward $-\infty$, and round toward 0 modes are performed in a manner identical to that for IEEE operation; refer to the **Rounding** section under **Operation In IEEE Mode**. The round to nearest mode is similar to that for IEEE operation but differs in one respect: for the case in which the infinitely precise result of an operation is exactly halfway between two representable values, DEC round to nearest mode rounds to the value with the larger magnitude, rather than to the value whose LSB is 0.

Flag Operation

The Am29C325 generates six status flags to monitor floating-point processor operation. The following is a summary of flag operation in DEC mode:

Invalid Operation Flag: The invalid operation flag is HIGH if the FP-TO-INT operation is performed on a floating-point number too large to be converted to an integer. The final result for such an operation will be the DEC-reserved operand 80000000_{16}.

Overflow Flag: The overflow flag is HIGH if an R PLUS S, R MINUS S, R TIMES S, or 2 MINUS S operation produces a result which, after rounding, has a magnitude greater than or equal to 2^{127}. The final result will be the DEC-reserved operand 80000000_{16}.

Underflow Flag: The underflow flag is HIGH if an R PLUS S, R MINUS S, or R TIMES S operation produces a result which, after rounding, has a magnitude in the range:

$0 < \text{magnitude} < 2^{-128}$.

The final result will be 0 (00000000_{16}) in such cases.

Inexact Flag: The inexact flag is HIGH if the final result of an R PLUS S, R MINUS S, R TIMES S, 2 MINUS S, INT-TO-FP, or FP-TO-INT operation is not equal to the infinitely precise result. Note that if the underflow or overflow flag is HIGH, the inexact flag will also be HIGH.

Zero Flag: The zero flag is HIGH if the final result of an operation is 0. For operations producing an integer or a DEC floating-point number, the flag accompanies the output 0 (00000000_{16}). (It should be noted that any operation producing a floating-point 0 in DEC mode will output 00000000_{16}.)

NAN Flag: The NAN flag is HIGH if an R PLUS S, R MINUS S, R TIMES S, 2 MINUS S, or FP-TO-INT operation produces a DEC-reserved operand as the final result.

IEEE-TO-DEC and DEC-TO-IEEE Operations

The IEEE-TO-DEC and DEC-TO-IEEE operations are used to convert floating-point numbers between the IEEE and DEC formats. Both operations work in a manner independent of the IEEE/$\overline{\text{DEC}}$ mode control.

IEEE-TO-DEC Conversion

The operation converts an IEEE floating-point number to DEC floating-point format. Most conversions are exact; in no case does the round mode have any effect on the final result. There are, however, a few exceptional cases:

a) If the IEEE floating-point input has a magnitude greater than or equal to 2^{127}, it is too large to be represented by a DEC floating-point number. The final result will be the DEC-reserved operand 80000000_{16}; the overflow, inexact, and NAN flags will be HIGH.

b) If the IEEE floating-point input is a NAN, the final result will be the DEC-reserved operand 80000000_{16}; the invalid and NAN flags will be HIGH.

c) If the IEEE floating-point input is a denormalized number, the final result will be a DEC 0 (0000000_{16}); the zero flag will be HIGH.

d) If the IEEE floating-point input is + 0 or −0, the final result will be a DEC 0 (0000000_{16}); the zero flag will be HIGH.

DEC-TO-IEEE Conversion

This operation converts a DEC floating-point number to IEEE floating-point format. Most conversions are exact; in no case does the round mode have any effect on the final result. There are, however, a few exceptional cases:

a) If the DEC floating-point input is not 0 but has a magnitude less than 2^{-126}, it is too small to be expressed as a normalized IEEE floating-point number. The final result will be an IEEE floating-point 0 having the same sign as the input (0000000_{16} for positive inputs and 80000000_{16} for negative inputs); the underflow, inexact, and zero flags will be HIGH.

b) If the DEC floating-point input is a DEC-reserved operand, the result will be quiet NAN $7FA0000_{16}$; the invalid operation and NAN flags will be HIGH.

c) If the DEC floating-point input is 0, the final result will be IEEE floating-point + 0 (0000000_{16}); the zero flag will be HIGH.

APPLICATIONS

Suggestions for Power and Ground Pin Connections

The Am29C325 operates in an environment of fast signal rise times and substantial switching currents. Therefore, care must be exercised during circuit board design and layout, as with any high-performance component. The following is a suggested layout, but since systems vary widely in electrical configuration, an empirical evaluation of the intended layout is recommended.

The V_{CCO} and GNDO pins carry output driver switching currents and can be electrically noisy. The V_{CC} and GND pins, which supply the logic core of the device, tend to produce less noise, and the circuits they supply may be adversely affected by noise spikes on the V_{CC} plane. For this reason, it is best to provide isolation between the V_{CC} and V_{CCO} pins, as well as independent decoupling for each. Isolating the GND and GNDO pins is not required.

Printed Circuit-Board Layout Suggestions

1. Use of a multi-layer PC board with separate power, ground, and signal planes is highly recommended.

2. All V_{CC} and V_{CCO} pins should be connected to the V_{CC} plane. V_{CC} pins should be isolated from V_{CCO} pins by means of an isolation slot which is cut in the V_{CC} plane; see Figure 18. By physically separating the V_{CC} and V_{CCO} pins, coupled noise will be reduced.

3. All GND and GNDO pins should be connected directly to the ground plane.

4. The V_{CCO} pins should be decoupled to ground with a 0.1-μF ceramic capacitor and a 10-μF electrolytic capacitor, placed as closely to the Am29C325 as is practical. V_{CC} pins should be decoupled to ground in a similar manner.

A suggested layout is shown in Figure 18.

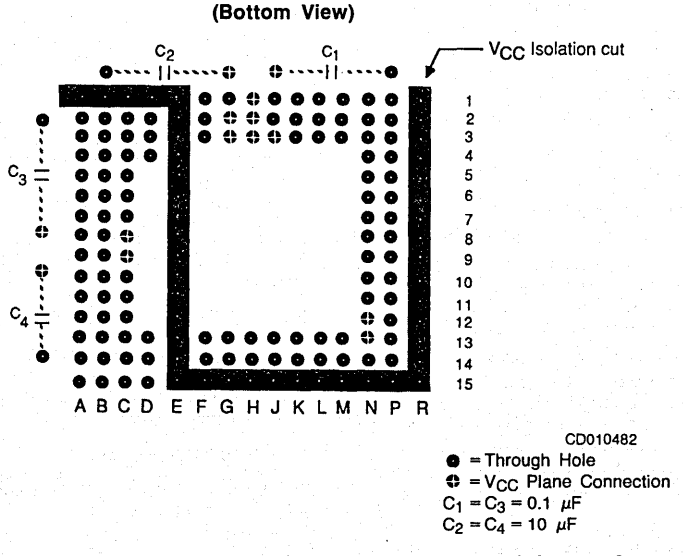


Figure 18. Suggested Printed-Circuit Board Layout (Power and Ground Connections)

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature Under Bias ..... −55 to +125°C
Supply Voltage to Ground Potential
Continuous ..... −0.3 to +7.0 V
DC Voltage Applied to Outputs
for HIGH Output State ..... −0.3 V to +V_{CC} + 0.3 V
DC Input Voltage ..... −0.3 to V_{CC} + 0.3 V
DC Output Current, into LOW Outputs ..... 30 mA
DC Input Current ..... −10 to +10 mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature, Case (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V

Military* (M) Devices
Temperature (T_A) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... +4.5 V to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

Thermal Resistance (Typical)

Symbol	CGX145	Unit
θ_{JA}	23	°C/W

*Military product 100% tested at T_A = +25°C, +125°C, and −55°C.

DC CHARACTERISTICS over operating range unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions (Note 1)			Min.	Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC} = Min. V_{IN} = V_{IL} or V_{IH}		I_{OH} = 0.4 mA	2.4		V
V_{OL}	Ouput LOW Voltage	V_{CC} = Min. V_{IN} = V_{IL} or V_{IH}		I_{OL} = 4.0 mA		0.5	V
V_{IH}	Guaranteed Input Logical HIGH Voltage (Note 2)				2.0		V
V_{IL}	Guaranteed Input Logical LOW Voltage (Note 2)					0.8	V
I_{IL}	Input LOW Current	V_{CC} = Max. V_{IN} = 0.5 V				−10	μA
I_{IH}	Input HIGH Current	V_{CC} = Max. V_{IN} = V_{CC} − 0.5 V				10	μA
I_{OZH}	Off-State (HIGH Impedance) Output Current	V_{CC} = Max., V_O = 2.4 V				10	μA
I_{OZL}	Off-State (HIGH Impedance) Output Current	V_{CC} = Max., V_O = 0.5 V				−10	μA
I_{CC}	Static Power Supply Current (Note 3)	V_{CC} = Max. I_O = 0 μA	COM'L T_A = 0 to +70°C	CMOS V_{IN} = V_{CC} or GND		20	mA
				TTL V_{IN} = 0.5 V or 2.4 V		20	
			MIL T_A = −55 to +125°C	CMOS V_{IN} = V_{CC} or GND		25	
				TTL V_{IN} = 0.5 V or 2.4 V		25	
C_{PD}	Power Dissipation Capacitance (Notes 4,5)	V_{CC} = MAX No Load				9,000	pF

Notes: 1. V_{CC} conditions shown as Min. or Max. refer to the commercial and military V_{CC} limits.
2. These input levels provide zero-noise immunity and should only be statically tested in a noise-free environment (not functionally tested).
3. Use CMOS I_{CC} when the device is driven by CMOS circuits and TTL I_{CC} when the device is driven by TTL circuits.
4. C_{PD} determines the dynamic current consumption:

I_{CC} (Total) = I_{CC} (Static) + $(C_{PD} + nCL)V_{CC}\frac{f}{2}$, where f is the clock frequency, CL output load capacitance, and n number of loads.

5. Tested on a sample basis.

CAPACITANCE

C_{IN}	Input Capacitance	fc = 1 MHz (Note 5)	12	pF
C_{OUT}	Output Capacitance		12	pF

SWITCHING CHARACTERISTICS over COMMERCIAL operating range

No.	Parameter Symbol	Parameter Description		Test Conditions	29C325 Min.	29C325 Max.	29C325-1 Min.	29C325-1 Max.	29C325-2 Min.	29C325-2 Max.	Unit
1	t_{ASC}	Clocked Add, Subtract Time (R PLUS S, R MINUS S, 2 MINUS S)				124		109		87	ns
2	t_{MC}	Clocked Multiply Time (R TIMES S)				107		97		78	ns
3	t_{CC}	Clocked Conversion Time (INT-TO-FP, FP-TO-INT, IEEE-TO-DEC, DEC-TO-IEEE)				105		94		75	ns
4	t_{ASUC}	Unclocked Add, Subtract Time (R, S to F, Flags) for R PLUS S, R MINUS S,and 2 MINUS S Instructions				146		135		108	ns
5	t_{MUC}	Unclocked Multiply Time (R, S to F, Flags) for R TIMES S Instruction		FT_0 = HIGH FT_1 = HIGH		154		142		114	ns
6	t_{CUC}	Unclocked Conversion Time (R, S to F, Flags) for INT-TO-FP, FP-TO-INT, IEEE- TO-DEC and DEC-TO-IEEE Instructions				133		122		98	ns
7	t_{PWH}	Clock Pulse Width HIGH			15		15		15		ns
8	t_{PWL}	Clock Pulse Width LOW			15		15		15		ns
9	t_{PDOF1}	Clock to F_0-F_{31} and Flag Outputs		FT_0 = LOW FT_1 = HIGH		145		130		104	ns
10	t_{PDOF2}			FT_1 = LOW		26		24		22	ns
11	t_{PZL}	OE Enable Time	Z to LOW			22		22		20	ns
12	t_{PZH}		Z to HIGH			22		22		20	ns
13	t_{PLZ}	OE Disable Time	LOW to Z			13		13		12	ns
14	t_{PHZ}		HIGH to Z			13		13		12	ns
15	t_{PZLLSB}	Clock ↑ to F_0-F_{15} Enable, 16-Bit I/O Mode	Z to LOW	S16/32 = HIGH ONEBUS = LOW		26		26		24	ns
16	t_{PZHLSB}		Z to HIGH			26		26		24	ns
17	t_{PLZLSB}	Clock ↓ to F_0-F_{15} Disable, 16-Bit I/O Mode	LOW to Z			22		22		20	ns
18	t_{PHZLSB}		HIGH TO Z			22		22		20	ns
19	t_{PZLMSB}	Clock ↓ to $F_{16}-F_{31}$ Enable, 16-Bit I/O Mode	Z to LOW	S16/32 = HIGH ONEBUS = LOW		26		26		24	ns
20	t_{PZHMSB}		Z to HIGH			26		26		24	ns
21	t_{PLZMSB}	Clock ↑ to $F_{16}-F_{31}$ Disable,16-Bit I/O Mode	LOW to Z			22		22		20	ns
22	t_{PHZMSB}		HIGH to Z			22		22		20	ns
23	t_{SCE}	Register Clock Enable Setup Time		FT_0 = LOW FT_1 = LOW	10		9		8		ns
24	t_{HCE}	Register Clock Enable Hold Time		FT_0 = LOW FT_1 = LOW	3		3		3		ns
25	t_{SD1}	R_0-R_{31}, S_0-S_{31} Setup Time (see note below)		FT_0 = LOW	20		16		14		ns
26	t_{HD1}	R_0-R_{31}, S_0-S_{31} Hold Time (see note below)			5		5		4		ns
27	t_{SD2}	R_0-R_{31}, S_0-S_{31} Setup Time (see note below)		FT_0 = HIGH FT_1 = LOW	133		118		94		ns
28	t_{HD2}	R_0-R_{31}, S_0-S_{31} Hold Time (see note below)			0		0		0		ns
29	t_{SI02}	I_0-I_2 Instruction Select Setup Time		FT for Destination Register = LOW	132		114		91		ns
30	t_{HI02}	I_0-I_2 Instruction Select Hold Time			1		1		1		ns
31	t_{PDI02}	I_0-I_2 Instruction Select to F_0-F_{31}, Flags		FT_1 = HIGH		150		134		107	ns
32	t_{SI3}	I_3 Port S Input Select Setup Time		FT_1 = LOW	89		78		63		ns
33	t_{HI3}	I_3 Port S Input Select Hold Time			0		0		0		ns
34	t_{SI4}	I_4 Register R Input Select Setup Time (see note below)		FT_0 = LOW	18		13		11		ns
35	t_{HI4}	I_4 Register R Input Select Hold Time (see note below)			3		3		3		ns
36	t_{SRM}	Round Mode Select Setup Time		FT for Destination Register = LOW	67		57		46		ns
37	t_{HRM}	Round Mode Select Hold Time			3		3		3		ns
38	t_{PRF}	Round Mode Select to F_0-F_{31}, Flags		FT_1 = HIGH		78		69		55	ns

Note: See timing diagram for desired mode of operation to determine clock edge to which these setup and hold times apply.

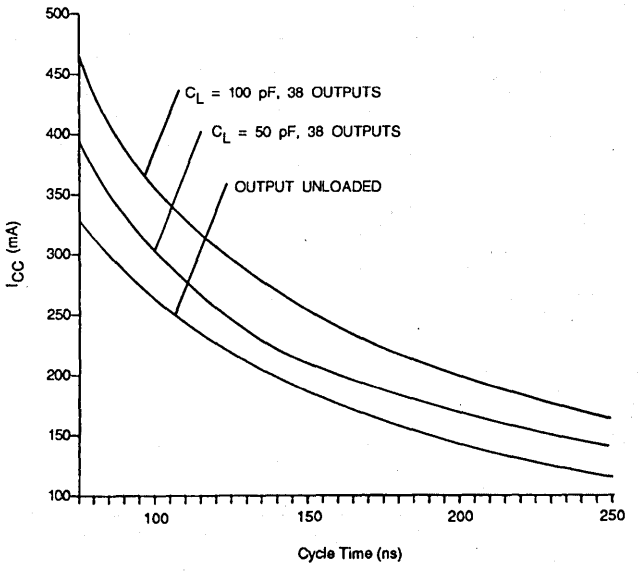
Preliminary

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

No.	Parameter Symbol	Parameter Description		Test Conditions	29C325 Min.	29C325 Max.	29C325-1 Min.	29C325-1 Max.	Unit
1	t_{ASC}	Clocked Add, Subtract Time (R PLUS S, R MINUS S, 2 MINUS S)				149		131	ns
2	t_{MC}	Clocked Multiply Time (R TIMES S)				133		117	ns
3	t_{CC}	Clocked Conversion Time (INT-TO-FP, FP-TO-INT, IEEE-TO-DEC, DEC-TO-IEEE)				130		114	ns
4	t_{ASUC}	Unclocked Add, Subtract Time (R, S to F, Flags) for R PLUS S, R MINUS S, and 2 MINUS S Instructions				178		163	ns
5	t_{MUC}	Unclocked Multiply Time (R, S to F, Flags) for R TIMES S Instruction		FT_0 = HIGH FT_1 = HIGH		190		179	ns
6	t_{CUC}	Unclocked Conversion Time (R, S to F, Flags) for INT-TO-FP, FP-TO-INT, IEEE-TO-DEC and DEC-TO-IEEE Instructions				161		150	ns
7	t_{PWH}	Clock Pulse Width HIGH			15		15		ns
8	t_{PWL}	Clock Pulse Width LOW			15		15		ns
9	t_{PDOF1}	Clock to $F_0 - F_{31}$ and Flag Outputs		FT_0 = LOW FT_1 = HIGH		166		156	ns
10	t_{PDOF2}			FT_1 = LOW		35		33	ns
11	t_{PZL}	$\overline{OE}$ Enable Time	Z to LOW			29		29	ns
12	t_{PZH}		Z to HIGH			29		29	ns
13	t_{PLZ}	$\overline{OE}$ Disable Time	LOW to Z			16		16	ns
14	t_{PHZ}		HIGH to Z			16		16	ns
15	t_{PZLLSB}	Clock ↑ to $F_0 - F_{15}$ Enable, 16-Bit I/O Mode	Z to LOW	S16/$\overline{32}$ = HIGH ONEBUS = LOW		31		31	ns
16	t_{PZHLSB}		Z to HIGH			31		31	ns
17	t_{PLZLSB}	Clock ↓ to $F_0 - F_{15}$ Disable, 16-Bit I/O Mode	LOW to Z			23		23	ns
18	t_{PHZLSB}		HIGH TO Z			23		23	ns
19	t_{PZLMSB}	Clock ↓ to $F_{16} - F_{31}$ Enable, 16-Bit I/O Mode	Z to LOW	S16/$\overline{32}$ = HIGH ONEBUS = LOW		31		31	ns
20	t_{PZHMSB}		Z to HIGH			31		31	ns
21	t_{PLZMSB}	Clock ↑ to $F_{16} - F_{31}$ Disable,16-Bit I/O Mode	LOW to Z			23		23	ns
22	t_{PHZMSB}		HIGH to Z			23		23	ns
23	t_{SCE}	Register Clock Enable Setup Time		FT_0 = LOW FT_1 = LOW	11		10		ns
24	t_{HCE}	Register Clock Enable Hold Time		FT_0 = LOW FT_1 = LOW	4		4		ns
25	t_{SD1}	$R_0 - R_{31}$, $S_0 - S_{31}$ Setup Time (see note below)		FT_0 = LOW	21		16		ns
26	t_{HD1}	$R_0 - R_{31}$, $S_0 - S_{31}$ Hold Time (see note below)			6		6		ns
27	t_{SD2}	$R_0 - R_{31}$, $S_0 - S_{31}$ Setup Time (see note below)		FT_0 = HIGH FT_1 = LOW	163		148		ns
28	t_{HD2}	$R_0 - R_{31}$, $S_0 - S_{31}$ Hold Time (see note below)			0		0		ns
29	t_{SI02}	$I_0 - I_2$ Instruction Select Setup Time		FT for Destination Register = LOW	167		143		ns
30	t_{HI02}	$I_0 - I_2$ Instruction Select Hold Time			2		2		ns
31	t_{PDI02}	$I_0 - I_2$ Instruction Select to $F_0 - F_{31}$, Flags		FT_1 = HIGH		186		166	ns
32	t_{SI3}	I_3 Port S Input Select Setup Time		FT_1 = LOW	109		94		ns
33	t_{HI3}	I_3 Port S Input Select Hold Time			0		0		ns
34	t_{SI4}	I_4 Register R Input Select Setup Time (see note below)		FT_0 = LOW	18		13		ns
35	t_{HI4}	I_4 Register R Input Select Hold Time (see note below)			5		5		ns
36	t_{SRM}	Round Mode Select Setup Time		FT for Destination Register = LOW	85		71		ns
37	t_{HRM}	Round Mode Select Hold Time			5		5		ns
38	t_{PRF}	Round Mode Select to $F_0 - F_{31}$, Flags		FT_1 = HIGH		102		95	ns

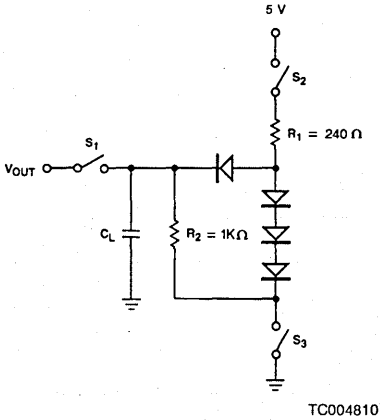
Note: See timing diagram for desired mode of operation to determind clock edge to which these setup and hold times apply.

Am29C325 I_{CC} vs Cycle Time



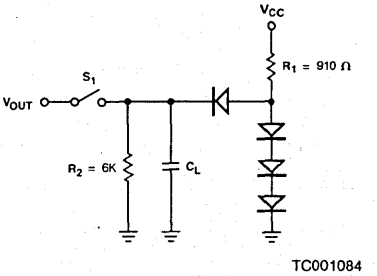
OP002770

SWITCHING TEST CIRCUITS



TC004810

A. Three-State Outputs



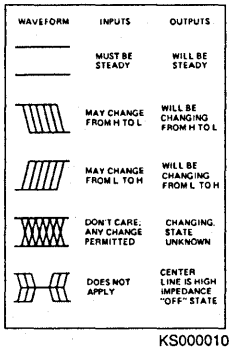
TC001084

B. Normal Outputs

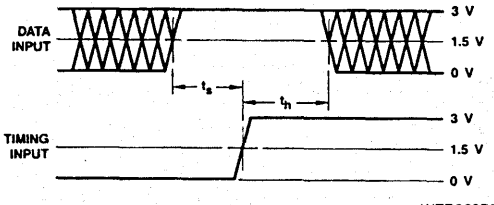
Notes: 1. C_L = 50 pF includes scope probe, wiring, and stray capacitances without device in test fixture.
2. S_1, S_2, S_3 are closed during function tests and all AC tests except output enable tests.
3. S_1 and S_3 are closed while S_2 is open for t_{PZH} test.
S_1 and S_2 are closed while S_3 is open for t_{PZL} test.
4. C_L = 5.0 pF for output disable tests.

SWITCHING TEST WAVEFORMS

KEY TO SWITCHING WAVEFORMS



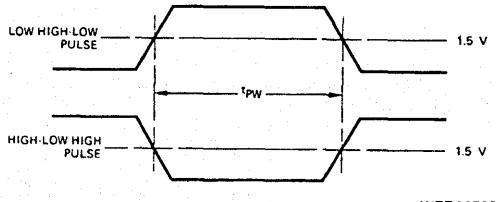
KS000010



WFR02970

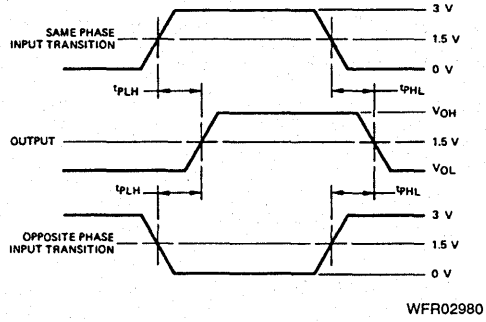
Notes: 1. Diagram shown for HIGH data only.
Output transition may be opposite sense.
2. Cross hatched area is don't care condition.

Set-Up, Hold, and Release Times



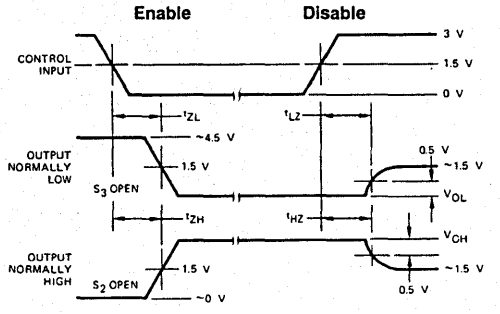
WFR02790

Pulse Width



WFR02980

Propagation Delay

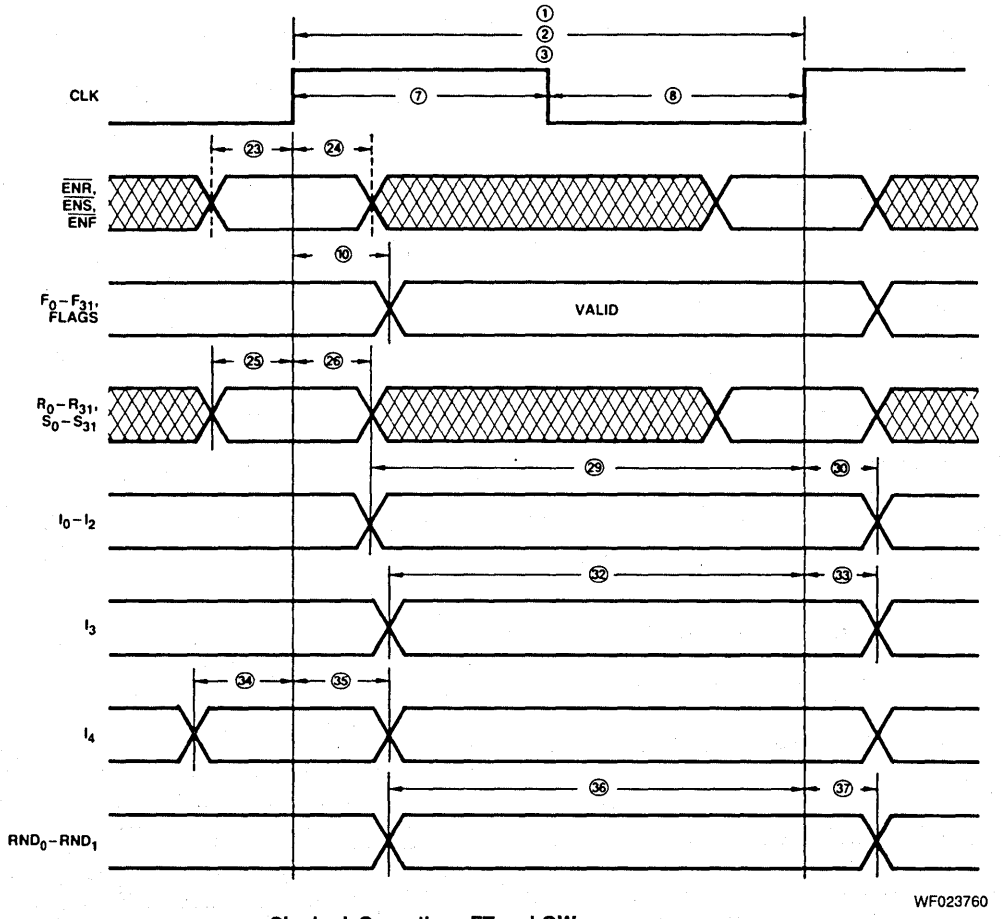


WFR02660

Notes: 1. Diagram shown for Input Control Enable-LOW and Input Control Disable-HIGH.
2. S_1, S_2 and S_3 of Load Circuit are closed except where shown.

Enable and Disable Times

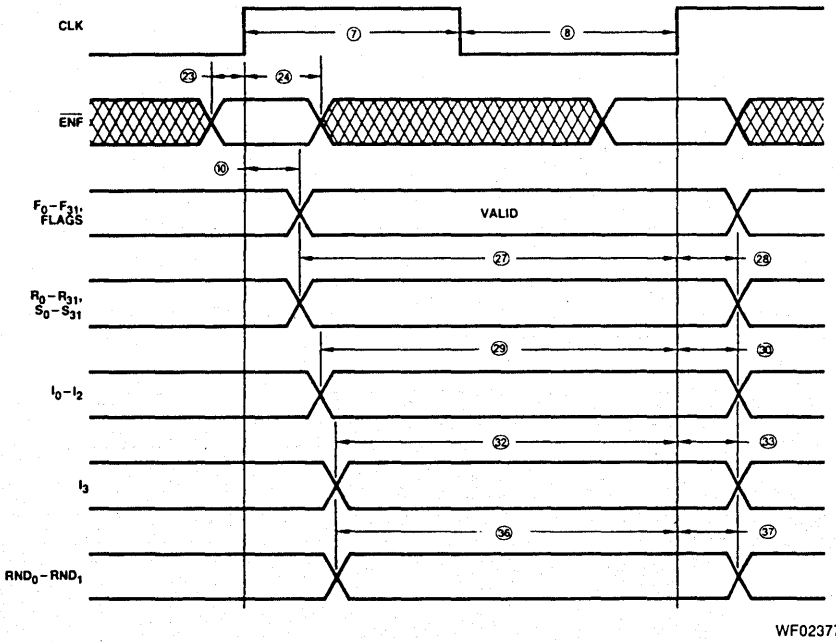
SWITCHING WAVEFORMS



WF023760

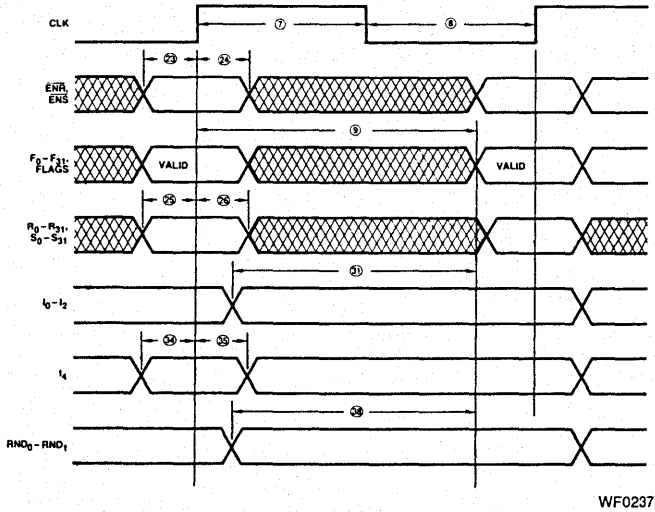
Clocked Operation: FT_0 = LOW
FT_1 = LOW

SWITCHING WAVEFORMS (Cont'd.)



WF023770

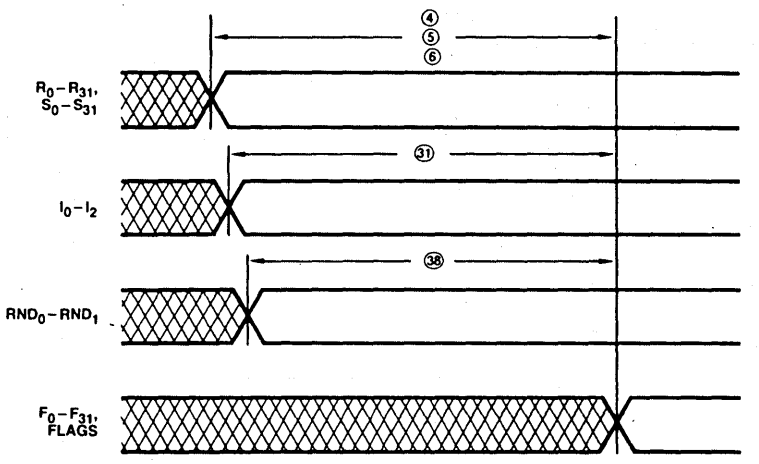
Clocked Operation: FT_0 = HIGH
FT_1 = LOW



WF023780

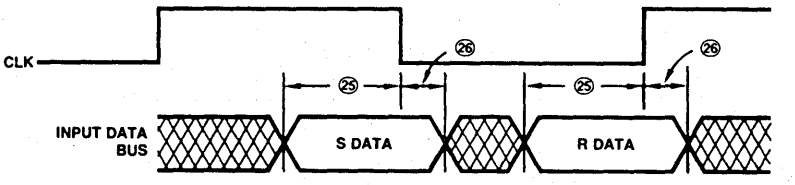
Clocked Operation: FT_0 = LOW
FT_1 = HIGH

SWITCHING WAVEFORMS (Cont'd.)



WF023790

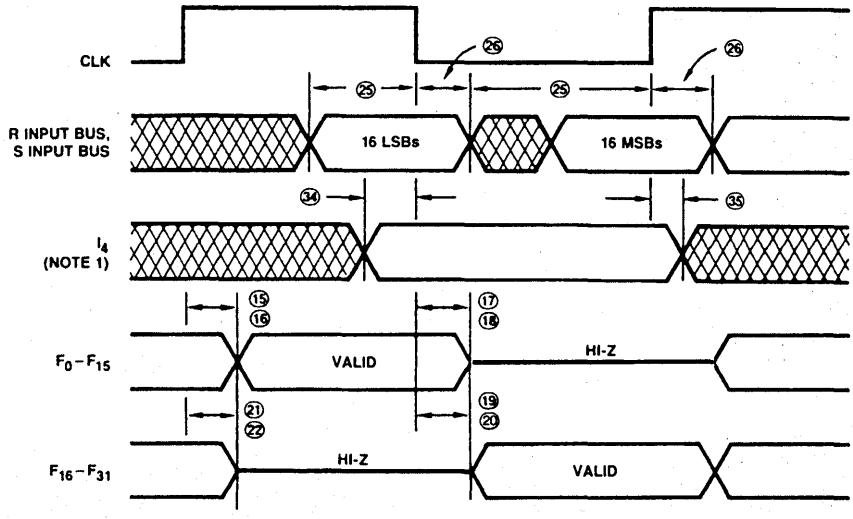
Flow-Through Operation (FT_0 = HIGH, FT_1 = HIGH)



WF023800

32-Bit, Single-Input Bus Mode

SWITCHING WAVEFORMS (Cont'd.)



WF023811

Note 1. 1_4 has special setup and hold time requirements in this mode. All other control signals have timing requirements as shown in the diagram "Clocked operation, FT_0 = LOW, FT_1 = LOW."

16-Bit, Two-Input Bus Mode

TEST PHILOSOPHY AND METHODS

The following eight points describe AMD's philosophy for high volume, high speed automatic testing.

1. Ensure that the part is adequately decoupled at the test head. Large changes in V_{CC} current as the device switches may cause erroneous function failures due to V_{CC} changes.

2. Do not leave inputs floating during any tests, as they may start to oscillate at high frequency.

3. Do not attempt to perform threshold tests at high speed. Following an output transition, ground current may change by as much as 400 mA in 5-8 ns. Inductance in the ground cable may allow the ground pin at the device to rise by hundreds of millivolts momentarily.

4. Use extreme care in defining point input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins and they may not actually reach V_{IL} or V_{IH} until the noise has settled. AMD recommends using $V_{IL} \leqslant 0$ V and $V_{IH} \geqslant 3.0$ V for AC tests.

5. To simplify failure analysis, programs should be designed to perform DC, Function, and AC tests as three distinct groups of tests.

6. Capacitive Loading for AC Testing

   Automatic testers and their associated hardware have stray capacitance that varies from one type of tester to another, but is generally around 50 pF. This, of course, makes it impossible to make direct measurements of parameters which call for smaller capacitive load than the associated stray capacitance. Typical examples of this are the so-called "float delays," which measure the propagation delays into the high-impedance state and are usually specified at a load capacitance of 5.0 pF. In these cases, the test is performed at the higher load capacitance (typically 50 pF), and engineering correlations based on data taken with a bench setup are used to predict the result at the lower capacitance.

   Similarly, a product may be specified at more than one capacitive load. Since the typical automatic tester is not capable of switching loads in mid-test, it is impossible to make measurements at both capacitances even though they may both be greater than the stray capacitance. In these cases, a measurement is made at one of the two capacitances. The result at the other capacitance is predicted from engineering correlations based on data taken with a bench setup and the knowledge that certain DC measurements (I_{OH}, I_{OL} for example) have already been taken and are within spec. In some cases, special DC tests are performed in order to facilitate this correlation.

7. Threshold Testing

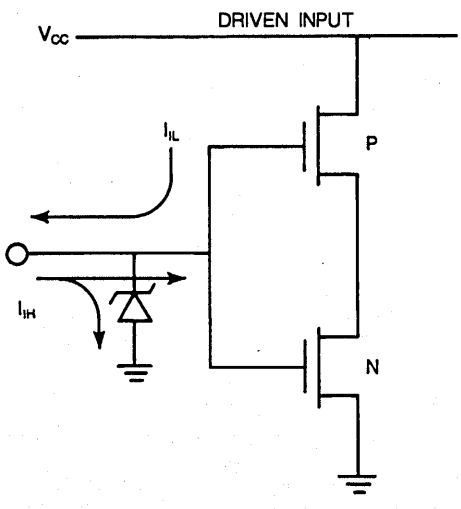
   The noise associated with automatic testing (due to the long, inductive cables) and the high gain of the tested device when in the vicinity of the actual device threshold, frequently give rise to oscillations when testing high speed circuits. These oscillations are not indicative of a reject device, but instead of an overtaxed test system. To minimize this problem, thresholds are tested at least once for each input pin. Thereafter, "hard" high and low levels are used for other tests. Generally this means that function and AC testing are performed at "hard" input levels rather than at V_{IL} Max. and V_{IH} Min.

8. AC Testing

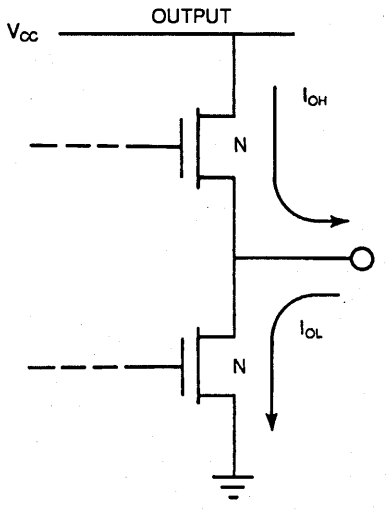
   Occasionally, parameters are specified that cannot be measured directly on automatic testers because of tester limitations. Data input hold times often fall into this category. In these cases, the parameter in question is guaranteed by correlating these tests with other AC tests that have been performed. These correlations are arrived at by the cognizant engineer by using precise bench measurements in conjunction with the knowledge that certain DC parameters have already been measured and are within spec.

   In some cases, certain AC tests are redundant, since they can be shown to be predicted by some other tests which have already been performed. In these cases, the redundant tests are not performed.

INPUT/OUTPUT CIRCUIT DIAGRAMS



IC000865



IC000870

2

APPENDIX A

DIFFERENCES BETWEEN THE IEEE PROPOSED STANDARD FOR BINARY FLOATING-POINT ARITHMETIC AND THE Am29C325'S IEEE MODE

When operated in IEEE mode, the Am29C325 high-speed floating-point processor complies with the single-precision portion of the IEEE Proposed Standard for Binary Floating-Point Arithmetic (P754, draft 10.0) in most respects. There are, however, several differences:

Denormalized Numbers

The Am29C325 does not handle denormalized numbers. A denormalized input will be converted to zero of the same sign before the specified operation takes place. The operation proceeds in exactly the same manner as if the input were +0 or −0, producing the same numerical result and flags.

If the result of an operation, after rounding, has a magnitude smaller than 2^{-126}, the result is replaced by a zero of the same sign.

Representation of Overflows

In some rounding modes the proposed IEEE standard requires that overflows be represented as the format's most-positive or most-negative finite number. In particular:

When rounding toward 0, all overflows should produce a result of the largest representable finite number with the sign of the intermediate result.

When rounding toward $-\infty$, all positive overflows should produce a result of the largest representable positive finite number.

When rounding toward $+\infty$, all negative overflows should produce a result of the largest representable negative finite number.

The Am29C325, however, always represents positive overflows as $+\infty$ and negative overflows as $-\infty$, regardless of rounding mode.

Projective Mode

The proposed IEEE standard provides only for an affine mode to control the handling of infinities. The Am29C325 provides both affine and projective modes; the desired mode can be selected by the user.

Traps

The proposed IEEE standard stipulates that the user be able to request a trap on any exception. The Am29C325 does not support trapped operation and behaves as if traps are disabled.

Resetting of Flags

The proposed IEEE standard states that once an exception flag has been set, it is reset only at the user's request. The Am29C325's flags, however, reflect the status of the most recent operation.

Generation of the Underflow Flag

The proposed IEEE standard suggests several possible criteria for determining if underflow occurs. These criteria generate underflow flags that differ in subtle ways. The underflow criteria chosen for the Am29C325 stipulate that underflow occurs if:

a) the rounded result of an operation has a magnitude in the range:

$$0 < \text{magnitude} < 2^{-126},$$

and

b) the final result is not equal to the infinitely precise result.

Since the Am29C325 never produces a denormalized number as the final result of a calculation, condition (b) is true whenever (a) is true. Note then that the operation of the Am29C325's underflow flag is somewhat different than that of an "IEEE standard" system using the same underflow criteria. For example, if an operation should produce an infinitely precise result that is exactly 2^{-127}, an "IEEE standard" system would produce that value as the final result, expressed as a denormalized number. Since that system's final result is exact, the underflow flag would remain LOW. The Am29C325, on the other hand, would output zero; since its final result is not exact, the underflow flag would be HIGH.

APPENDIX B

DIFFERENCES BETWEEN DEC VAX AND Am29C325 DEC MODE

Operation in DEC mode complies with most aspects of single-precision floating-point operation outlined in the Digital Equipment Corporation's VAX Architecture Manual. However, there are some differences that should be noted:

Format

The Am29C325's DEC format is:

sign	– bit 31
exponent	– bits 30 – 23
mantissa	– 22 – 0

The VAX format is:

sign	– bit 15
exponent	– 14 – 7
mantissa	– bits 6 – 0, bits 31 – 16

In both cases, fields are listed from MSB to LSB, with bit 31 the MSB of the 32-bit word. The Am29C325's DEC format can be converted to VAX format by swapping the 16 LSBs and 16 MSBs of the 32-bit word.

Flags vs. Exceptions

In DEC VAX operation, certain unusual conditions arising during system operation may incur an exception or an indication to the operating system that special handling is needed.

The VAX recognizes a number of arithmetic exceptions. The following exceptions are relevant to the operations supported by the Am29C325:

Integer Overflow Trap: indicates that the last operation produced an integer overflow. The LSBs of the correct result are stored in the destination operand.

Floating-Point Overflow Trap/Fault: indicates that the last operation produced, after normalization and rounding, a floating-point number with magnitude greater than or equal to 2^{127}. A trap replaces the destination operand with the DEC-reserved operand 80000000_{16}; a fault leaves the destination operand unchanged.

Floating-Point Underflow Trap/Fault: indicates that the last operation produced, after normalization and rounding, a floating-point number with magnitude less than 2^{-128}. A trap replaces the destination operand with zero; a fault leaves the destination operand unchanged.

Reserved Operand Fault: indicates that the last operation had a reserved operand as an input. The destination operand is unchanged.

The Am29C325 does not directly support DEC traps and faults. Rather, it indicates unusual conditions by setting one or more of the six status flags HIGH. Table D2 describes flag operation in DEC mode.

Integer Overflow

In cases of integer overflow, the VAX signals the integer overflow trap and stores the LSBs of the correct result. The Am29C325 sets the invalid operation flag and outputs the DEC-reserved operand 80000000_{16}.

Floating-Point Underflow/Overflow Operation

The VAX Architecture Manual specifies the action to be taken on the destination operand when floating-point underflow or overflow is encountered. The Am29C325 has no immediate control over this destination operand, as it resides somewhere off-chip, either in a register or memory location. This isn't so much a difference between the VAX specification and Am29C325 operation as it is a difference in scope.

The Am29C325 responds to floating-point underflow by producing a final result of 0 (00000000_{16}); the underflow, inexact, and zero flags will be HIGH. It responds to floating-point overflow by producing the DEC-reserved operand 80000000_{16} as the final result; the overflow, inexact, and NAN flags will be HIGH.

Handling of DEC-Reserved Operands

If an operation has a DEC-reserved operand as an input, the Am29C325 will produce that operand as the final result. If an operation has two input arguments and both are DEC-reserved operands, the operand on port R becomes the final result. For the VAX, operations with a DEC-reserved operand input or inputs do not modify the destination operand. As mentioned above, control of the destination operand is beyond the scope of the Am29C325's operation.

Inexact Flag

The Am29C325 provides an inexact flag to indicate that the final result produced by an operation is not equal to the infinitely precise result. The VAX does not provide this flag.

APPENDIX C

PERFORMING FLOATING-POINT DIVISION ON THE Am29C325

While the Am29C325 does not have a floating-point division instruction, it can be used to evaluate reciprocals. The division:

$$C = A/B$$

can then be performed by evaluating:

$$C = A*(1/B)$$

Only a modest amount of external hardware is needed to implement the reciprocal function.

The technique for calculating reciprocals is based on the Newton-Raphson method for obtaining the roots of an equation. The roots of equation:

$$F(x) = 0$$

can be found by iteratively evaluating the equation:

$$x_{i+1} = x_i - F(x_i)/F'(x_i)$$

The process begins by making a guess as to the value of x_i and using this guess or "seed" value to perform the first iteration. Iterations are continued until the root is evaluated to the desired accuracy. The number of iterations needed to achieve a given accuracy depends both on the accuracy of the seed value and the nature of F(x).

Now consider the equation:

$$F(x) = (1/x) - B$$

The root of F(x) is 1/B. The reciprocal of B, then, can be found by using the Newton-Raphson method to find the root of F(x). The iterative equation for finding the root is:

$$\begin{aligned} x_{i+1} &= x_i - F(x_i)/F'(X_i) \\ &= x_i - (1/x_i - B)/-(x_i)^{-2} \\ &= x_i\,(2 - B*x_i) \end{aligned}$$

It can be shown that, in order for this iterative equation to converge, the seed value x_0 must fall in the range:

$$\begin{aligned} &0 < x_0 < 2/B && \text{if } B > 0 \\ \text{or}\quad &2/B < x_0 < 0 && \text{if } B < 0 \end{aligned}$$

For example, if the reciprocal of 3 is to be evaluated, the seed value must be between 0 and 2/3.

The error of x_i reduces quadratically; that is, if the error of x_i is e, the error is reduced to order e^2 by the next iteration. The number of bits of accuracy in the result, then, roughly doubles after every iteration. While this is only an approximation of the actual error produced, it is a handy rule of thumb for determining the number of iterations needed to produce a result of a certain accuracy, given the accuracy of the seed.

Example 1:

Find the reciprocal of 7.25.

Solution:

The seed value must fall in the range:

$$0 < x_0 < 2/7.25$$
$$\text{or } 0 < x_0 < .275862$$

Suppose x_0 is chosen to be .1:

Iteration 1: $x_1 = x_0\,(2 - B*x_0)$
$= .1(2 - (7.25)\,(.1))$
$= .1275$

Iteration 2: $x_2 = x_1\,(2 - B*x_1)$
$= .1275(2 - (7.25)\,(.1275))$
$= .1371421875$

Iteration 3: $x_3 = x_2\,(2 - B*x_2)$
$= .1371421875*$
$(2 - (7.25)\,(.1371421875))$
$= .1379265230$

The actual value of 1/7.25, to ten decimal places, is .1379310345.

The error after each iteration is:

Iteration	x_i	Error to Ten Places
0	0.1	-0.0379310345
1	0.1275	-0.0104310345
2	0.1371421875	-0.0007888470
3	0.1379265230	-0.0000045115

Example 2:

Find the reciprocal of -0.3.

Solution:

The seed value must fall in the range:

$$2/(-0.3) < x_0 < 0$$
$$\text{or } -6.66 < x_0 < 0$$

Suppose x_0 is chosen to be -2.0:

Iteration 1: $x_1 = x_0\,(2 - B*x_0)$
$= -2.0(2 - (-0.3)\,(-2.0))$
$= -2.8$

Iteration 2: $x_2 = x_1\,(2 - B*x_1)$
$= -2.8(2 - (-0.3)\,(-2.8))$
$= -3.248$

Iteration 3: $x_3 = x_2\,(2 - B*x_2)$
$= -3.248(2-(-0.3)\,(-3.248))$
$= -3.3311488$

Iteration 4: $x_4 = x_3\,(2 - B*x_3)$
$= -3.3311488*$
$(2-(-0.3)\,(-3.3311488))$
$= -3.333331902$

The actual value of 1/(-0.3), to ten decimal places, is -3.333333333.

The error after each iteration is:

I	x_i	Error to Ten Places
0	-2.0	1.333333333
1	-2.8	0.533333333
2	-3.248	0.085333333
3	-3.3311488	0.002184533
4	-3.333331902	0.000001431

In order to implement the Newton-Raphson method on the Am29C325, some means is needed to generate the seed used in the first iteration. One approach is to place a hardware seed look-up table between the R bus and the Am29C325; see Table C1. A more detailed diagram of the look-up table appears in Figure C2.

TABLE C1. CONTENTS OF THE SEED EXPONENT PROM

DEC		IEEE	
Address (16)	**Data (16)**	**Address (16)**	**Data (16)**
000	(Note 1)	100	(Note 1)
001	(Note 1)	101	FC
002	FF	102	FB
003	FE	103	FA
004	FD	104	F9
005	FC	105	F8
006	FB	106	F7
007	FA	107	F6
008	F9	108	F5
009	F8	109	F4
00A	F7	10A	F3
00B	F6	10B	F2
00C	F5	10C	F1
00D	F4	10D	F0
00E	F3	10E	EF
00F	F2	10F	EE
010	F1	110	ED
011	F0	111	EC
012	EF	112	EB
•	•	•	•
•	•	•	•
•	•	•	•
0EE	13	1EE	0F
0EF	12	1EF	0E
0F0	11	1F0	0D
0F1	10	1F1	0C
0F2	0F	1F2	0B
0F3	0E	1F3	0A
0F4	0D	1F4	09
0F5	0C	1F5	08
0F6	0B	1F6	07
0F7	0A	1F7	06
0F8	09	1F8	05
0F9	08	1F9	04
0FA	07	1FA	03
0FB	06	1FB	02
0FC	05	1FC	01
0FD	04	1FD	(Note 2)
0FE	03	1FE	(Note 2)
0FF	02	1FF	(Note 2)

Notes: 1. The reciprocals of these numbers are too large to be represented in the selected format.
2. The reciprocals of these numbers are too small to be represented in normalized IEEE format.

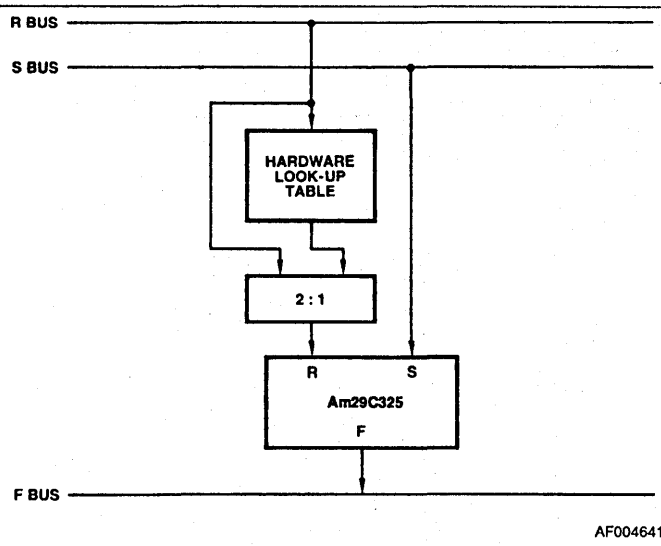


Figure C1. Adding a Hardware Look-Up Table to the Am29C325

The look-up table has two sections: a biased exponent look-up PROM, and a fraction look-up PROM. The seed-biased exponent look-up table is stored in a 512-by-8-bit PROM. This table consists of two sections: the DEC format section (which occupies addresses $000 - 0FF_{16}$), and the IEEE section (which occupies addresses $100 - 1FF_{16}$. The appropriate table will be selected automatically if address line A_8 is wired to the Am29C325's IEEE/$\overline{DEC}$ pin. The equations implemented by these table sections are:

DEC table: seed biased exponent
$= 257_{10}$ – input biased exponent

IEEE table: seed biased exponent
$= 253_{10}$ – input biased exponent

Table C1 lists the contents of this PROM.

The seed fraction look-up table is stored in one or more PROMs, the number of PROMs depending on the desired accuracy of the seed value. The hardware depicted in Figure C2 uses two 4K-by-8-bit PROMs to implement a fraction look-up table whose inputs are the 12 MSBs of the input argument's fraction. These PROMs output the 16 MSBs of the seed's fraction field — the remaining 7 bits of fraction are set to 0. The equation implemented in this table is:

$$\text{seed fraction} = \frac{2}{1 + \text{input fraction}} - 1$$

where the value of the input fraction falls in the range

$$0 \leqslant \text{input fraction} < 1$$

Note that the seed fraction must also be constrained to fall in the range

$$0 \leqslant \text{seed fraction} < 1$$

Therefore, if the input fraction is 0, the corresponding seed fraction stored in the table must be $.111...111_2$, not 1.0_2. The same seed fraction look-up table may be used for both IEEE and DEC formats. Table C2 contains a partial listing for the seed fraction look-up table shown in Figure C2.

TABLE C2. CONTENTS OF THE SEED FRACTION PROMS

Address (16)	Value of Input Fraction (10)	Value of Seed Fraction (10)	PROM Outputs (16)	
			$R_{22}-R_{15}$	$R_{14}-R_7$
000	0.0	0.9999999999 (see text)	FF	FF
001	0.0002441406	0.9995118370	FF	E0
002	0.0004882812	0.9990239150	FF	C0
003	0.0007324219	0.9985362280	FF	A0
004	0.0009765625	0.9980487790	FF	80
005	0.0012207031	0.9975615710	FF	60
006	0.0014648438	0.9970745970	FF	40
007	0.0017089844	0.9965878630	FF	20
008	0.0019531250	0.9961013650	FF	00
009	0.0021972656	0.9956151030	FE	E1
00A	0.0024414063	0.9951290800	FE	C0
00B	0.0026855469	0.9946432920	FE	A1
00C	0.0029296875	0.9941577400	FE	81
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
FF6	0.9975585938	0.0012221950	00	50
FF7	0.9978027344	0.0010998410	00	48
FF8	0.9980486750	0.0009775170	00	40
FF9	0.9982910156	0.0008552230	00	38
FFA	0.9985351563	0.0007329590	00	30
FFB	0.9987792969	0.0006107240	00	28
FFC	0.9990234375	0.0004885200	00	20
FFD	0.9992675781	0.0003663450	00	18
FFE	0.9995117188	0.0002442000	00	10
FFF	0.9997558594	0.0001220850	00	08

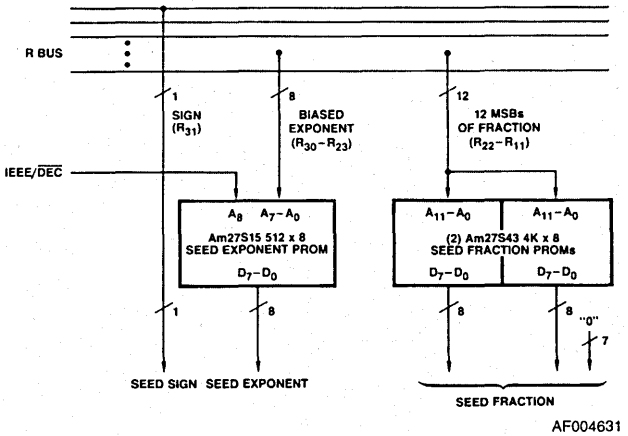


Figure C2. The Hardware Look-Up Table

With the hardware look-up table in place, the reciprocal of value B can be calculated with the following series of operations:

1) Place B on both the R and S buses. The 2 : 1 multiplexer at the output of the hardware look-up table should select the output of the look-up table (see Figure C3-A).

2) Load the seed value x_0 into register R and load B into register S. Select the R TIMES S operation (see Figure C3-B).

3) Load product $B*x_0$ into register F. Select the 2 MINUS S operation, and select register F as the input to the ALU S port (see Figure C3-C).

4) Load $2-B*x_0$ into register F. Select the R TIMES S operation and select register F as the input to the ALU S port (see Figure C3-D).

5) Load the value x_1 ($x_1 = x_0(2-B*x_0)$) into registers R and F. Select the R TIMES S operation (see Figure C3-E).

6) Repeat steps 3 through 5 until the result has the accuracy desired.

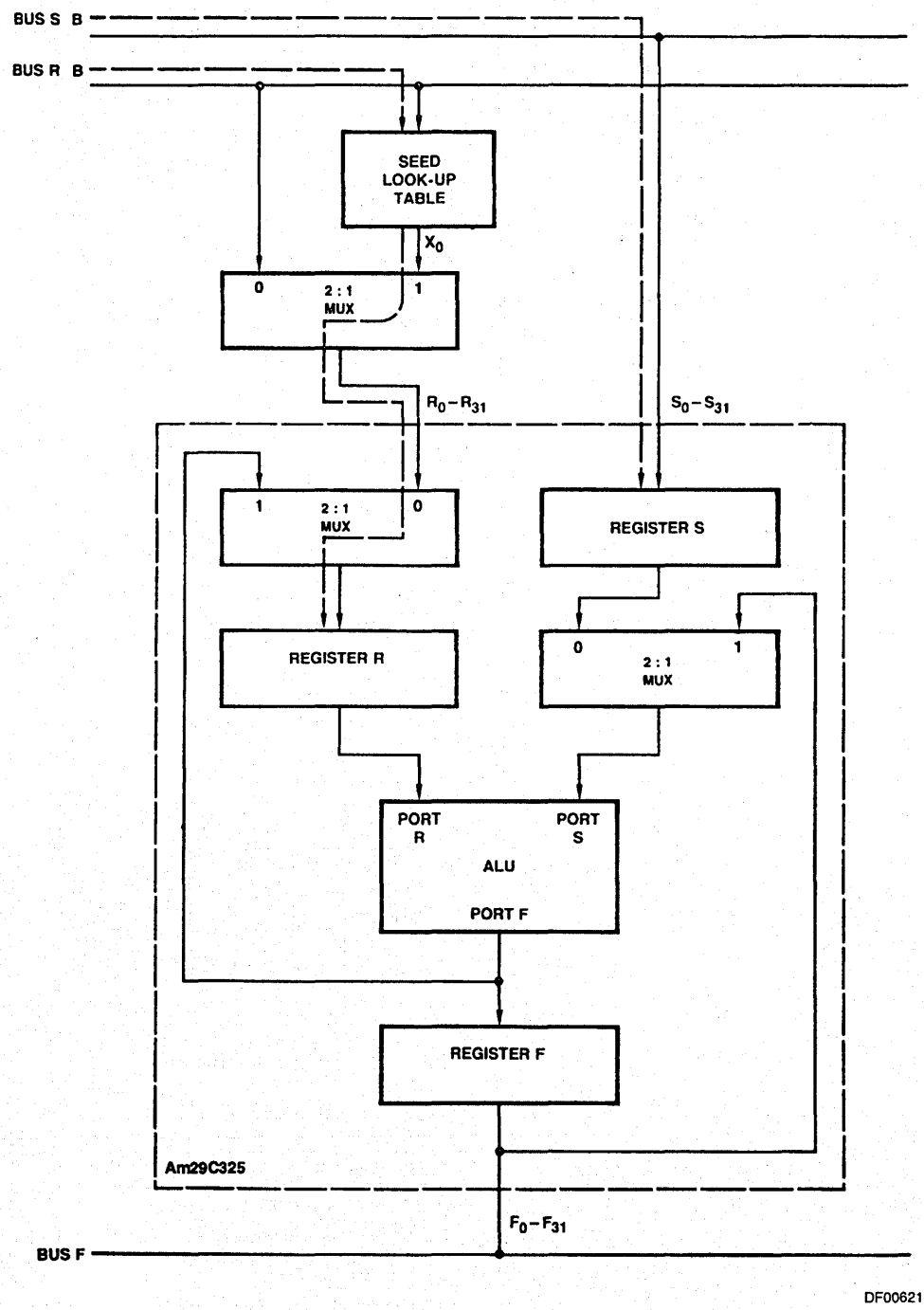


Figure C3-A. Data Flow for Step 1 of the Reciprocal Procedure

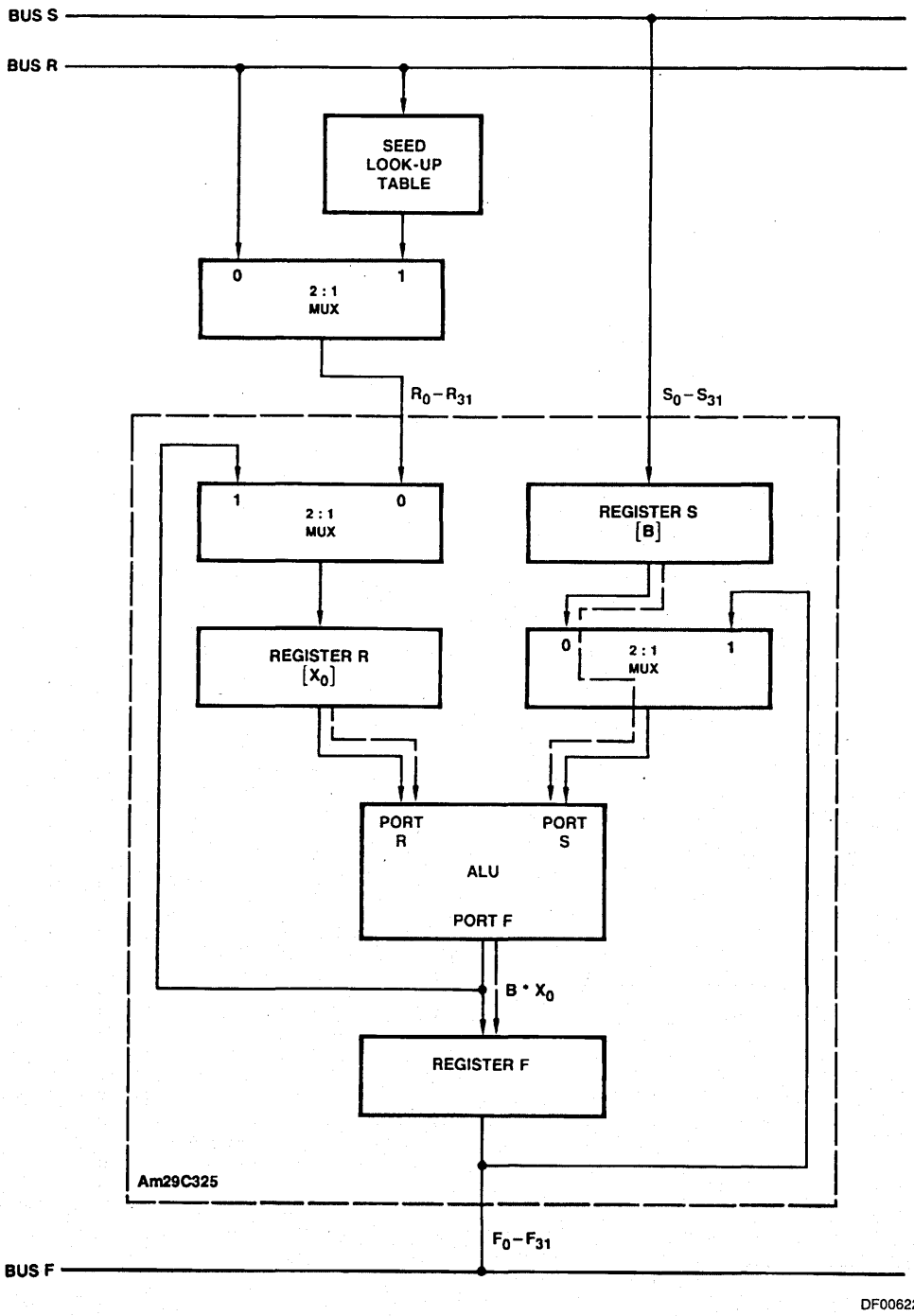


Figure C3-B. Data Flow for Step 2 of the Reciprocal Procedure

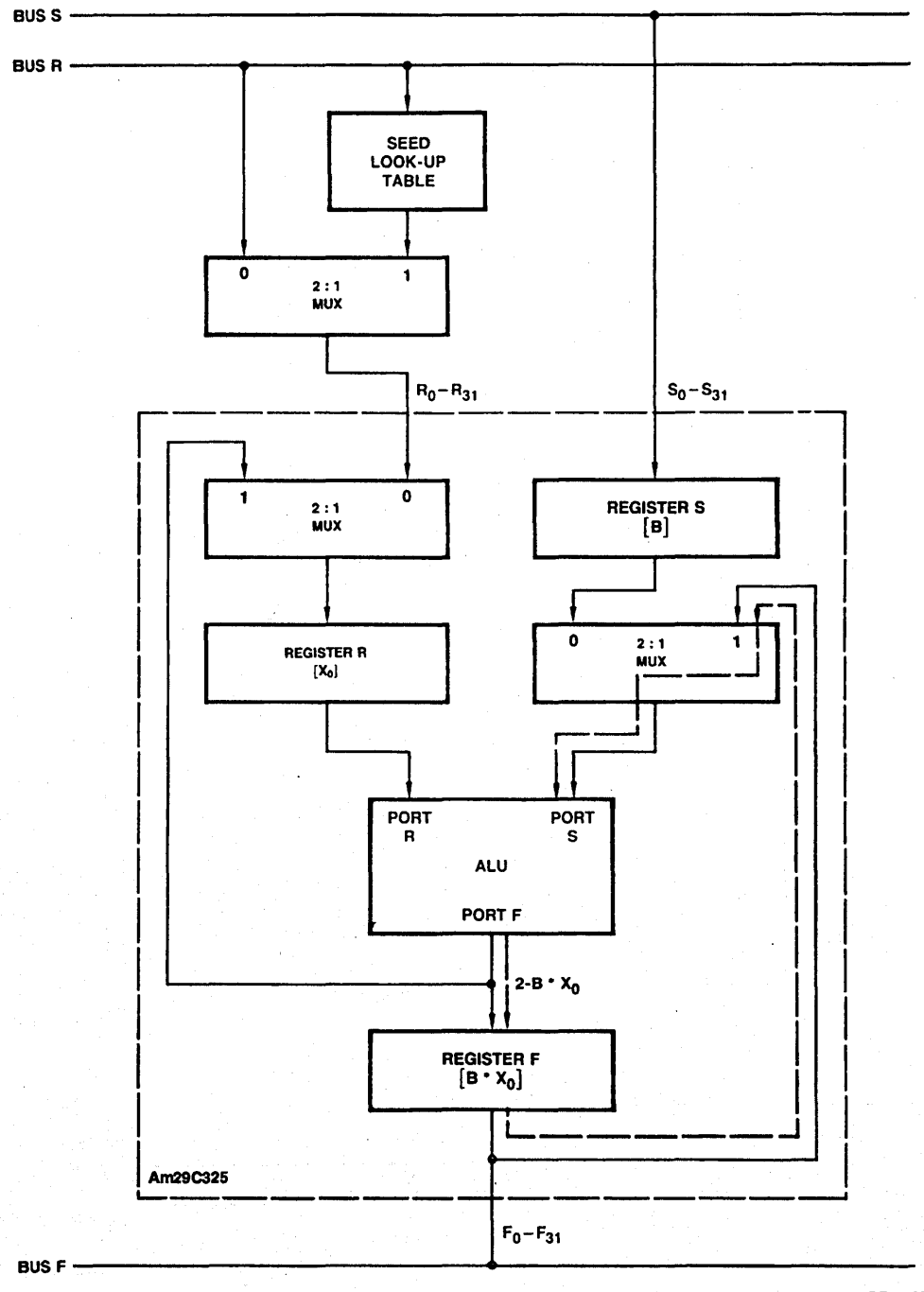


Figure C3-C. Data Flow for Step 3 of the Reciprocal Procedure

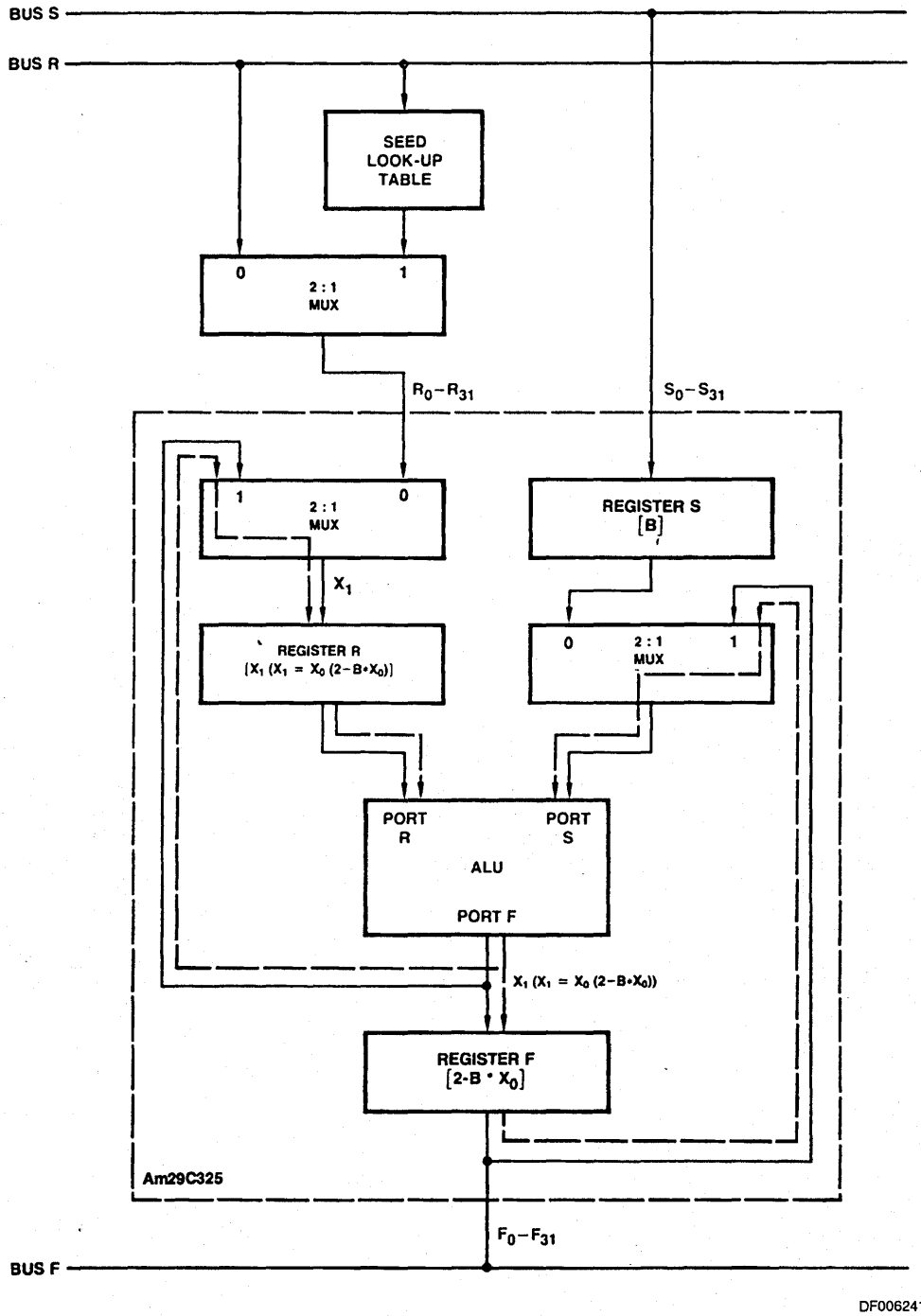


Figure C3-D. Data Flow for Step 4 of the Reciprocal Procedure

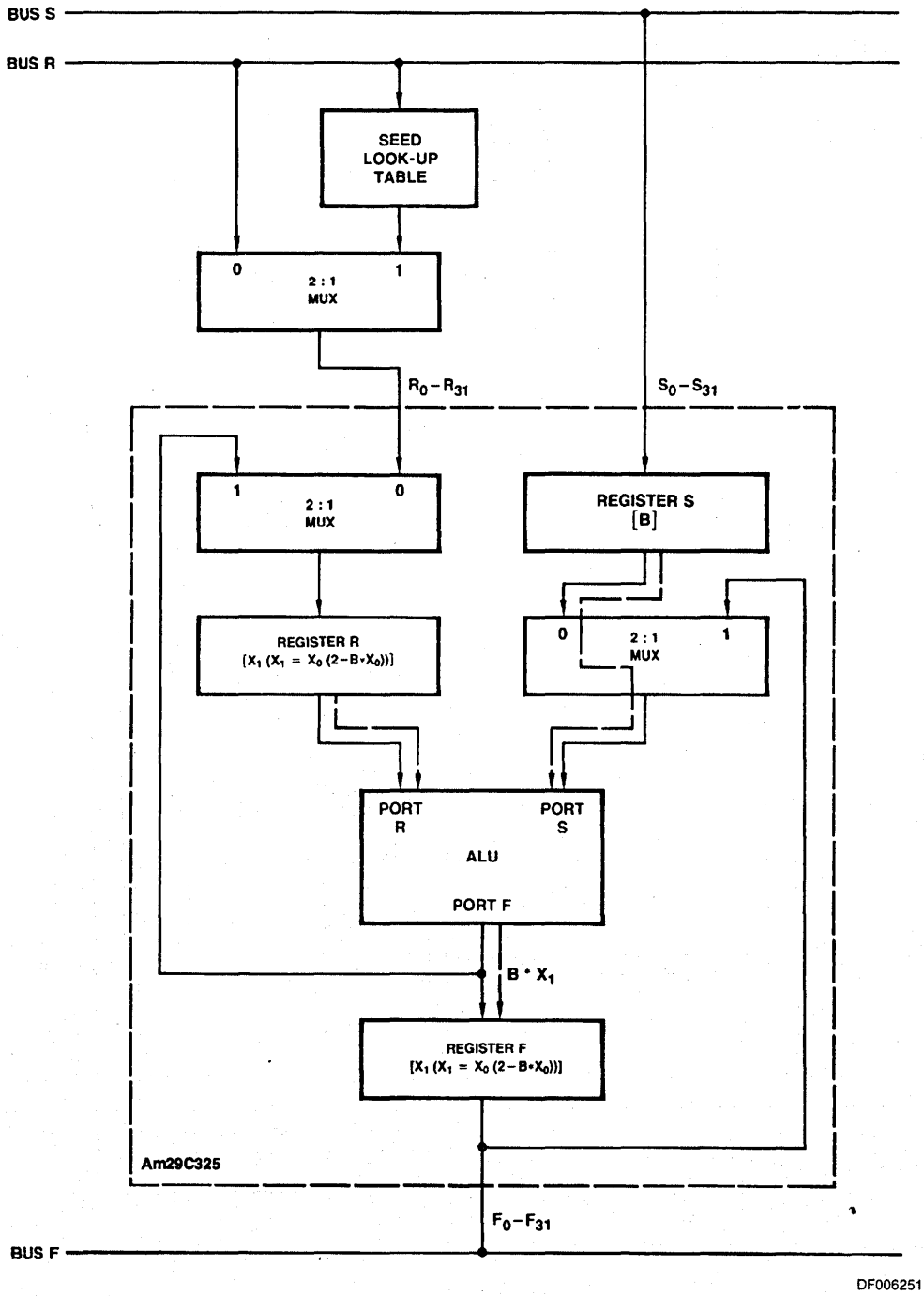


Figure C3-E. Data Flow for Step 5 of the Reciprocal Procedure

A tabular description of the operations above is given in Table C3. The following examples, performed in IEEE format, illustrate the process.

Example 1:

Find the reciprocal of 25.3.

Solution: The IEEE floating-point representation for 25.3 is $41CA6666_{16}$. The reciprocal process is begun by feeding this value to both the seed look-up table and port S. The look-up table produces the value $.03952789_{10}$ ($3D21E800_{16}$). The reciprocal is evaluated using the procedure described above; register values for each step are given in Table C4. The expected result, to the precision of the floating-point word, is $.03952569_{10}$ ($3D21E5B1_{16}$). In this case the expected result is produced after the first iteration. All subsequent iterations produce the same result and are therefore unnecessary.

TABLE C3. SEQUENCE OF EVENTS FOR EVALUATING RECIPROCALS

Clock Cycle	$I_0 - I_2$	I_3	I_4	$\overline{ENR}$	$\overline{ENS}$	$\overline{ENF}$	Register R	Register S	Register F	
1	Y	X	0	0	0	X	–	–	–	
2	R TIMES S	0	X	1	1	0	X_0	B	–	
3	2 MINUS S	1	X	1	1	0	X_0	B	$B*X_0$	First iteration
4	R TIMES S	1	1	0	1	0	X_0	B	$2-B*X_0$	
5	R TIMES S	0	X	1	1	0	$X_1(=X_0(2-B*X_0))$	B	$X_1(=X_0(2-B*X_0))$	
6	2 MINUS S	1	X	1	1	0	X_1	B	$B*X_1$	Second iteration
7	R TIMES S	1	1	0	1	0	X_1	B	$2-B*X_1$	
8	R TIMES S	0	X	1	1	0	$X_2(=X_1(2-B*X_1))$	B	$X_2(=X_1(2-B*X_1))$	

X = DON'T CARE

TABLE C4. INPUT BUS AND REGISTER VALUES FOR EXAMPLE 1

Clock Cycle	R Input	S Input	Register R	Register S	Register F	
1	3D21E800 (.03952789)	$41CA6666_{16}$ (25.3)	–	–	–	
2	–	–	$3D21E800_{16}$ (.03952789)	$41CA6666_{16}$ (25.3)	–	
3	–	–	$3D21E800_{16}$ (.03952789)	$41CA6666_{16}$ (25.3)	$3F8001D3_{16}$ (1.0000556)	
4	–	–	$3D21E800_{16}$ (.03952789)	$41CA6666_{16}$ (25.3)	$3F7FFC5A_{16}$ (.99984419)	
5	–	–	$3D21E5B1_{16}$ (.03952569)	$41CA6666_{16}$ (25.3)	$3D21E5B1_{16}$ (.03952569)	← Result of first iteration
6	–	–	$3D21E5B1_{16}$ (.03952569)	$41CA6666_{16}$ (25.3)	$3F7FFFFF_{16}$ (.99999994)	
7	–	–	$3D21E5B1_{16}$ (.03952569)	$41CA6666_{16}$ (25.3)	$3F800000_{16}$ (1.0)	
8	–	–	$3D21E5B1_{16}$ (.03952569)	$41CA6666_{16}$ (25.3)	$3D21E5B1_{16}$ (.03952569)	← Result of second iteration

Example 2:

Find the reciprocal of −0.4725.

Solution: The IEEE floating-point representation for −0.4725 is $BEF1EB85_{16}$. The reciprocal process is begun by feeding this value to both the seed look-up table and port S. The look-up table produces the value -2.11621094_{10} ($C0077000_{16}$). The reciprocal is evaluated using the procedure described above; register values for each step are given in Table C5. The expected result, to the precision of the floating-point word, is -2.116402_{10} ($C0077322_{16}$). In this case the expected result is produced after the first iteration. All subsequent iterations produce the same result and are therefore unnecessary.

TABLE C5. INPUT BUS AND REGISTER VALUES FOR EXAMPLE 2

Clock Cycle	R Input	S Input	Register R	Register S	Register F	
1	$C0077000_{16}$ (−2.1162109)	$BEF1EB85_{16}$ (−0.4725)	–	–	–	
2	–	–	$C0077000_{16}$ (−2.1162109)	$BEF1EB85_{16}$ (−0.4725)	–	
3	–	–	$C0077000_{16}$ (−2.1162109)	$BEF1EB85_{16}$ (−0.4725)	$3F7FFA14_{16}$ (0.99990963)	
4	–	–	$C0077000_{16}$ (−2.1162109)	$BEF1EB85_{16}$ (−0.4725)	$3F8002F6_{16}$ (1.0000904)	
5	–	–	$C0077322_{16}$ (−2.116402)	$BEF1EB85_{16}$ (−0.4725)	$C0077322_{16}$ (−2.116402)	← Result of first iteration
6	–	–	$C0077322_{16}$ (−2.116402)	$BEF1EB85_{16}$ (−0.4725)	$3F800000_{16}$ (1.0)	
7	–	–	$C0077322_{16}$ (−2.116402)	$BEF1EB85_{16}$ (−0.4725)	$3F800000_{16}$ (1.0)	
8	–	–	$C0077322_{16}$ (−2.116402)	$BEF1EB85_{16}$ (−0.4725)	$C0077322_{16}$ (−2.116402)	← Result of second iteration

APPENDIX D

SUMMARY OF FLAG OPERATION

Tables D1, D2, and D3 summarize flag operation for the IEEE mode, the DEC mode and for the IEEE-TO-DEC and DEC-TO-IEEE operations.

TABLE D1. FLAG SUMMARY FOR IEEE MODE

Operation	Condition(s)	INV	OVF	UNF	INE	ZER	NAN
Any operation listed in the IEEE Invalid Operations Table		H	L	L	L	L	H
R PLUS S R MINUS S R TIMES S 2 MINUS S	Input operands are finite $\|\text{rounded result}\| \geq 2^{128}$	L	H	L	H	L	L
R PLUS S R MINUS S R TIMES S	$0 < \|\text{rounded result}\| < 2^{-126}$	L	L	H	H	H	L
R PLUS S R MINUS S R TIMES S 2 MINUS S INT-TO-FP FP-TO-INT	Final result does not equal infinitely precise result	L	*	*	H	*	L
R PLUS S R MINUS S R TIMES S 2 MINUS S INT-TO-FP FP-TO-INT	Final result is zero	L	L	*	*	H	L
R PLUS S R MINUS S R TIMES S 2 MINUS S FP-TO-INT	Final result is a NAN	*	L	L	L	L	H

Notes: INV = Invalid operation flag
OVF = Overflow flag
UNF = Underflow flag
INE = Inexact flag
ZER = Zero flag
NAN = NAN flag
L = LOW
H = HIGH
* = State of flag depends on the input operands and the operation performed

TABLE D2. FLAG SUMMARY FOR DEC MODE

Operation	Condition(s)	INV	OVF	UNF	INE	ZER	NAN
FP-TO-INT	Rounded result $> 2^{31}-1$ or rounded result $< -2^{31}$	H	L	L	L	L	H
FP-TO-INT	Input is a DEC-reserved operand	L	L	L	L	L	H
R PLUS S R MINUS S R TIMES S 2 MINUS S	$\lvert$Rounded result$\rvert \geq 2^{127}$	L	H	L	H	L	H
R PLUS S R MINUS S R TIMES S	$0 < \lvert$rounded result$\rvert < 2^{-128}$	L	L	H	H	H	L
R PLUS S R MINUS S R TIMES S 2 MIMUS S INT-TO-FP FP-TO-INT	Final result does not equal infinitely precise result	L	*	*	H	*	*
R PLUS S R MINUS S R TIMES S 2 MINUS S INT-TO-FP FP-TO-INT	Final result is zero	L	L	*	*	H	L
R PLUS S R MINUS S R TIMES S 2 MINUS S FP-TO-INT	Final result is a DEC-reserved operand	*	*	L	L	L	H

Notes: INV = Invalid operation flag
OVF = Overflow flag
UNF = Underflow flag
INE = Inexact flag
ZER = Zero flag
NAN = NAN flag
L = LOW
H = HIGH
* = State of flag depends on the input operands and the operation performed

TABLE D3. FLAG SUMMARY FOR IEEE-TO-DEC AND DEC-TO-IEEE CONVERSIONS

Operation	Condition(s)	INV	OVF	UNF	INE	ZER	NAN
IEEE-TO-DEC	Input is a NAN	H	L	L	L	L	H
IEEE-TO-DEC	$\lvert$Input$\rvert \geq 2^{127}$	L	H	L	H	L	H
DEC-TO-IEEE	Input is a DEC-reserved operand	H	L	L	L	L	H
DEC-TO-IEEE	$0 < \lvert$rounded result$\rvert < 2^{-126}$	L	L	H	H	H	L
DEC-TO-IEEE IEEE-TO-DEC	Final result is zero	L	L	*	*	H	L

Notes: INV = Invalid operation flag
OVF = Overflow flag
UNF = Underflow flag
INE = Inexact flag
ZER = Zero flag
NAN = NAN flag
L = LOW
H = HIGH
* = State of flag depends on the input operands and the operation performed

Am29C327

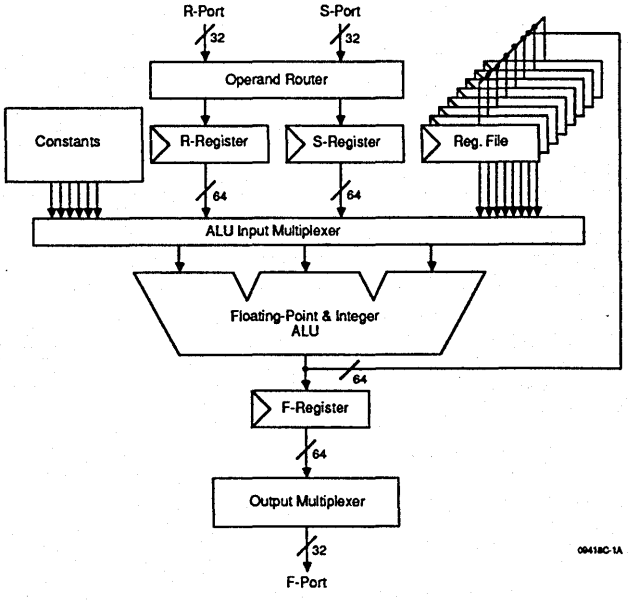
Double-Precision Floating-Point Processor

FINAL

DISTINCTIVE CHARACTERISTICS

- High-performance double-precision floating-point processor
- Comprehensive floating-point and integer instruction sets
- Single VLSI device performs single-, double-, and mixed-precision operations
- Performs conversions between precisions and between data formats
- Complies with seven industry-standard floating-point formats
  - ANSI/IEEE standard 754-1985; single- and double-precision
  - DEC F, DEC D, DEC G
  - IBM single- and double-precision
- Exact IEEE compliance for denormalized numbers with no speed penalty
- Eight-deep register file for intermediate results and on-chip 64-bit data path facilitates compound operations; e.g., Newton-Raphson division, sum-of-products, and transcendentals
- Supports pipelined or flow-through operation
- Fabricated with Advanced Micro Devices' 1.2 micron CMOS process

SIMPLIFIED BLOCK DIAGRAM



BD008080

GENERAL DESCRIPTION

The Am29C327 double-precision floating-point processor is a single VLSI device that implements an extensive floating-point and integer Instruction set. The three most widely used floating-point standards – IEEE, DEC, and IBM – are supported for both single- and double-precision operations. IEEE operations comply with the ANSI/IEEE Standard 754, with direct implementation of special features such as gradual underflow and handling of traps and denormalized numbers.

The Am29C327 consists of a 64-bit ALU, a 64-bit datapath, and a control unit. The ALU has three data input ports, and can perform single-operand, two-operand, and three-operand operations. The data path comprises two 64-bit input operand registers, an 8-by-64-bit register file for storage of intermediate results, three operand-selection multiplexers that provide for orthogonal selection of input operands, a 64-bit output register, and an output multiplexer that allows access to the 32 MSBs or 32 LSBs of the result data. Control signals determine the operation to be performed, the source of operands, the operand precisions, the rounding mode, and other aspects of device operation.

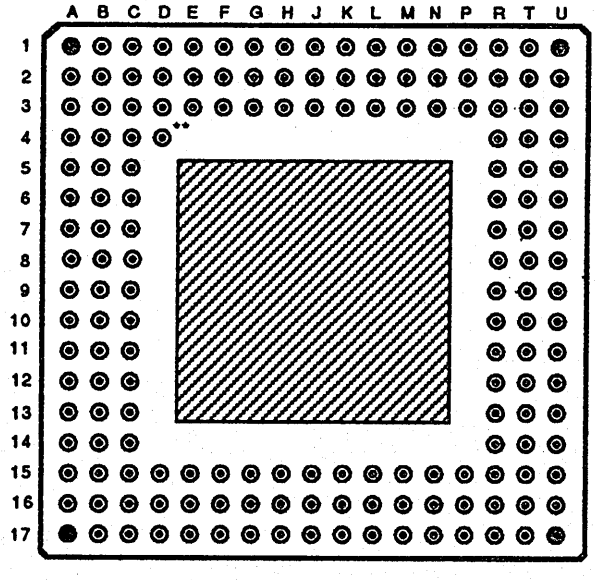
Operations can be performed in either of two modes: flow-through or pipelined. In the flow-through mode, the ALU is completely combinatorial; this mode is best suited for scalar operations. Pipelined mode divides the ALU into one or two pipeline stages, for use in vector operations, as often found in graphics or signal processing.

Fabricated with AMD's 1.2 micron CMOS technology, the Am29C327 is housed in a 169-lead pin-grid-array (PGA) package.

RELATED AMD PRODUCTS

Part No.	Description
Am29C10A	CMOS 12-Bit Sequencer
Am29C111	CMOS 16-Bit Sequencer
Am29C116	CMOS 16-Bit Microprocessor
Am29CPL141	CMOS 64 x 16 EPROM Programmable Controller
Am29CPL142	CMOS 128 x 16 EPROM Programmable Controller
Am29CPL144	CMOS 512 x 16 EPROM Programmable Controller
Am29CPL151	CMOS 64 x 16 EPROM Programmable Controller-Slim DIP
Am29CPL152	CMOS 128 x 16 EPROM Programmable Controller-Slim DIP
Am29CPL154	CMOS 512 x 16 EPROM Programmable Controller-Slim DIP
Am29C323	CMOS 32-Bit Parallel Multiplier
Am29C325	CMOS 32-Bit Floating-Point Processor
Am29C331	CMOS 16-Bit Sequencer
Am29C332	CMOS 32-Bit Arithmetic Logic Unit
Am29C334	CMOS Four-Port, Dual-Access Register File

CONNECTION DIAGRAM
169-Lead PGA*
Bottom View



09418C-5A
CD011601

*Pinout observed from pin side of package.
**Alignment pin (not connected internally).

PGA PIN DESIGNATIONS
(Sorted by Pin No.)

PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME
A-1	S_{31}	C-9	V_{CCO}	J-15	$TSEL_1$	R-10	V_{CC}
A-2	F_4	C-10	F_{20}	J-16	$TSEL_0$	R-11	CLOCK
A-3	F_6	C-11	V_{CCO}	J-17	$TSEL_2$	R-12	$\overline{ENF}$
A-4	F_8	C-12	GNDO	K-1	S_9	R-13	$\overline{ENI}$
A-5	F_{10}	C-13	F_{29}	K-2	S_{10}	R-14	FSEL
A-6	F_{12}	C-14	GNDO	K-3	GND	R-15	$RFSEL_1$
A-7	F_{14}	C-15	V_{CCO}	K-15	$QSEL_1$	R-16	$PSEL_3$
A-8	F_{16}	C-16	I_2	K-16	$QSEL_0$	R-17	$PSEL_0$
A-9	F_{18}	C-17	I_6	K-17	$TSEL_3$	T-1	R_{28}
A-10	F_{21}	D-1	S_{24}	L-1	S_8	T-2	R_{23}
A-11	F_{22}	D-2	S_{25}	L-2	S_7	T-3	R_{21}
A-12	F_{24}	D-3	S_{29}	L-3	S_6	T-4	R_{18}
A-13	F_{27}	D-15	I_0	L-15	GNDO	T-5	R_{16}
A-14	F_{28}	D-16	I_3	L-16	$QSEL_3$	T-6	R_{13}
A-15	F_{31}	D-17	I_8	L-17	$QSEL_2$	T-7	R_{10}
A-16	$\overline{SLAVE}$	E-1	S_{21}	M-1	S_5	T-8	R_7
A-17	I_1	E-2	S_{23}	M-2	S_4	T-9	R_5
B-1	S_{30}	E-3	S_{26}	M-3	S_2	T-10	R_3
B-2	F_1	E-15	I_4	M-15	V_{CCO}	T-11	R_0
B-3	F_3	E-16	I_7	M-16	$FLAG_2$	T-12	RM_1
B-4	F_5	E-17	I_9	M-17	$FLAG_1$	T-13	$\overline{ENS}$
B-5	F_7	F-1	S_{18}	N-1	S_3	T-14	$\overline{OES}$
B-6	F_9	F-2	S_{20}	N-2	S_1	T-15	$\overline{ENRF}$
B-7	F_{13}	F-3	S_{22}	N-3	R_{30}	T-16	$RFSEL_0$
B-8	F_{15}	F-15	V_{CC}	N-15	$FLAG_6$	T-17	$PSEL_1$
B-9	F_{17}	F-16	I_{10}	N-16	$FLAG_4$	U-1	R_{25}
B-10	F_{19}	F-17	I_{12}	N-17	$FLAG_3$	U-2	R_{22}
B-11	F_{23}	G-1	S_{15}	P-1	S_0	U-3	R_{19}
B-12	F_{25}	G-2	S_{17}	P-2	R_{29}	U-4	R_{17}
B-13	F_{26}	G-3	S_{19}	P-3	R_{26}	U-5	R_{15}
B-14	F_{30}	G-15	GND	P-15	$PSEL_2$	U-6	R_{14}
B-15	V_{CC}	G-16	I_{11}	P-16	SIGN	U-7	R_{11}
B-16	MSERR	G-17	$S/\overline{DF}$	P-17	$FLAG_5$	U-8	R_9
B-17	I_5	H-1	S_{13}	R-1	R_{31}	U-9	R_6
C-1	S_{27}	H-2	S_{14}	R-2	R_{27}	U-10	R_4
C-2	S_{28}	H-3	S_{16}	R-3	R_{24}	U-11	R_2
C-3	F_0	H-15	$S/\overline{DS}$	R-4	R_{20}	U-12	R_1
C-4	F_2	H-16	I_{13}	R-5	V_{CC}	U-13	RM_0
C-5	V_{CCO}	H-17	$S/\overline{DR}$	R-6	GND	U-14	RM_2
C-6	GNDO	J-1	S_{11}	R-7	R_{12}	U-15	$\overline{ENR}$
C-7	F_{11}	J-2	S_{12}	R-8	R_8	U-16	$\overline{OEF}$
C-8	GNDO	J-3	V_{CC}	R-9	GND	U-17	$RFSEL_2$

Note: Pin number D-4 = Alignment Pin
V_{CCO} and GNDO are power and ground pins for the output buffers.
V_{CC} and GND are power and ground pins for the rest of the logic.

PGA PIN DESIGNATIONS (Cont'd.)
(Sorted by Pin Name)

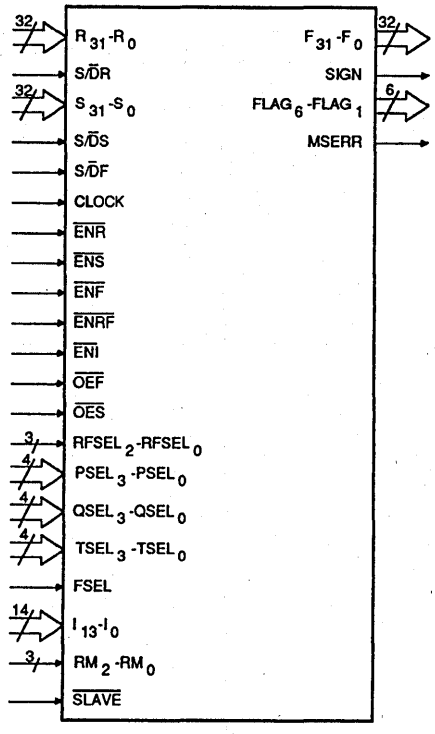
PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME
R-11	CLOCK	P-17	$FLAG_5$	T-9	R_5	K-1	S_9
R-12	$\overline{ENF}$	N-15	$FLAG_6$	U-9	R_6	K-2	S_{10}
R-13	$\overline{ENI}$	R-14	FSEL	T-8	R_7	J-1	S_{11}
U-15	$\overline{ENR}$	G-15	GND	R-8	R_8	J-2	S_{12}
T-15	$\overline{ENRF}$	K-3	GND	U-8	R_9	H-1	S_{13}
T-13	$\overline{ENS}$	R-6	GND	T-7	R_{10}	H-2	S_{14}
C-3	F_0	R-9	GND	U-7	R_{11}	G-1	S_{15}
B-2	F_1	C-6	GNDO	R-7	R_{12}	H-3	S_{16}
C-4	F_2	C-8	GNDO	T-6	R_{13}	G-2	S_{17}
B-3	F_3	C-12	GNDO	U-6	R_{14}	F-1	S_{18}
A-2	F_4	C-14	GNDO	U-5	R_{15}	G-3	S_{19}
B-4	F_5	L-15	GNDO	T-5	R_{16}	F-2	S_{20}
A-3	F_6	D-15	I0	U-4	R_{17}	E-1	S_{21}
B-5	F_7	A-17	I1	T-4	R_{18}	F-3	S_{22}
A-4	F_8	C-16	I2	U-3	R_{19}	E-2	S_{23}
B-6	F_9	D-16	I3	R-4	R_{20}	D-1	S_{24}
A-5	F_{10}	E-15	I4	T-3	R_{21}	D-2	S_{25}
C-7	F_{11}	B-17	I_5	U-2	R_{22}	E-3	S_{26}
A-6	F_{12}	C-17	I_6	T-2	R_{23}	C-1	S_{27}
B-7	F_{13}	E-16	I_7	R-3	R_{24}	C-2	S_{28}
A-7	F_{14}	D-17	I_8	U-1	R_{25}	D-3	S_{29}
B-8	F_{15}	E-17	I_9	P-3	R_{26}	B-1	S_{30}
A-8	F_{16}	F-16	I_{10}	R-2	R_{27}	A-1	S_{31}
B-9	F_{17}	G-16	I_{11}	T-1	R_{28}	G-17	$S/\overline{DF}$
A-9	F_{18}	F-17	I_{12}	P-2	R_{29}	H-17	$S/\overline{DR}$
B-10	F_{19}	H-16	I_{13}	N-3	R_{30}	H-15	$S/\overline{DS}$
C-10	F_{20}	B-16	MSERR	R-1	R_{31}	P-16	SIGN
A-10	F_{21}	U-16	$\overline{OEF}$	T-16	$RFSEL_0$	A-16	$\overline{SLAVE}$
A-11	F_{22}	T-14	$\overline{OES}$	R-15	$RFSEL_1$	J-16	$TSEL_0$
B-11	F_{23}	R-17	$PSEL_0$	U-17	$RFSEL_2$	J-15	$TSEL_1$
A-12	F_{24}	T-17	$PSEL_1$	U-13	RM_0	J-17	$TSEL_2$
B-12	F_{25}	P-15	$PSEL_2$	T-12	RM_1	K-17	$TSEL_3$
B-13	F_{26}	R-16	$PSEL_3$	U-14	RM_2	B-15	VCC
A-13	F_{27}	K-16	$QSEL_0$	P-1	S_0	F-15	VCC
A-14	F_{28}	K-15	$QSEL_1$	N-2	S_1	J-3	VCC
C-13	F_{29}	L-17	$QSEL_2$	M-3	S_2	R-5	VCC
B-14	F_{30}	L-16	$QSEL_3$	N-1	S_3	R-10	VCC
A-15	F_{31}	T-11	R_0	M-2	S_4	C-5	VCCO
M-17	$FLAG_1$	U-12	R_1	M-1	S_5	C-9	VCCO
M-16	$FLAG_2$	U-11	R_2	L-3	S_6	C-11	VCCO
N-17	$FLAG_3$	T-10	R_3	L-2	S_7	C-15	VCCO
N-16	$FLAG_4$	U-10	R_4	L-1	S_8	M-15	VCCO

Note: Pin number D-4 = Alignment Pin

V_{CCO} and GNDO are power and ground pins for the output buffers.

V_{CC} and GND are power and ground pins for the rest of the logic.

LOGIC SYMBOL



09418C-6A

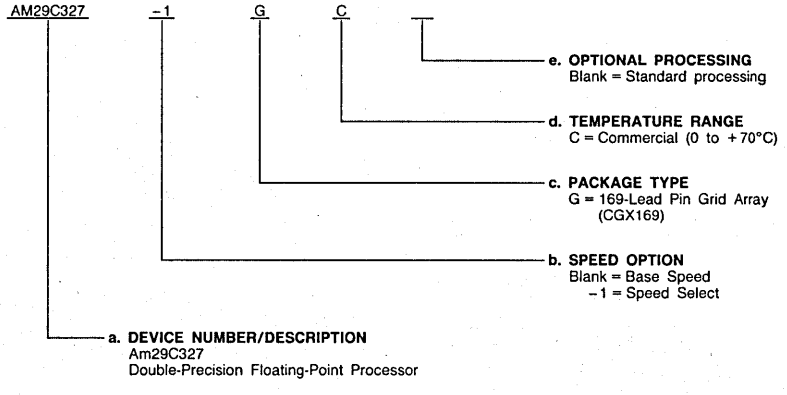
LS003281

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM29C327	GC
AM29C327-1	

Valid Combinations

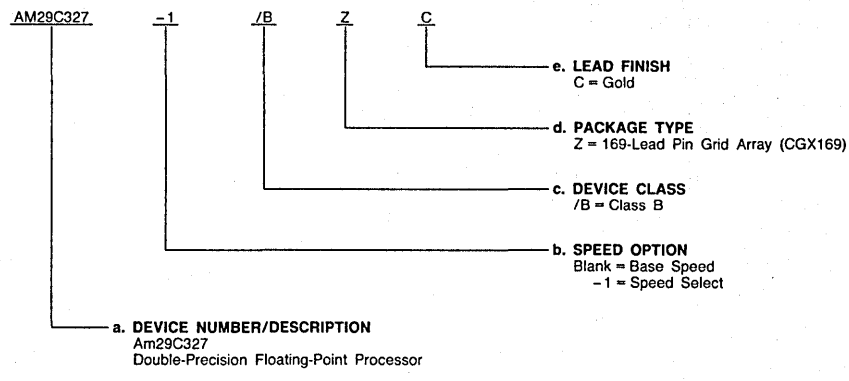
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

2

MILITARY ORDERING INFORMATION

APL Products

AMD standard products for Aerospace and Defence applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM29C327	/BZC
AM29C327-1	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Group A Tests

Group A tests consist of Subgroups
1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

CLOCK Clock (Input)
Clock input to all registers. The Am29C327 is fully static — no data is lost from the internal registers if the clock is stopped for an extended period.

$\overline{ENF}$ F-Register Enable (Input; Active LOW)
When $\overline{ENF}$ is HIGH, the contents of the F-register are static. When $\overline{ENF}$ is LOW, the 64-bit ALU output is clocked into the F-register on the next LOW-to-HIGH transition of the CLOCK input. As described in the Mode Register Description section, the F-register can be made transparent by setting the mode register bit M17 appropriately, in which case $\overline{ENF}$ has no effect. This input is not clocked into the instruction register, and must be valid at the LOW-to-HIGH CLOCK transition on which the desired data is to be clocked into the F-register.

$\overline{ENI}$ Instruction Register Enable (Input; Active LOW)
When $\overline{ENI}$ is HIGH, the contents of the instruction register are static. When $\overline{ENI}$ is LOW, the 30-bit instruction word, comprising the fields $PSEL_{3-0}$, $QSEL_{3-0}$, $TSEL_{3-0}$, RM_{2-0}, $S/\overline{DF}$ and I_{13-0}, is clocked into the instruction register on the next LOW-to-HIGH transition of the CLOCK input. This input is not clocked into the instruction register and must be valid at the LOW-to-HIGH CLOCK transition on which the desired data is to be clocked into the instruction register.

$\overline{ENR}$ R-Register Enable (Input; Active LOW)
When $\overline{ENR}$ is HIGH, the contents of the R-register are static. When $\overline{ENR}$ is LOW, a new 64-bit operand, together with the precision control input $S/\overline{DR}$, is clocked into the 65-bit R-register on the next LOW-to-HIGH transition of the CLOCK input. As described in the Input Modes section, the user can select from eight different input modes, as appropriate for the system environment. This input is not clocked into the instruction register and must be valid at the LOW-to-HIGH CLOCK transition on which the desired data is to be clocked into the R-register.

$\overline{ENRF}$ Register File Enable (Input; Active LOW)
When $\overline{ENRF}$ is HIGH, the contents of the register file are static. When $\overline{ENRF}$ is LOW, the 64-bit ALU result, together with a "tag" indicating its precision, is clocked into one of the 65-bit registers RF7 to RF0 on the next LOW-to-HIGH transition of the CLOCK input. The inputs $RFSEL_{2-0}$ determine which of the eight registers in the register file is the destination for the ALU result and its precision tag. This input is not clocked into the instruction register and must be valid at the LOW-to-HIGH CLOCK transition on which the desired data is to be clocked into the register file.

$\overline{ENS}$ S-Register Enable (Input; Active LOW)
When $\overline{ENS}$ is HIGH, the contents of the S-register are static. When $\overline{ENS}$ is LOW, a new 64-bit operand, together with the precision control input $S/\overline{DS}$, is clocked into the 65-bit S-register on the next LOW-to-HIGH transition of the CLOCK input. As described in the Input Modes section, the user can select from eight different input modes, as appropriate for the system environment. This input is not clocked into the instruction register and must be valid at the LOW-to-HIGH CLOCK transition on which the desired data is to be clocked into the S-register.

F_{31-0} Output Bus (Bidirectional)
The 32-bit output bus is bidirectional to support Master/Slave checking.

$FLAG_{6-1}$ Flag Outputs (Bidirectional)
The six flag outputs $FLAG_6$-$FLAG_1$ report the status of the previous ALU operation. The outputs are bidirectional to support Master/Slave checking.

FSEL Output Multiplexer Control (Input)
When FSEL is HIGH, the most-significant 32 bits of the 64-bit F-register are connected to the drivers on the F_{31-0} output bus. When FSEL is LOW, the least-significant 32 bits of the 64-bit F-register are connected to the drivers on the F_{31-0} output bus. The state of this input pin may be changed at twice the rate of the CLOCK input, to allow a full 64-bit result to be output from the Am29C327 in a single clock cycle. This input is not clocked into the instruction register and must be valid while the required data is accessed via the output bus.

I_{13-0} ALU Instruction (Input)
I_{13-0} determine the ALU instruction to be executed in the next cycle. The inputs are clocked into the instruction register under the control of the $\overline{ENI}$ input.

MSERR Master/Slave Error (Output)
The MSERR output is asserted (HIGH) whenever a Master/Slave error is detected on any enabled output, i.e. F_{31-0} if $\overline{OEF}$ is LOW, SIGN and $FLAG_{6-1}$ if $\overline{OES}$ is LOW.

$\overline{OEF}$ F-Output Enable (Input; Active LOW)
When $\overline{OEF}$ is HIGH, the F_{31-0} output bus assumes a high-impedance state. When $\overline{OEF}$ is LOW (and $\overline{SLAVE}$ is HIGH, specifying "Master" mode), the F_{31-0} output drivers are enabled. This input is not clocked into the instruction register and must be valid at all times.

$\overline{OES}$ Flag Output Enable (Input; Active LOW)
When $\overline{OES}$ is HIGH, the 7 outputs $FLAG_{6-1}$ and SIGN assume a high-impedance state. When $\overline{OES}$ is LOW (and $\overline{SLAVE}$ is HIGH, specifying "Master" mode), the output drivers for the $FLAG_{6-1}$ and SIGN outputs are enabled. This input is not clocked into the instruction register and must be valid at all times.

$PSEL_{3-0}$ P-Input Multiplexer Control (Input)
The $PSEL_{3-0}$ inputs control the P-Input Multiplexer, selecting the source of operands for the P-Input of the ALU. The $PSEL_{3-0}$ inputs are clocked into the instruction register under the control of the $\overline{ENI}$ input.

$QSEL_{3-0}$ Q-Input Multiplexer Control (Input)
The $QSEL_{3-0}$ inputs control the Q-Input Multiplexer, selecting the source of operands for the Q-Input of the ALU. The $QSEL_{3-0}$ inputs are clocked into the instruction register under the control of the $\overline{ENI}$ input.

R_{31-0} R-Input Bus (Input)
The 32-bit R-Input bus, R_{31-0}, is used to load operands into one or both of the input registers, R and S. It is also used to load data into the 32-bit mode register.

$RFSEL_{2-0}$ Register File Destination Select (Input)
The $RFSEL_{2-0}$ inputs select which of the registers RF7 through RF0 is the destination for the ALU result. Operands are clocked into the selected register only when the $\overline{ENRF}$ input is LOW. This input is not clocked into the instruction register and must be valid at the LOW-to-HIGH CLOCK transition on which the desired data is to be clocked into the register file.

RM_{2-0} Rounding Mode Select (Input)
The RM_{2-0} inputs select which of the six available rounding modes is to be applied to the next ALU operation. Rounding is discussed in Appendix B. The RM_{2-0} inputs are clocked into the instruction register under the control of the $\overline{ENI}$ input.

S_{31-0} S-Input Bus (Input)
The 32-bit S-Input bus, S_{31-0}, is used to load operands into one or both of the input registers, R and S.

S/$\overline{DF}$ F-Precision Control (Input)
When S/$\overline{DF}$ is HIGH, the next ALU operation produces a single-precision (32-bit) result. When S/$\overline{DF}$ is LOW, the next ALU operation produces a double-precision (64-bit) result. The S/$\overline{DF}$ input is clocked into the instruction register under the control of the $\overline{ENI}$ input.

S/$\overline{DR}$ R-Precision Control (Input)
When S/$\overline{DR}$ is HIGH, the data clocked into the R-register is treated as single-precision (32-bit) by the processor. When S/$\overline{DR}$ is LOW, the data clocked into the R-register is treated as double-precision (64-bit) by the processor. The S/$\overline{DR}$ input is clocked into the 65th bit of the R-register as the "precision tag" for the R-operand, under the control of the $\overline{ENR}$ input.

S/$\overline{DS}$ S-Precision Control (Input)
When S/$\overline{DS}$ is HIGH, the data clocked into the S-register is treated as single-precision (32-bit) by the processor. When S/$\overline{DS}$ is LOW, the data clocked into the S-register is treated as double-precision (64-bit) by the processor. The S/$\overline{DS}$ input is clocked into the 65th bit of the S-register as the "precision tag" for the S-operand, under the control of the $\overline{ENS}$ input.

SIGN Sign Flag (Bidirectional)
If the result of the previous ALU operation was negative, the SIGN output is HIGH (provided that $\overline{OES}$ is LOW). If the result of the previous ALU operation was not negative, the SIGN output is LOW (provided that $\overline{OES}$ is LOW). The output is bidirectional to support Master/Slave checking.

$\overline{SLAVE}$ Master/Slave Mode Select (Input; Active LOW)
When $\overline{SLAVE}$ is HIGH, the "Master" mode of operation is selected. When $\overline{SLAVE}$ is LOW, the "Slave" mode of operation is selected and all outputs except MSERR are disabled (high-impedance). This input is not clocked into the instruction register and must be valid at all times.

$TSEL_{3-0}$ T-Input Multiplexer Control (Input)
The $TSEL_{3-0}$ inputs control the T-Input Multiplexer, selecting the source of operands for the T-Input of the ALU. The $TSEL_{3-0}$ inputs are clocked into the instruction register under the control of the $\overline{ENI}$ input.

V_{CC} GND Power
Power supply pins for the internal logic.

V_{CCO} GNDO Power
Power supply pins for the output buffers.

FUNCTIONAL DESCRIPTION

Overview

The Am29C327 is a high-performance, single-chip, double-precision floating-point processor.

Architecture

The Am29C327 comprises a high-speed ALU, a 64-bit data path, and control circuitry.

The core of the Am29C327 is a 64-bit floating-point/integer ALU. This ALU takes operands from three 64-bit input ports and performs the selected operation, placing the result on a 64-bit output port. Thirteen ALU flags report operation status via the $FLAG_{6-1}$ and SIGN outputs. The ALU is completely combinatorial for minimum latency; optional pipelining is available to boost throughput for array operations.

The data path consists of the 32-bit input buses R and S; two 64-bit input operand registers; an 8-by-64-bit register file for storage of intermediate results; three operand-selection multiplexers that provide for orthogonal selection of input operands; a 64-bit output register; and an output multiplexer that permits the selection of 32 MSBs, or 32 LSBs of data. Input operands enter the processor through the R and S buses, and are then demultiplexed and buffered for subsequent storage in registers R and S. The operand selection multiplexers route the operands to the ALU. Operation results are stored in register F, and leave the device on the 32-bit output bus F. The results can also be stored in the register file for use in subsequent operations.

Instruction Set

The Am29C327 implements 58 arithmetic and logical instructions. Thirty-five instructions operate on floating-point numbers; these instructions fall into the following categories:

- Addition/subtraction
- Multiplication
- Multiplication-accumulation
- Comparison
- Selecting the maximum or minimum of two numbers
- Rounding to integral value
- Absolute value, negation, pass
- Reciprocal seed generation
- Conversion between any of the supported floating-point formats
- Conversion of a floating-point number to an integer format, with or without a scale factor

By concatenating these operations, the user can also perform division, square-root extraction, polynomial evaluation, and other functions not implemented directly.

Twenty-two instructions operate on integers, and belong to the following general categories:

- Addition/subtraction
- Multiplication
- Comparison
- Selecting the maximum or minimum of two numbers
- Absolute value, negation, pass
- Logical operations; e.g., AND, OR, XOR, NOT
- Arithmetic, logical, and funnel shifts
- Conversion between single- and double-precision integer formats
- Conversion of an integer number to a floating-point format, with or without a scale factor

One special instruction is provided to move data from the P-Port to the F-Port, and another to load the mode register.

Mixed-Precision Operations

All Am29C327 instructions, floating-point or integer, can be performed with either single- or double-precision operands. In addition, the user can elect to mix precisions within an operation. All operations are performed in double-precision internally; the user specifies the precisions of the input operands and the required precision for the output operand. The necessary precision conversions are made in concert with the selected operation, with no additional cycle-time overhead.

I/O Modes

The Am29C327 supports eight I/O modes that afford flexible interface to a variety of 32- and 64-bit systems.

Fault Detection Features

The Am29C327 contains special comparison hardware to allow the operation of two processors in parallel, with one processor (the slave) checking the results produced by the other (the master). This feature is of particular importance in the design of high-reliability systems.

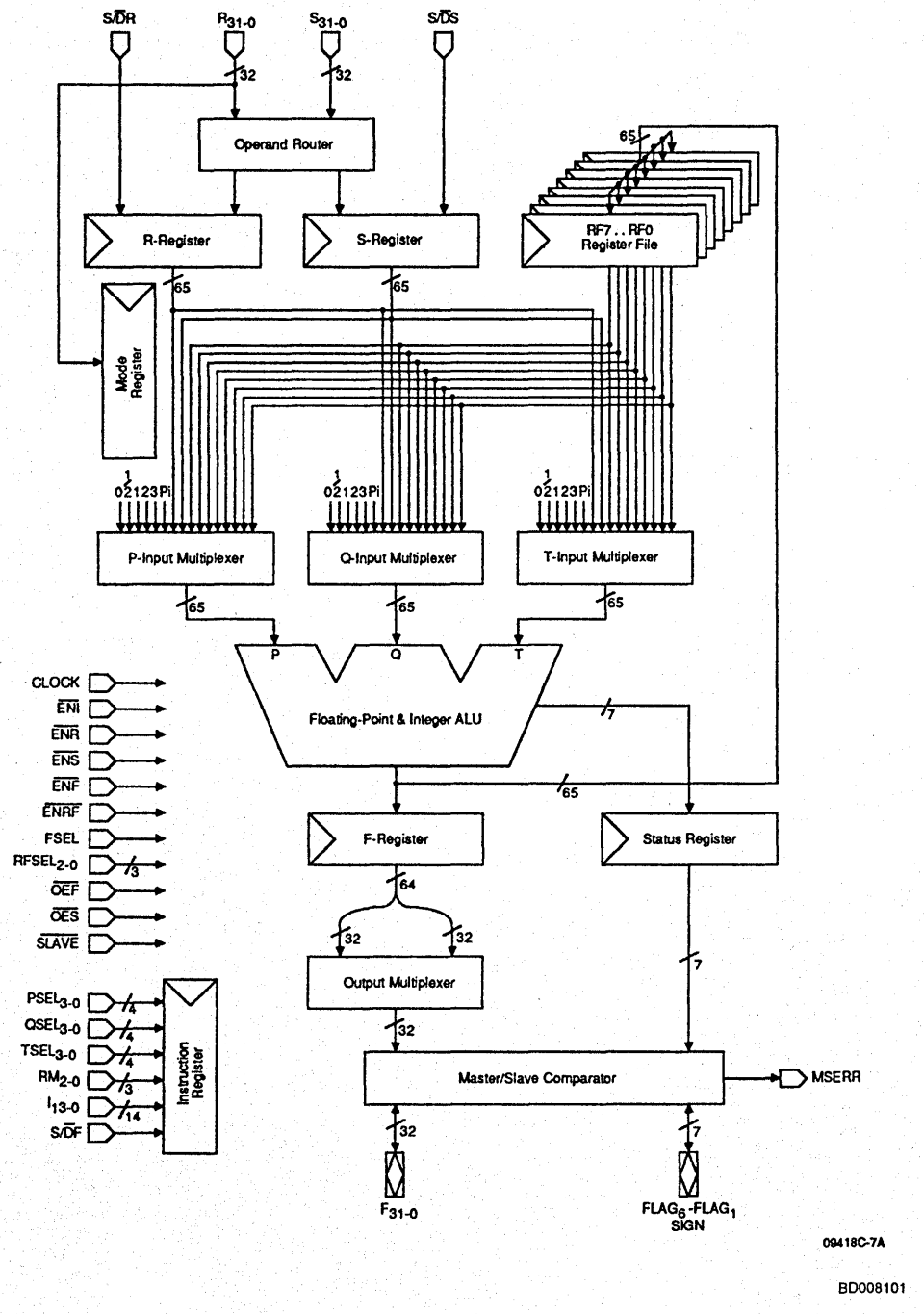


Figure 1. Block Diagram – Am29C327 Double-Precision Floating-Point Processor

Block Diagram Description

A block diagram of the Am29C327 is shown in Figure 1. The Am29C327 comprises the input registers, the operand selection multiplexers, the instruction register, the ALU, the output register/the register file, the status register, the output selection multiplexer, the mode register, and the master/slave comparator.

Input Registers/Input Modes

Operands are loaded into the processor through the R and S buses, and are then demultiplexed and buffered for subsequent storage in the 65-bit registers R and S. Input operands may be either single-precision (32-bit) or double-precision (64-bit) as specified by S/$\overline{D}$R and S/$\overline{D}$S. Accompanying the input registers are two 32-bit temporary registers, that allow for the overlapping of operand transfers and ALU operations. This arrangement of temporary registers and demultiplexers permits data and corresponding precision bit S/$\overline{D}$R or S/$\overline{D}$S to be loaded into the 65-bit R register and 65-bit S register via one of the eight input modes:

1. 32-bit-bus, double-cycle, LSWs first
2. 32-bit-bus, double-cycle, MSWs first
3. 32-bit-bus, single-cycle, LSWs first
4. 32-bit-bus, single-cycle, MSWs first
5. 64-bit-bus, double-cycle, R first
6. 64-bit-bus, double-cycle, S first
7. 64-bit-bus, single-cycle, R first
8. 64-bit-bus, single-cycle, S first

These modes are described in detail in the Input Modes section.

Operand Selection Multiplexers

The operand selection multiplexers route operands to the ALU. These multiplexers, as well as selecting operands from input registers R and S and register file locations RF7 – RF0, also have access to a set of constants (0, 0.5, 1, 2, 3, Pi). These constants are double-precision preprogrammed numbers for use in ALU operations, and are automatically provided in the appropriate floating-point or integer format.

Instruction Register

The instruction register stores a 32-bit word specifying the current processor operation. Included in the instruction word are fields that specify the P, Q, and T multiplexer selects, the rounding modes; the core operation to be performed by the ALU; sign-change controls for ALU input and result operands; and the single/double-precision control for the output operand. The multiplexer select controls and the instruction word are described in detail in the Instruction Set section; Rounding modes are described in Appendix B.

ALU

The ALU is a combinatorial arithmetic/logic unit that performs a large repertoire of floating-point and integer operations. The ALU has three operand inputs. Some operations require a single input operand, for example, conversion operations. Others, such as addition and multiplication, require two input operands. The multiplication-accumulation and funnel shift operations require three input operands. Many ALU arithmetic operations allow for the independent control of operand signs, thus greatly increasing the number of arithmetic expressions that can be evaluated in a single ALU pass.

The ALU can be configured in either a flow-through mode, for which the ALU is completely combinatorial, or a pipelined mode, for which ALU operations incur one or two pipeline delays, but which results in a higher throughput than flow-through mode.

A detailed description of ALU operations appears in the Instruction Set section.

Output Register/Register File

The results of the operations performed by the ALU are stored in the 64-bit output register F. Results can also be stored in the 8-by-64-bit register file for use in subsequent operations. Each register file location contains a 65th bit indicating the precision of the operand stored in that location, thus permitting the ALU to correctly process the operand in subsequent operations.

Status Register

The status register is a 7-bit register that stores flags pertaining to the most recently performed operation. A detailed description is provided in the Instruction Set section.

Output Multiplexer

The output multiplexer routes operation results to the F bus. This multiplexer selects either the 32 MSBs or the 32 LSBs of the data stored in the output register.

Master/Slave Comparator

Each Am29C327 output signal has associated logic that compares that signal with the signal that the processor is providing internally to the output driver; any discrepancies are indicated by assertion of signal MSERR.

For a single processor, this output comparison detects short circuits in output signals or defective output drivers, but does not detect open circuits. It is possible to connect a second processor in parallel with the first, with the second processor's outputs disabled by assertion of signal $\overline{\text{SLAVE}}$. The second processor detects open-circuit signals, as well as providing a check of the outputs of the first.

Mode Register

The mode register contains processor control parameters that are changed infrequently. The 32-bit mode word is loaded into the register via the R bus. A detailed description of the mode register is provided in the Mode Register Description section.

Mode Register Description

The 32-bit mode register contains parameters specifying the overall operating mode of the Am29C327. These parameters are typically not changed on an instruction-by-instruction basis. The register is loaded from the R-port R_{31-0}, using the "Load Mode Register" instruction. This section provides a comprehensive explanation of the function of each field within the register.

Bits M31-M21

Reserved for factory test and future upgrades. Must be set to logic 0.

Bits M20-M19 — Pipeline Mode Select

This field determines whether the ALU operates in flow-through mode or pipelined mode, and specifies whether pipelined multiply-accumulate operations are single-pipelined or double-pipelined:

M20	M19	
0	X	The ALU operates in flow-through mode for all operations — all pipeline registers are transparent.
1	0	The ALU operates in single-pipelined mode for all operations.
1	1	The ALU operates in double-pipelined mode for multiply-accumulate operations. The ALU operates in single-pipelined mode for all other operations.

A detailed description of the pipeline modes is contained in the Pipelining of Operations section.

Bit M18 — Status Register Feedthrough Enable

If M18 is HIGH, the 7-bit status register is made transparent and operates in flow-through mode. This mode is generally used when it is necessary to minimize the overall latency of the processor. If M18 is LOW, the status register operates in clocked mode, status information being clocked into the register from the ALU on every rising edge of the clock input. This mode is generally used when it is necessary to maximize the overall throughput of the processor.

Bit M17 – F-Register Feedthrough Enable

If M17 is HIGH, the 64-bit F-register is made transparent and operates in flow-through mode. This mode is generally used when it is necessary to minimize the overall latency of the processor. If M17 is LOW, the F-register operates in clocked mode, ALU results being clocked into the register from the ALU on every rising edge of the clock input, provided that $\overline{ENF}$ is LOW. This mode is generally used when it is necessary to maximize the overall throughput of the processor.

Bits M16-M14 — Input Mode Select

This field determines which of the eight available modes is used to input operands to the R-register and S-register:

M16	M15	M14	Input Mode
0	0	0	32-bit bus, single-cycle, LSW first.
0	0	1	32-bit bus, single-cycle, MSW first.
0	1	0	32-bit bus, double-cycle, LSW first.
0	1	1	32-bit bus, double-cycle, MSW first.
1	0	0	64-bit bus, single-cycle, R first.
1	0	1	64-bit bus, single-cycle, S first.
1	1	0	64-bit bus, double-cycle, R first.
1	1	1	64-bit bus, double-cycle, S first.

A detailed description of the input modes is contained in the Input Mode section.

Bits M13-M12 — Integer Multiplication Format Adjust

This field determines the output format for integer multiplication operations, selecting either the MSBs or the LSBs of the product and specifying whether or not format-adjusting (i.e., shifting the product one place left before the MSBs/LSBs selection) is performed:

M13	M12	Integer Multiplication Output
0	0	LSBs selected, no format-adjust performed.
0	1	LSBs selected, format-adjust performed.
1	0	MSBs selected, no format-adjust performed.
1	1	MSBs selected, format-adjust performed.

This field has no effect on operations other than integer multiplications.

Bit M11 — Integer Multiplication Signed/Unsigned

If M11 is HIGH, the input operands for integer multiplications are treated as two's-complement (signed) operands and two's-complement multiplications are performed. If M11 is LOW, the input operands for integer multiplications are treated as unsigned operands and unsigned multiplications are performed.

This bit has no effect on operations other than integer multiplications.

Bit M10

Reserved for factory test and future upgrades. Must be set to logic 0.

Bit M9 — IBM Underflow Mask Enable

If M9 is HIGH, underflowed results in IBM format are output with the (underflowed) exponent increased by 128 to bring its value into the representable range. If M9 is LOW, underflowed results in IBM format are replaced by an IBM "True Zero" (sign = 0, exponent = 0, fraction = 0).

This bit has no effect on operations for which the result format is not IBM.

Bit M8 — IBM Significance Mask Enable

If M8 is HIGH, results in IBM format that contain a non-zero exponent and a zero fraction (known as IBM "Floating-Point Zeros") are output unchanged. If M8 is LOW, results in IBM format that contain a non-zero exponent and a zero fraction are replaced by an IBM "True Zero" (sign = 0, exponent = 0, fraction = 0).

This bit has no effect on operations for which the result format is not IBM.

Bit M7 — IEEE Sudden Underflow Enable

If M7 is HIGH (and M6 is LOW, disabling IEEE traps), results in IEEE format that are denormalized are replaced by a zero of the same sign ("Sudden Underflow" mode). If M7 is LOW (and M6 is LOW, disabling IEEE traps), results in IEEE format that are denormalized are output unchanged as valid denormalized numbers ("Gradual Underflow" mode).

This bit has no effect on operations for which the result format is not IEEE or on operations for which the result format is IEEE but traps are enabled.

Bit M6 — IEEE Trapped Operation Enable

If M6 is HIGH, the exponents of IEEE overflowed results are reduced by 192 (single-precision) or 1536 (double-precision) to bring them into the representable range, the signs and fractions being unchanged, regardless of the setting of M4 (Saturate Enable). Similarly, if M6 is HIGH, the exponents of IEEE underflowed and denormalized results are increased by 192 (single-precision) or 1536 (double-precision) to bring them

into the representable range, the signs and fractions being unchanged, regardless of the setting of M7 (Sudden Underflow Enable). If M6 is LOW, the final results of overflowed and underflowed IEEE operations are determined by bits M4 (Saturate Enable) and M7 (Sudden Underflow Enable) respectively.

This bit has no effect on operations for which the result format is not IEEE.

Bit M5 — IEEE Affine/Projective Mode Select

If M5 is HIGH, IEEE infinities are interpreted in the "Affine" sense for addition, comparison and multiply-accumulate operations. If M5 is LOW, IEEE infinities are interpreted in the "Projective" sense for addition, comparison and multiply-accumulate operations. The difference between affine and projective interpretations is summarized below.

Affine and Projective Interpretations of Infinities

Operation	Affine Result & Flags	Projective Result & Flags
(+oo) + (+oo)	+oo; No flags	Quiet Nan; Flags I, R
(−oo) + (−oo)	−oo; No flags	Quiet Nan; Flags I, R

Flag I is the "Invalid Operation" flag. Flag R is the "Reserved Operand" flag.

This bit has no effect on base operations other than IEEE addition, IEEE comparison and IEEE multiply-accumulate.

Bit M4 – Saturate Enable

If M4 is HIGH, overflowed results are replaced by the largest representable number in the result format, with the same sign as the overflowed result, unless the result format is IEEE and M6 is HIGH, enabling IEEE trapped operation. If M4 is LOW, overflowed results are replaced by infinities (IEEE results, provided that M6 is LOW, disabling trapped operations), replaced by DEC Reserved Operands (DEC results) or left unchanged (IBM and integer results).

When IEEE traps are enabled (M6 is HIGH), this bit has no effect on operations for which the result format is IEEE.

Bits M3-M2 — Alternate Floating-Point Format Select

This field determines the alternate floating-point format:

M3	M2	Alternate Floating-Point Format (Double/Single)
0	0	IEEE (IEEE double-precision/IEEE single-precision)
0	1	DEC D (DEC D/DEC F)
1	0	DEC G (DEC G/DEC F)
1	1	IBM (IBM double-precision/IBM single-precision)

The alternate floating-point format is used for floating-point conversion operations to specify the destination format for the operation "Convert T to Alternate Format" and to specify the source format for the operation "Convert T from Alternate Format".

The floating-point formats are specified according to their double-precision names, since the information regarding the precisions of operands is provided to the processor via the S/$\overline{DR}$, S/$\overline{DS}$ and S/$\overline{DF}$ inputs. Note that the two, distinct, double-precision DEC formats, DEC D and DEC G, have the same single-precision format, DEC F.

Bits M1-M0 — Primary Floating-Point Format Select

This field determines the primary floating-point format:

M1	M0	Primary Floating-Point Format (Double/Single)
0	0	IEEE (IEEE double-precision/IEEE single-precision)
0	1	DEC D (DEC D/DEC F)
1	0	DEC G (DEC G/DEC F)
1	1	IBM (IBM double-precision/IBM single-precision)

The primary floating-point format is the format in which all floating-point inputs are assumed to be represented and in which all floating-point results are generated, taking into account specified precisions, with the following exceptions: the T-input for the operation "Convert T from Alternate Format" is assumed to be the alternate floating-point format and the result of the operation "Convert T to Alternate Format" is generated in the alternate floating-point format, again taking into account specified precisions.

The floating-point formats are specified according to their double-precision names, since the information regarding the precisions of operands is provided to the processor via the S/$\overline{DR}$, S/$\overline{DS}$ and S/$\overline{DF}$ inputs. Note that the two, distinct, double-precision DEC formats, DEC D and DEC G, have the same single-precision format, DEC F.

The mode register bits, which affect the arithmetic result of an operation rather than controlling data movement within the processor, are bits M13 through M4. The table below indicates the settings that should be used for these 10 bits to ensure strict conformance with the IEEE, DEC, and IBM standards. As regards IEEE traps and IBM masks, the relevant standards specify that these bits be set as dictated by the application. In all cases, of course, bits M1 and M0 must be set to specify the desired primary format.

It should be noted that the IEEE, DEC, and IBM standards do not explicitly define conversion operations between floating-point formats. When executing these operations, the appropriate settings for the mode register are, in general, determined by the system environment and the alogorithms to be executed. Bits M3 and M2 must be set to specify the required alternate floating-point format.

Mode Settings for Exact Compliance with Standards

Bit	Function	IEEE	DEC	IBM
13-12	Int FMTADJ	X	X	X
11	Int SIGNSEL	X	X	X
9	IBM UNFMASK	X	X	As Reqd.
8	IBM SIGMASK	X	X	As Reqd.
7	IEEE SUEN	0	X	X
6	IEEE TRAPEN	As Reqd.	X	X
5	IEEE AFF/$\overline{PROJ}$	1	X	X
4	SATEN	0	0	0

"0" = set to logic 0.
"1" = set to logic 1.
"X" = don't care.

Input Modes

The Am29C327 supports a total of eight input modes for loading data into the R and S registers.

The 32-bit bus modes allow the user to connect each input port ($R_{31} - R_0$ and $S_{31} - S_0$) to a separate 32-bit bus. 64-bit operands can then be loaded by placing the MSBs and LSBs alternately on the appropriate ports. In the 64-bit bus modes, the two input ports are configured internally as a single 64-bit port. The Am29C327 may then be connected directly to a 64-bit bus, and 64-bit operands may be loaded in a single operation. Either the 32-bit bus modes or the 64-bit bus modes may be used regardless of the precision of the operands being transferred — the choice of input modes will in practice be determined by the system into which the Am29C327 is to be integrated.

Single-cycle input modes allow two 64-bit operands to be loaded in a single clock cycle. This necessitates driving the input buses at twice the speed of the Am29C327. In systems where this is not practical, the double-cycle modes allow the loading of one 64-bit operand (or two 32-bit operands) per clock cycle.

Data may be loaded from the input buses to the R register and S register using one of the eight input modes:

1. 32-Bit Bus, Single-Cycle, LSWs First
2. 32-Bit Bus, Single-Cycle, MSWs First
3. 32-Bit Bus, Double-Cycle, LSWs First
4. 32-Bit Bus, Double-Cycle, MSWs First
5. 64-Bit Bus, Single-Cycle, R First
6. 64-Bit Bus, Single-Cycle, S First
7. 64-Bit Bus, Double-Cycle, R First
8. 64-Bit Bus, Double-Cycle, S First

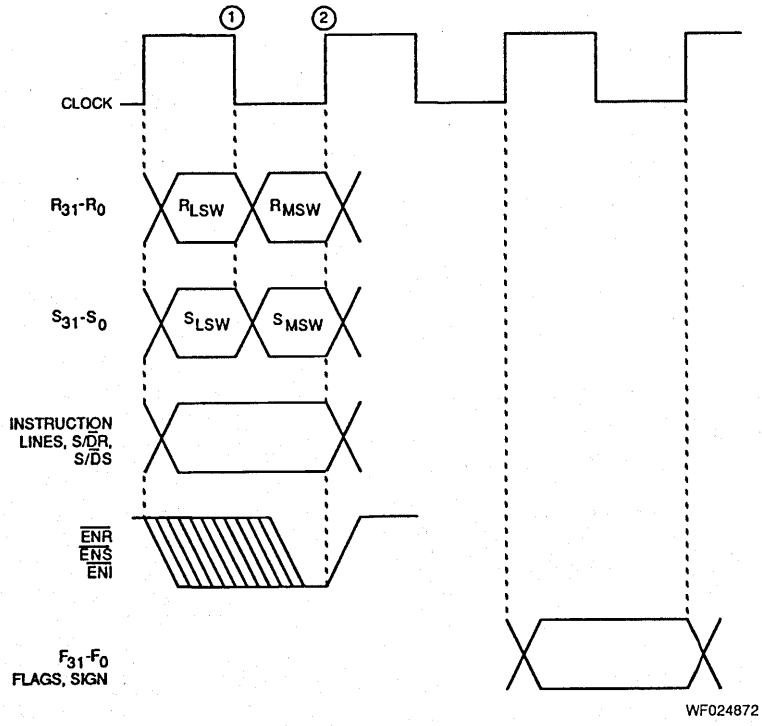
The choice of input mode is determined by mode register bits M16 – M14.

In order to permit the loading of new operands to be overlapped with the execution of a current operation, temporary registers are provided within the "operand router" block (shown in Figure 1). The operation of these temporary registers is transparent to the user. The conditions under which they are loaded depends on the input mode selected.

The eight input modes are described in the following pages.

32-Bit Bus, Single-Cycle, LSW First (M16 = 0, M15 = 0, M14 = 0)

In this mode, the two halves of the 64-bit R operand are placed on the R-port in successive half-cycles, with the S operand similarly placed on the S-port. After one complete cycle, the R and S registers contain the R and S operands, respectively.



Timing of Operations with Input Mode 1 (32-Bit Bus, Single-Cycle, LSW First)*

*Assumes processor flow-through mode, F register, and status register clocked.

The temporary registers are clocked on every HIGH-to-LOW clock transition.

At 1, the least-significant 32 bits of the R operand are loaded from the R-input port into the R-temp register, and the least-significant 32 bits of the S operand are loaded from the S-input port into the S-temp register. Both words are loaded on the HIGH-to-LOW transition of the clock.

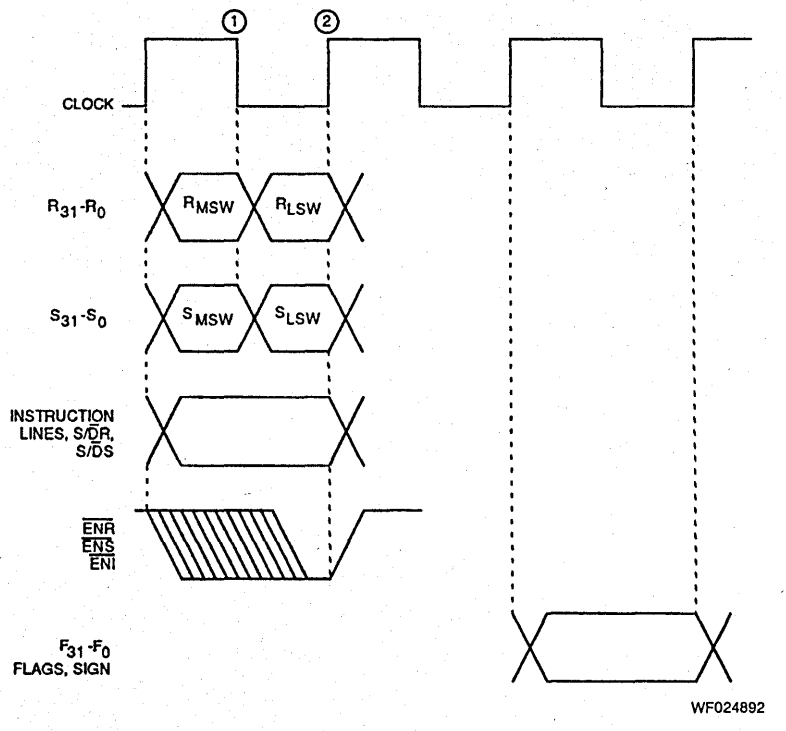
At 2, the most-significant 32 bits of the R operand are loaded from the R-input port into the most-significant half of the R register, and the most-significant 32 bits of the S operand are loaded from the S-input port into the most-significant half of the S register.

At the same time, at 2, the output of the R-temp register is loaded into the least-significant half of the R register, and the output of the S-temp register is loaded into the least-significant half of the S register.

If the R-operand is a single-precision (32-bit) floating-point operand, it is input in the R-MSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-LSW position is ignored by the processor. If the R-operand is a single-precision (32-bit) integer operand, it is input in the R-LSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-MSW position is ignored by the processor. The same procedure applies to the S-operand, if it is a single-precision operand.

32-Bit Bus, Single-Cycle, MSW First (M16 = 0, M15 = 0, M14 = 1)

In this mode, the two halves of the 64-bit R operand are placed on the R-port in successive half-cycles, with the S operand similarly placed on the S-port. After one complete cycle, the R and S registers contain the R and S operands, respectively.



Timing of Operations with Input Mode 2 (32-Bit Bus, Single-Cycle, MSW First)*

*Assumes processor flow-through mode, F register, and status register clocked.

The temporary registers are clocked on every HIGH-to-LOW clock transition.

At 1, the most-significant 32 bits of the R operand are loaded from the R-input port into the R-temp register, and the most-significant 32 bits of the S operand are loaded from the S-input port into the S-temp register. Both words are loaded on the HIGH-to-LOW transition of the clock.

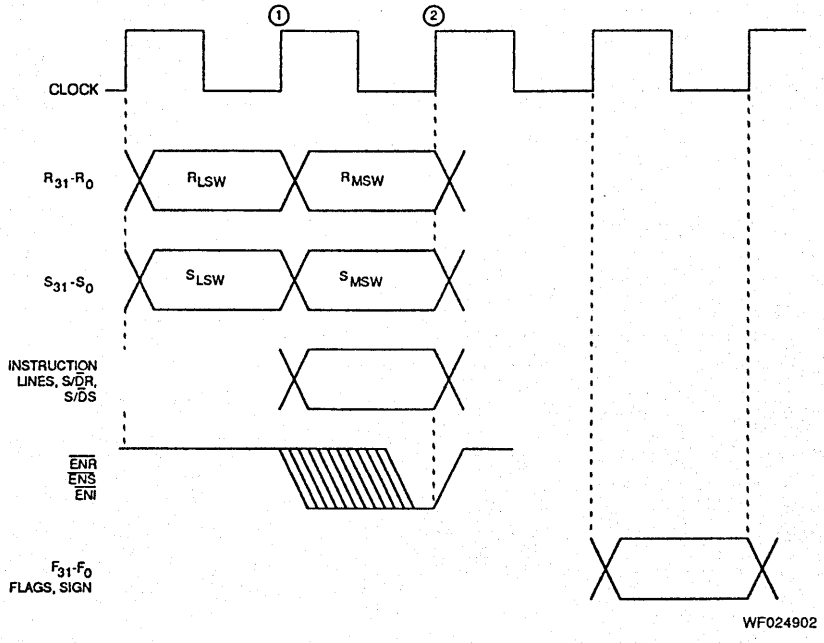
At 2, the least-significant 32 bits of the R operand are loaded from the R-input port into the least-significant half of the R register, and the least-significant 32 bits of the S operand are loaded from the S-input port into the least-significant half of the S register.

At the same time, at 2, the output of the R-temp register is loaded into the most-significant half of the R register, and the output of the S-temp register is loaded into the most-significant half of the S register.

If the R-operand is a single-precision (32-bit) floating-point operand, it is input in the R-MSW position (with S/$\overline{D}$R = HIGH) and the data in the R-LSW position is ignored by the processor. If the R-operand is a single-precision (32-bit) integer operand, it is input in the R-LSW position (with S/$\overline{D}$R = HIGH) and the data in the R-MSW position is ignored by the processor. The same procedure applies to the S-operand, if it is a single-precision operand.

32-Bit Bus, Double-Cycle, LSW First (M16 = 0, M15 = 1, M14 = 0)

In this mode, the two halves of the 64-bit R operand are placed on the R-port in successive cycles, with the S operand similarly placed on the S-port. After two cycles, the R and S registers contain the R and S operands, respectively.



Timing of Operations with Input Mode 3 (32-Bit Bus, Double-Cycle, LSW First)*

*Assumes processor flow-through mode, F register, and status register clocked.

The temporary registers are clocked on every LOW-to-HIGH clock transition.

At 1, the least-significant 32 bits of the R operand are loaded from the R-input port into the R-temp register, and the least-significant 32 bits of the S operand are loaded from the S-input port into the S-temp register.

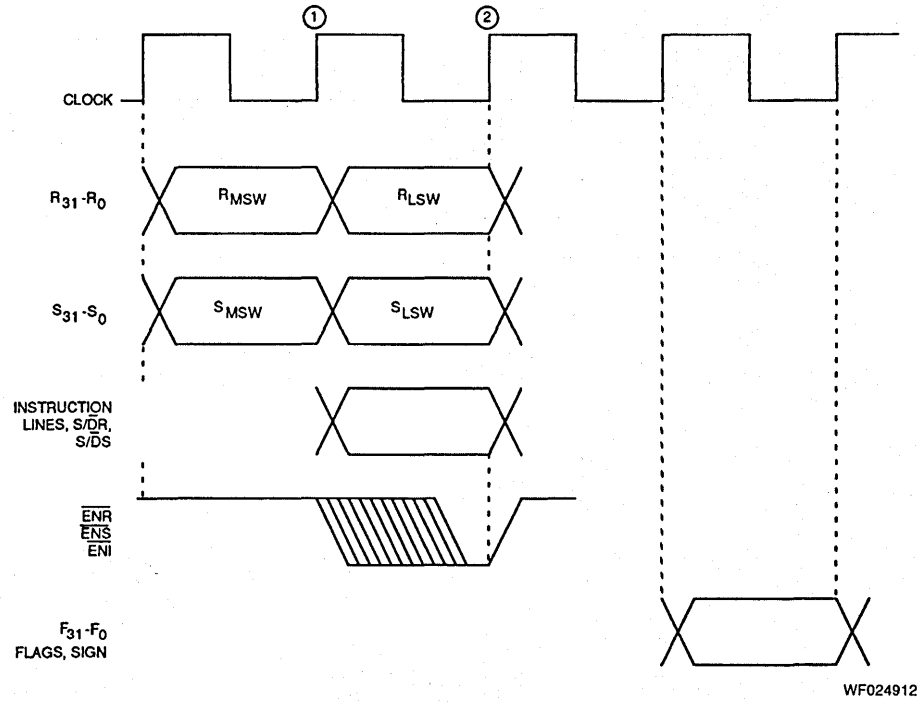
At 2, the most-significant 32 bits of the R operand are loaded from the R-input port into the most-significant half of the R register, and the most-significant 32 bits of the S operand are loaded from the S-input port into the most-significant half of the S register.

At the same time, at 2, the output of the R-temp register is loaded into the least-significant half of the R register, and the output of the S-temp register is loaded into the least-significant half of the S register.

If the R-operand is a single-precision (32-bit) floating-point operand, it is input in the R-MSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-LSW position is ignored by the processor. If the R-operand is a single-precision (32-bit) integer operand, it is input in the R-LSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-MSW position is ignored by the processor. The same procedure applies to the S-operand, if it is a single-precision operand.

32-Bit Bus, Double-Cycle, MSW First (M16 = 0, M15 = 1, M14 = 1)

In this mode, the two halves of the 64-bit R operand are placed on the R-port in successive cycles, with the S operand similarly placed on the S-port. After two cycles, the R and S registers contain the R and S operands, respectively.



Timing of Operations with Input Mode 4 (32-Bit Bus, Double-Cycle, MSW First)*

*Assumes processor flow-through mode, F register, and status register clocked.

The temporary registers are clocked on every LOW-to-HIGH clock transition.

At 1, the most-significant 32 bits of the R operand are loaded from the R-input port into the R-temp register, and the most-significant 32 bits of the S operand are loaded from the S-input port into the S-temp register.

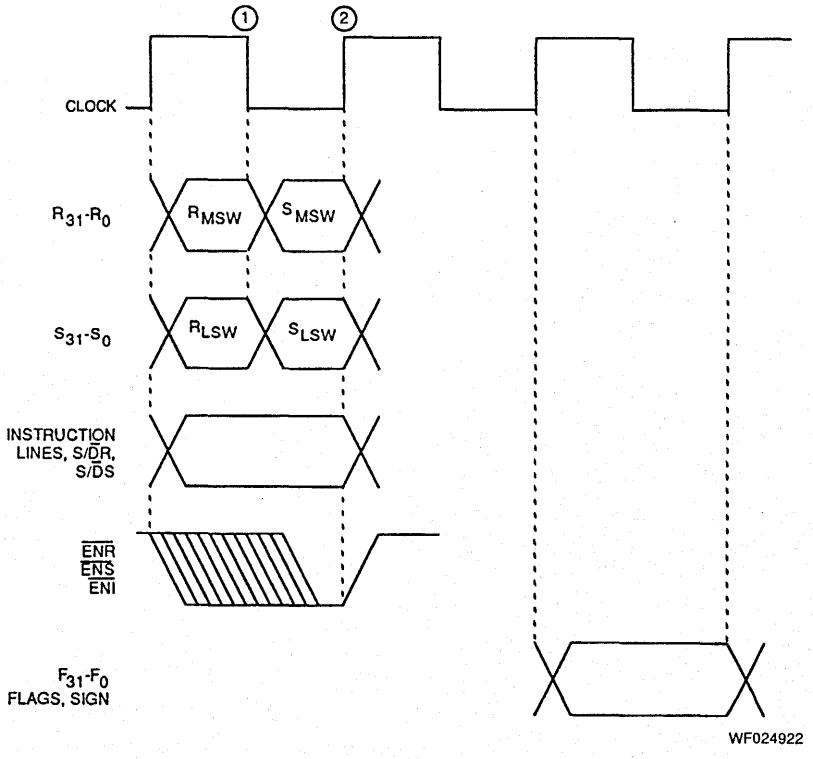
At 2, the least-significant 32 bits of the R operand are loaded from the R-input port into the least-significant half of the R register, and the least-significant 32 bits of the S operand are loaded from the S-input port into the least-significant half of the S register.

At the same time, at 2, the output of the R-temp register is loaded into the most-significant half of the R register, and the output of the S-temp register is loaded into the most-significant half of the S register.

If the R-operand is a single-precision (32-bit) floating-point operand, it is input in the R-MSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-LSW position is ignored by the processor. If the R-operand is a single-precision (32-bit) integer operand, it is input in the R-LSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-MSW position is ignored by the processor. The same procedure applies to the S-operand, if it is a single-precision operand.

64-Bit Bus, Single-Cycle, R First (M16 = 1, M15 = 0, M14 = 0)

In this mode, the MSW of the 64-bit R operand is placed on the R-port and the LSW on the S-port. Both halfwords are loaded in the first half cycle. Similarly, the two halves of the S operand are loaded in the second half cycle. After one full cycle, the R and S registers contain the R and S operands, respectively.



Timing of Operations with Input Mode 5 (64-Bit Bus, Single-Cycle, R First)*

*Assumes processor flow-through mode, F register, and status register clocked.

The temporary registers are clocked on every HIGH-to-LOW clock transition.

At 1, the most-significant 32 bits of the R operand are loaded from the R-input port into the R-temp register, and the least-significant 32 bits of the R operand are loaded from the S-input port into the S-temp register.

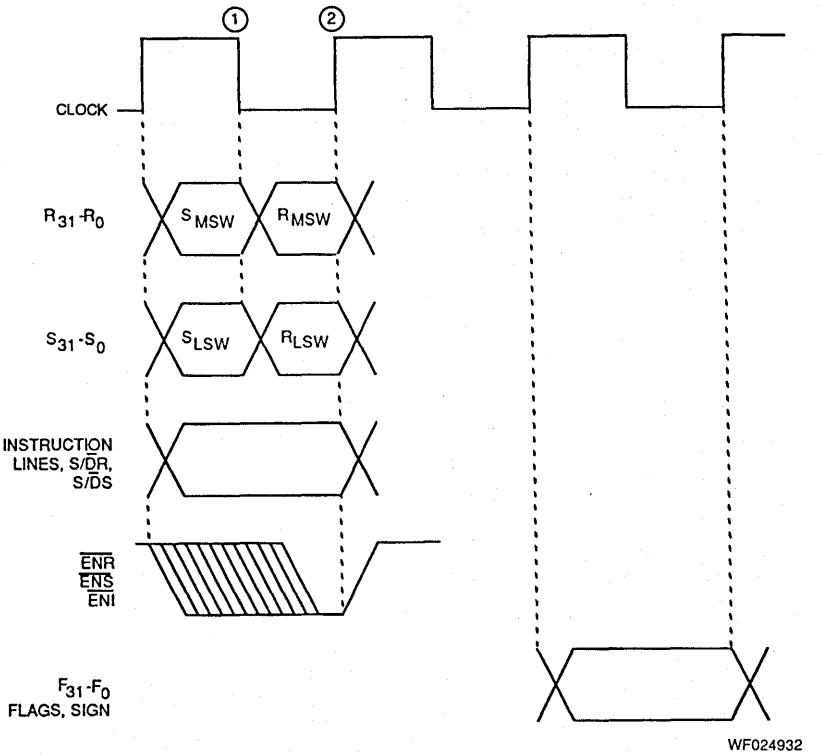
At 2, the most-significant 32 bits of the S operand are loaded from the R-input port into the most-significant half of the S register, and the least-significant 32 bits of the S operand are loaded from the S-input port into the least-significant half of the S register.

At the same time, at 2, the output of the R-temp register is loaded into the most-significant half of the R register, and the output of the S-temp register is loaded into the least-significant half of the R register.

If the R-operand is a single-precision (32-bit) floating-point operand, it is input in the R-MSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-LSW position is ignored by the processor. If the R-operand is a single-precision (32-bit) integer operand, it is input in the R-LSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-MSW position is ignored by the processor. The same procedure applies to the S-operand, if it is a single-precision operand.

64-Bit Bus, Single-Cycle, S First (M16 = 1, M15 = 0, M14 = 1)

In this mode, the MSW of the 64-bit S operand is placed on the R-port and the LSW on the S-port. Both halfwords are loaded in the first half cycle. Similarly, the two halves of the R operand are loaded in the second half cycle. After one full cycle, the R and S registers contain the R and S operands, respectively.



Timing of Operations with Input Mode 6 (64-Bit Bus, Single-Cycle, S First)*

*Assumes processor flow-through mode, F register, and status register clocked.

The temporary registers are clocked on every HIGH-to-LOW clock transition.

At 1, the most-significant 32 bits of the S operand are loaded from the R-input port into the R-temp register, and the least-significant 32 bits of the S operand are loaded from the S-input port into the S-temp register.

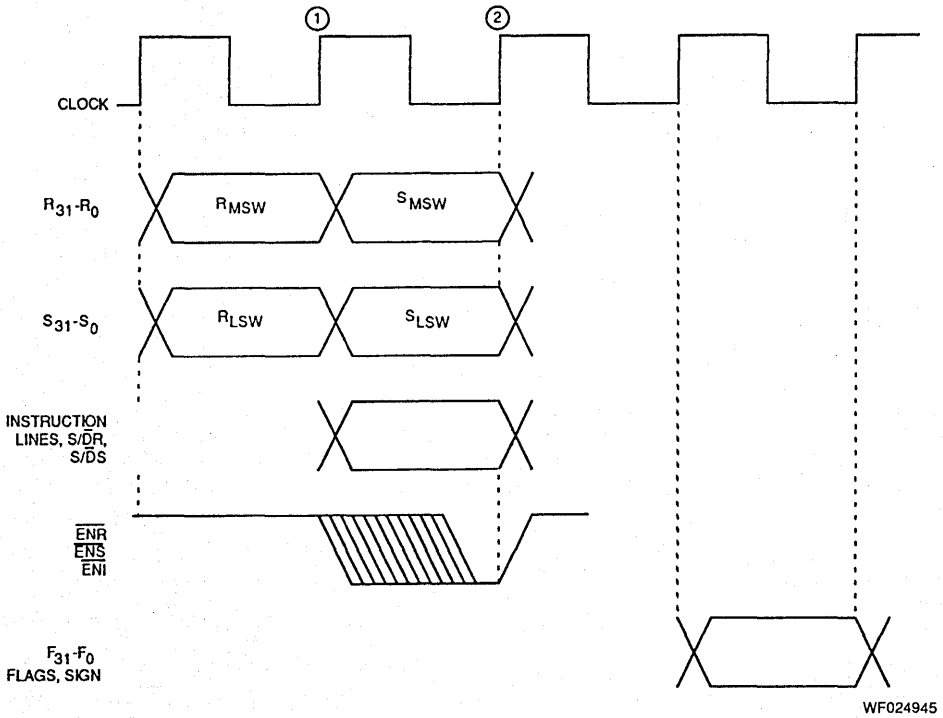
At 2, the most-significant 32 bits of the R operand are loaded from the R-input port into the most-significant half of the R register, and the least-significant 32 bits of the R operand are loaded from the S-input port into the least-significant half of the R register.

At the same time, at 2, the output of the R-temp register is loaded into the most-significant half of the S register, and the output of the S-temp register is loaded into the least-significant half of the S register.

If the R-operand is a single-precision (32-bit) floating-point operand, it is input in the R-MSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-LSW position is ignored by the processor. If the R-operand is a single-precision (32-bit) integer operand, it is input in the R-LSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-MSW position is ignored by the processor. The same procedure applies to the S-operand, if it is a single-precision operand.

64-Bit Bus, Double-Cycle, R First (M16 = 1, M15 = 1, M14 = 0)

In this mode, the MSW of the 64-bit R operand is placed on the R-port and the LSW on the S-port. Both halfwords are loaded in the first cycle. Similarly, the two halves of the S operand are loaded in the second cycle. After two cycles, the R and S registers contain the R and S operands, respectively.



Timing of Operations with Input Mode 7 (64-Bit Bus, Double-Cycle, R First)*

*Assumes processor flow-through mode, F register, and status register clocked.

The temporary registers are clocked on every LOW-to-HIGH clock transition.

At 1, the most-significant 32 bits of the R operand are loaded from the R-input port into the R-temp register, and the least-significant 32 bits of the R operand are loaded from the S-input port into the S-temp register.

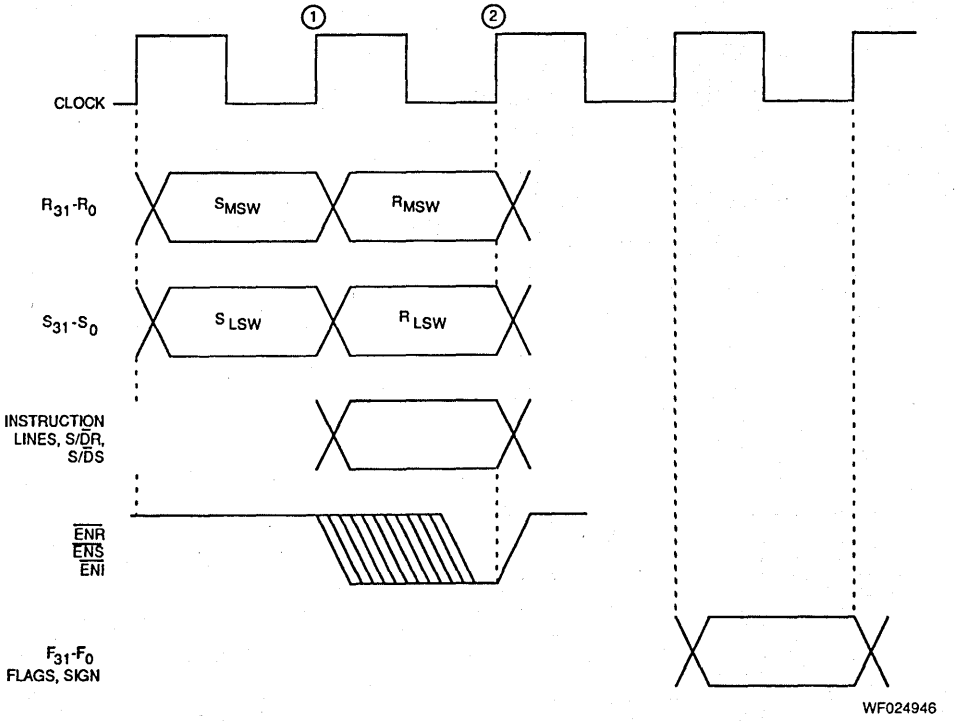
At 2, the most-significant 32 bits of the S operand are loaded from the R-input port into the most-significant half of the S register, and the least-significant 32 bits of the S operand are loaded from the S-input port into the least-significant half of the S register.

At the same time, at 2, the output of the R-temp register is loaded into the most-significant half of the R register, and the output of the S-temp register is loaded into the least-significant half of the R register.

If the R-operand is a single-precision (32-bit) floating-point operand, it is input in the R-MSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-LSW position is ignored by the processor. If the R-operand is a single-precision (32-bit) integer operand, it is input in the R-LSW position (with S/$\overline{DR}$ = HIGH) and the data in the R-MSW position is ignored by the processor. The same procedure applies to the S-operand, if it is a single-precision operand.

64-Bit Bus, Double-Cycle, S First (M16 = 1, M15 = 1, M14 = 1)

In this mode, the MSW of the 64-bit S operand is placed on the R-port and the LSW on the S-port. Both halfwords are loaded in the first cycle. Similarly, the two halves of the R operand are loaded in the second cycle. After two cycles, the R and S registers contain the R and S operands, respectively.



Timing of Operations with Input Mode 8 (64-Bit Bus, Double-Cycle, S First)*

*Assumes processor flow-through mode, F register, and status register clocked.

The temporary registers are clocked on every LOW-to-HIGH clock transition.

At 1, the most-significant 32 bits of the S operand are loaded from the R-input port into the R-temp register, and the least-significant 32 bits of the S operand are loaded from the S-input port into the S-temp register.

At 2, the most-significant 32 bits of the R operand are loaded from the R-input port into the most-significant half of the R register, and the least-significant 32 bits of the R operand are loaded from the S-input port into the least-significant half of the R register.

At the same time, at 2, the output of the R-temp register is loaded into the most-significant half of the S register, and the output of the S-temp register is loaded into the least-significant half of the S register.

If the R-operand is a single-precision (32-bit) floating-point operand, it is input in the R-MSW position (with S/$\overline{D}$R = HIGH) and the data in the R-LSW position is ignored by the processor. If the R-operand is a single-precision (32-bit) integer operand, it is input in the R-LSW position (with S/$\overline{D}$R = HIGH) and the data in the R-MSW position is ignored by the processor. The same procedure applies to the S-operand, if it is a single-precision operand.

Pipelining of Operations

The floating-point ALU of the Am29C327 may be operated in one of three modes:

1. Flow-Through Mode
2. Single-Pipelined Mode
3. Double-Pipelined Mode

Flow-Through Mode

In this mode the floating-point ALU acts as a purely combinatorial device.

Single-Pipelined Mode

In this mode the floating-point ALU contains a single pipeline delay for all operations; throughput is roughly double that for unpipelined mode. Simplified diagrams of the ALU configuration for single-pipelined mode are shown in Figure 2.

Double-Pipelined Mode

In this mode, which applies only to the multiply-accumulate operation, the ALU contains two pipeline delays; throughput is roughly triple that for the unpipelined multiplication-accumulation operation. A simplified block diagram is shown in Figure 3.

Figures 4 and 5 provide timing diagrams for all operations except multiply-accumulate, illustrating flow-through mode and pipelined mode, respectively. Figures 6, 7, and 8 provide timing diagrams for multiply-accumulate, illustrating flow-through mode, single-pipelined mode, and double-pipelined mode, respectively.

The choice of pipelining mode affects only the floating-point ALU. Operations of other parts of the Am29C327, such as the input registers, the output register, the mode register, and the instruction register are not affected by the choice of pipelining mode. However, the instruction bits are pipelined as they pass through the ALU. This permits instructions to be interleaved in pipelined mode.

The desired pipelined mode or modes can be invoked by setting mode register bits M20 and M19 to the appropriate values.

When using the Am29C327 in either single-pipelined or double-pipelined mode, two conditions must be observed:

1. The "load mode register" instruction is not pipelined, nor are any of the mode register bits. When the mode register is loaded, any differences between the current mode and the previous mode take effect immediately. In single-pipelined mode, the user should separate the last valid ALU instruction and the "load mode register" instruction with one "NO-OP" instruction. In double-pipelined mode, the user should separate them with two "NO-OP" instructions. A NO-OP instruction is any instruction whose result is not stored in register F, or the register file.
2. A multiplication-accumulation instruction cannot be immediately followed by any other type of instruction. This problem can be avoided by inserting a "dummy" multiplication-accumulation instruction at the end of a multiplication-accumulation instruction. This "dummy" instruction is any instruction whose results are not stored in register F or the register file.

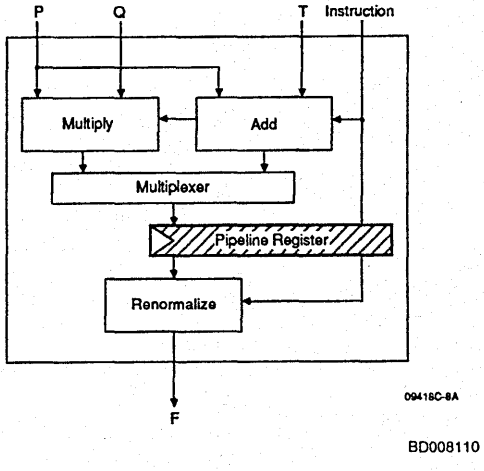


Figure 2.1. ALU in Pipelined Mode — All Operations Except Multiply-Accumulate

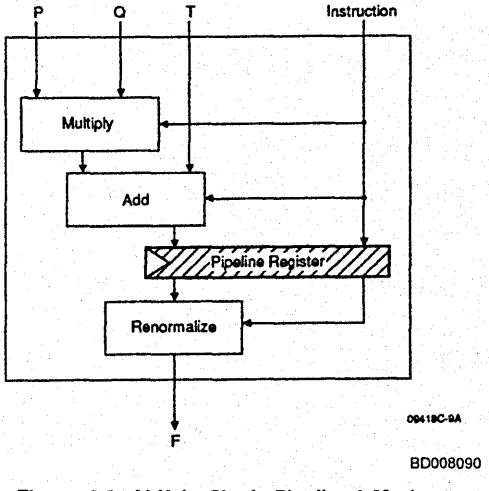
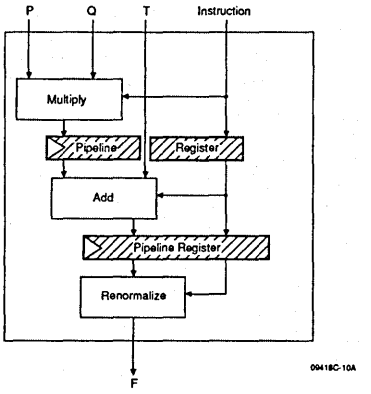
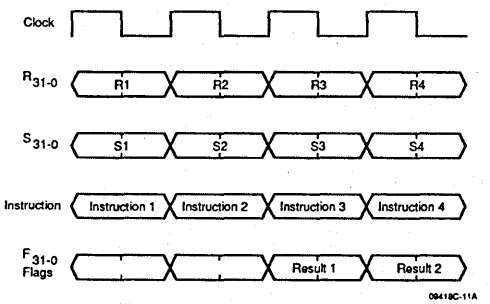


Figure 2.2. ALU in Single-Pipelined Mode — Multiply-Accumulate Only



BD008120

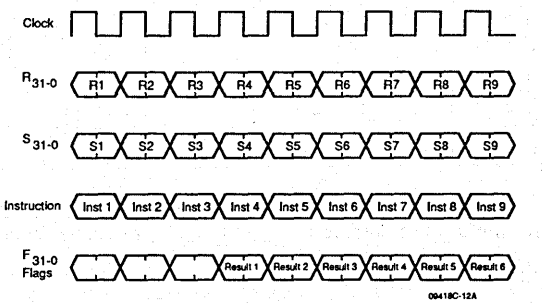
Figure 3. ALU Double-Pipelined Mode — Multiply-Accumulate Only



WF026460

Figure 4. Flowthrough Mode — All Operations Except Multiply-Accumulate*

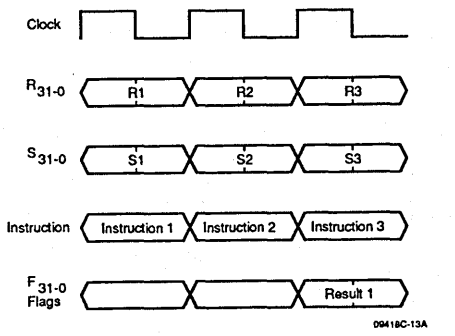
*Assumes F-register and status register clocked



WF026470

Figure 5. Pipelined Mode — All Operations Except Multiply-Accumulate*

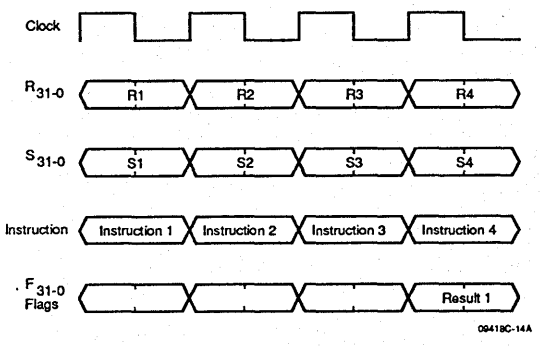
*Assumes F-register and status register clocked



WF026480

Figure 6. Flow-Through Mode — Multiply-Accumulate Operations Only*

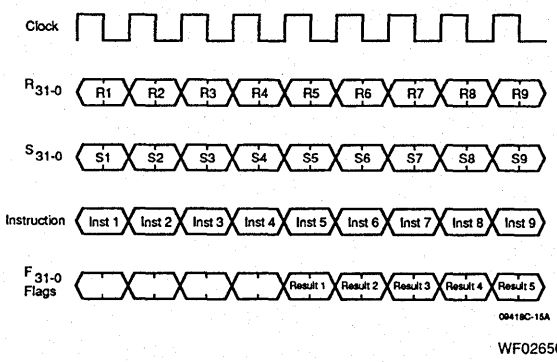
*Assumes F-register and status register clocked.



WF026490

Figure 7. Single-Pipelined Mode — Multiply-Accumulate Operations Only*

*Assumes F-register and status register clocked.



WF026500

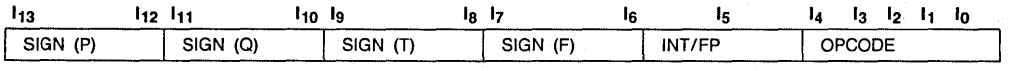
Figure 8. Double-Pipelined Mode — Multiply-Accumulate Operations Only*

*Assumes F-register and status register clocked.

Instruction Set

Instruction Register Format

The 14-bit instruction word $I_{13} - I_0$ comprises the sign-change controls, the integer/floating-point select bit, and the opcode.



The opcode field, $I_4 - I_0$, specifies the base operation to be performed by the ALU; instruction bit I_5 selects between floating-point and integer operations. The base operations and their corresponding opcodes are listed in Table 1.

TABLE 1. BASE OPERATIONS PERFORMED BY THE Am29C327

I5	I4 I3 I2 I1 I0	Mnemonic	Base Operation (Floating-Point)
0	0 0 0 0 0	FPAS	F' = P
0	0 0 0 0 1	FADD	F' = P' + T'
0	0 0 0 1 0	FMUL	F' = P' * Q'
0	0 0 0 1 1	FCMP	Compare P, T
0	0 0 1 0 0	FMAX	Maximum P, T
0	0 0 1 0 1	FMIN	Minimum P, T
0	0 0 1 1 0	FCTI	Convert T to Integer
0	0 0 1 1 1	FSTI	Scale T to Integer by Q
0	0 1 0 0 0	FMAC	F' = (P' * Q') + T'
0	0 1 0 0 1	FRND	Round T to Integral Value
0	0 1 0 1 0	FRCP	Reciprocal Seed of P
0	0 1 0 1 1	FCTA	Convert T to Alternate FP Format
0	0 1 1 0 0	FCFA	Convert T from Alternate FP Format
I5	**I4 I3 I2 I1 I0**	**Mnemonic**	**Base Operation (Integer)**
1	0 0 0 0 0	IPAS	F = P
1	0 0 0 0 1	IADD	F = P + T
1	0 0 0 1 0	IMUL	F = P * Q
1	0 0 0 1 1	ICMP	Compare P, T
1	0 0 1 0 0	IMAX	Maximum P, T
1	0 0 1 0 1	IMIN	Minimum P, T
1	0 0 1 1 0	ICTF	Convert T to Floating-Point
1	0 0 1 1 1	ISTF	Scale T to Floating-Point by Q
1	1 0 0 0 0	ILOR	F = P OR T
1	1 0 0 0 1	IAND	F = P AND T
1	1 0 0 1 0	IXOR	F = P XOR T
1	1 0 0 1 1	ILSH	Shift P Logical by Q Places
1	1 0 1 0 0	IASH	Shift P Arithmetic by Q Places
1	1 0 1 0 1	IFSH	Funnel Shift PT by Q Places
I5	**I4 I3 I2 I1 I0**	**Mnemonic**	**Base Operation (Format-Independent)**
X	1 1 0 0 0	MOVE	Move P
X	1 1 1 1 1	LMRG	Load Mode Register

Sign-Change Selects

Each ALU input and output operand has associated hardware that can be used to modify operand signs (see Figure 9). These sign-change blocks, when applied to base operations, greatly increase the number of available operations. A base operation of P' + T', for example, can be used to perform operations such as P – T, ABS(P + T), ABS(P) + ABS(T), and others, simply by modifying the signs of the input and output operands.

Using the sign-change blocks, the sign of an input operand may be left unchanged, complemented, set to zero, or set to one; the sign of the output operand may be left unchanged, complemented, set to zero, set to one, set to the sign of the P input operand, or set to the sign of the T input operand. Select decodes for the P, Q, T, and F operand sign-change blocks are shown in Table 2-1, 2-2, 2-3, and 2-4, respectively.

Note: The P-sign change block has no effect on the P input operand for the base operation F' = P.

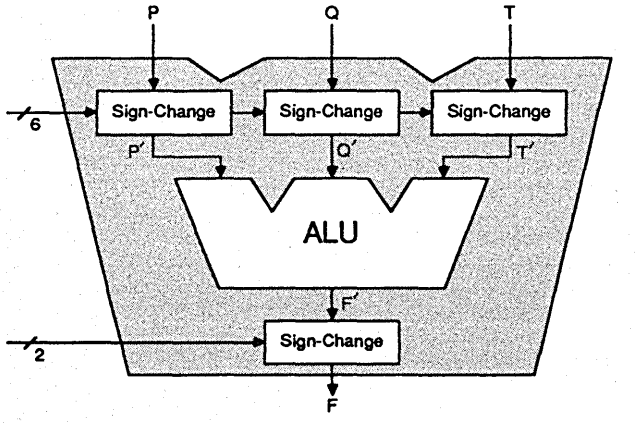


Figure 9. ALU Sign-Change Blocks

TABLE 2-1. SELECT DECODE FOR P OPERAND SIGN-CHANGE BLOCK

I_{13}	I_{12}	Sign (P')
0	0	SIGN (P)
0	1	$\overline{\text{SIGN}}$ (P)
1	0	0
1	1	1

Note: The P-sign change block has no effect on the P input operand for the base operation F' = P.

TABLE 2-2. SELECT DECODE FOR Q OPERAND SIGN-CHANGE BLOCK

I_{11}	I_{10}	Sign (Q')
0	0	SIGN (Q)
0	1	$\overline{\text{SIGN}}$ (Q)
1	0	0
1	1	1

TABLE 2-3. SELECT DECODE FOR T OPERAND SIGN-CHANGE BLOCK

I_9	I_8	Sign (T')
0	0	SIGN T
0	1	$\overline{\text{SIGN}}$ T
1	0	0
1	1	1

TABLE 2-4. SELECT DECODE FOR F OPERAND SIGN-CHANGE BLOCK

Base Operation	I_{11}	I_{10}	I_7	I_6	Sign (F)
F' = P	0	x	0	0	SIGN (F')
or	0	x	0	1	$\overline{\text{SIGN}}$ (F')
Maximum P, T	0	x	1	0	0
or	0	x	1	1	1
Minimum P, T	1	0	x	x	SIGN (P)
	1	1	x	x	SIGN (T)
	x	x	0	0	SIGN (F')
All other	x	x	0	1	$\overline{\text{SIGN}}$ (F')
base operations	x	x	1	0	0
	x	x	1	1	1

Operand Multiplexer Selects

The instruction fields $PSEL_3 - PSEL_0$, $QSEL_3 - QSEL_0$, and $TSEL_3 - TSEL_0$ specify the select codes for the P, Q, and T operand multiplexers, respectively; the codes are summarized in Table 3.

TABLE 3. OPERAND MULTIPLEXER SELECT CODES

$PSEL_3$ $QSEL_3$ $TSEL_3$	$PSEL_2$ $QSEL_2$ $TSEL_2$	$PSEL_1$ $QSEL_1$ $TSEL_1$	$PSEL_0$ $QSEL_0$ $TSEL_0$	P Q T
0	0	0	0	R - Register
0	0	0	1	S - Register
0	0	1	0	0
0	0	1	1	0.5 (Floating Point); -1 (Integer)
0	1	0	0	1
0	1	0	1	2
0	1	1	0	3
0	1	1	1	Pi (Floating Point); Max Neg. Value (Integer)
1	0	0	0	Register File 0 (RF0)
1	0	0	1	Register File 1 (RF1)
1	0	1	0	Register File 2 (RF2)
1	0	1	1	Register File 3 (RF3)
1	1	0	0	Register File 4 (RF4)
1	1	0	1	Register File 5 (RF5)
1	1	1	0	Register File 6 (RF6)
1	1	1	1	Register File 7 (RF7)

Operand Precisions

The Am29C327 supports mixed-precision operations, so that it is possible, for example, for an operation to have single-precision inputs and a double-precision output, or one single- and one double-precision input, or any other combination.

Precision of the operands in registers R and S is specified by signals $S/\overline{D}R$ and $S/\overline{D}S$ at the time the operands are loaded.

The precision of an operation result is specified by signal $S/\overline{D}F$.

Operands stored in the register file are each accompanied by a bit indicating that operand's precision; this precision information is automatically supplied to the ALU when a register file location is used as an input operand to an operation.

Processor Operations

Table 4 illustrates a number of possible ALU instructions comprising the opcode, integer/floating-point select, and sign-change fields. Note that the remaining instruction bits — P, Q, and T operand multiplexer selects; the rounding modes; and the output operand precision — can be specified independently.

The user may create instructions using instruction words other than those listed in Table 4. For some base operations, sign-change control settings are completely arbitrary; for others, only the sign-change field values shown in Table 4 are valid. Table 5 summarizes permissible sign-change field values for each core operation.

TABLE 4. INSTRUCTION WORDS

Operation	Sign P	Sign Q	Sign T	Sign F	I/$\overline{F}$	Opcode
FP F = P	xx	0x	xx	00	0	00000
FP F = −P	xx	0x	xx	01	0	00000
FP F = ABS (P)	xx	0x	xx	10	0	00000
FP F = Sign (T)*ABS (P)	xx	11	xx	xx	0	00000
FP F = P + T	00	xx	00	00	0	00001
FP F = P − T	00	xx	01	00	0	00001
FP F = T − P	01	xx	00	00	0	00001
FP F = −P − T	01	xx	01	00	0	00001
FP F = ABS (P + T)	00	xx	00	10	0	00001
FP F = ABS (P − T)	00	xx	01	10	0	00001
FP F = ABS (P) + ABS (T)	10	xx	10	00	0	00001
FP F = ABS (P) − ABS (T)	10	xx	11	00	0	00001
FP F = ABS (ABS (P) − ABS (T))	10	xx	11	10	0	00001
FP F = P * Q	00	00	xx	00	0	00010
FP F = (−P) * Q	01	00	xx	00	0	00010
FP F = ABS (P * Q)	00	00	xx	10	0	00010
FP Compare P, T	00	xx	01	00	0	00011
FP Max P, T	00	00	01	00	0	00100
FP Max ABS (P), ABS (T)	10	00	11	00	0	00100
FP Min P, T	01	00	00	00	0	00101
FP Min ABS (P), ABS (T)	11	00	10	00	0	00101
FP Limit P to Magnitude T	11	10	10	xx	0	00101
FP Convert T to Integer	xx	xx	00	00	0	00110
FP Scale T to Integer by Q	xx	00	00	00	0	00111
FP F = (P*Q) + T	00	00	00	00	0	01000
FP F = T − (P*Q)	01	00	00	00	0	01000
FP F = (P*Q)−T	00	00	01	00	0	01000
FP F = −T − P*Q	01	00	01	00	0	01000
FP F = ABS (T) + ABS (P*Q)	10	10	10	00	0	01000
FP F = ABS (T) − ABS (P*Q)	11	10	10	00	0	01000
FP F = ABS (P*Q) − ABS (T)	10	10	11	00	0	01000
FP Round T to Integral Value	xx	xx	00	00	0	01001
FP Reciprocal Seed (P)	00	xx	xx	00	0	01010
FP Convert T to Alternate Floating-point Format	xx	xx	00	00	0	01011
FP Convert T from Alternate Floating-point Format	xx	xx	00	00	0	01100
Int F = P	00	00	00	00	1	00000
Int F = −P	00	00	00	01	1	00000
Int F = ABS (P)	00	00	00	10	1	00000
Int F = sign (T)*ABS (P)	00	11	00	xx	1	00000
Int F = P + T	00	xx	00	00	1	00001
Int F = P − T	00	xx	01	00	1	00001
Int F = T − P	01	xx	00	00	1	00001
Int F = ABS (P + T)	00	xx	00	10	1	00001
Int F = ABS (P − T)	00	xx	01	10	1	00001
Int F = P * Q	00	00	xx	00	1	00010
Int Compare P, T	00	xx	01	00	1	00011
Int Max P, T	00	00	01	00	1	00100
Int Min P, T	01	00	00	00	1	00101

TABLE 4. INSTRUCTION WORDS (Cont'd.)

Operation	Sign P	Sign Q	Sign T	Sign F	I/$\overline{F}$	Opcode
Int **Convert T to Float**	xx	xx	00	00	1	00110
Int **Scale T to Float by Q**	xx	00	00	00	1	00111
Int **P OR T**	xx	xx	xx	xx	1	10000
Int **P AND T**	xx	xx	xx	xx	1	10001
Int **P XOR T**	xx	xx	xx	xx	1	10010
Int **NOT T (see Note 1)**	xx	xx	xx	xx	1	10010
Int **Shift P Logical by Q Places**	00	00	xx	00	1	10011
Int **Shift P Arithmetic by Q Places**	00	00	xx	00	1	10100
Int **Funnel Shift PT by Q Places**	00	00	00	00	1	10101
Move P	xx	xx	xx	xx	x	11000
Load Mode Register	xx	xx	xx	xx	x	11111

Notes: 1. NOT T is performed by XORing T with a word containing all 1's (integer −1). When invoking NOT T the user must set $PSEL_3 - PSEL_0$ to 0011_2, thus selecting integer constant −1.

TABLE 5. ALLOWABLE SIGN-CHANGE/BASE OPERATION COMBINATIONS

	I 11111 5 43210	Base Operation	Sign-Change Fields Sign (P)	Sign (Q)	Sign (T)	Sign (F)
	0 00000	FP F' = P*	V	V	x	V
	0 00001	FP F' = P' + T'	V	x	V	V
	0 00010	FP F' = P'*Q'	V	V	x	V
	0 00011	FP Compare P, T	F	x	F	F
Floating-	0 00100	FP Max P, T	F	F	F	F
Point	0 00101	FP Min P, T	F	F	F	F
Operations	0 00110	FP Cvt T to Int	x	x	F	F
	0 00111	FP Scale T to Int	x	F	F	F
	0 01000	FP F' = (P'*Q') + T'	V	V	V	V
	0 01001	FP Round T	x	x	F	F
	0 01010	FP Recip Seed P	F	x	x	F
	0 01011	FP Cvt T to Alt Fmt	x	x	F	F
	0 01100	FP Cvt T fm Alt Fmt	x	x	F	F
	1 00000	Int F = P	F	F	F	F
	1 00001	Int F = P + T	F	x	F	F
	1 00010	Int F = P*Q	F	F	x	F
	1 00011	Int Compare P, T	F	x	F	F
	1 00100	Int Max P, T	F	F	F	F
	1 00101	Int Min P, T	F	F	F	F
	1 00110	Int Cvt T to f.p.	x	x	F	F
Integer	1 00111	Int Scale T to f.p.	x	F	F	F
Operations	1 10000	Int F = P OR T	x	x	x	x
	1 10001	Int F = P AND T	x	x	x	x
	1 10010	Int F = P XOR T	x	x	x	x
	1 10011	Int Shift P Logical	F	F	x	F
	1 10100	Int Shift P Arith	F	F	x	F
	1 10101	Int Funnel Shift PT	F	F	F	F
	x 11000	Move P	x	x	x	x
	x 11111	Load Mode Reg	x	x	x	x

Key: V = Variable; user can specify arbitrary sign change.
F = Fixed; user is restricted to sign change combinations shown in Table 4.
x = Don't care; this field does not affect the operation or its result.

*Note: The P-sign change block has no effect on the P-input operand for the base operation F' = P.

Base Operation Code Description

F' = P (Floating-point) FPAS: The P-operand is passed through the ALU unchanged, except for any specified precision conversions. If the user specifies different input and output precisions, the operation may be used to perform single-to-double or double-to-single conversions. Instructions such as negation, absolute value extraction and sign-transfer may be executed by setting the sign-change controls appropriately while executing this base operation.

*Note: The P-sign change block has no effect on the P-input operand for the base operation F' = P.

F' = P' + T' (Floating-point) FADD: The two operands P' and T' are added, taking into account any specified precision conversions. Instructions such as subtraction, sum-of-absolute-values, difference-of-absolute-values, absolute-value-of-sum, and absolute-value-of-difference may be executed by setting the sign-change controls appropriately while executing this base operation.

F' = P' * Q' (Floating-point) FMUL: The Operands P' and Q' are multiplied, taking into account any specified precision conversions. Instructions such as negative-product and absolute-value-of-product may be executed by setting the sign-change controls appropriately while executing this base operation.

Compare P, T (Floating-point) FCMP: The two operands P and T are compared, taking into account any specified precision conversions. The output of the operation is the result of the subtraction (P-T). The flags are set appropriately to indicate the result of the comparison, conforming to the relevant parts of the floating-point standards. For IEEE and DEC operations, one of four flags (greater than, less than, equals or unordered) is set for any given compare operation. For IBM operations, the unordered flag does not apply since the format does not support any reserved operands.

Maximum P, T (Floating-point) FMAX: The two operands P and T are compared, taking into account any specified precision conversions. The most-positive operand is selected as the output. The "Winner" flag indicates which of the operands is selected. Additionally, the operation maximum-of-absolute-value may be performed by setting the appropriate sign-change controls.

Minimum P, T (Floating-point) FMIN: The two operands P and T are compared, taking into account any specified precision conversions. The most-negative operand is selected as the output. The "Winner" flag indicates which of the two operands is selected. Additionally, the operations minimum-of-absolute-values and limit-P-to-magnitude-T may be performed by setting the appropriate sign-change controls. The limit-P-to-magnitude-T operation is useful for "clipping" a sequence of operands to ensure that their magnitude never exceeds a preset limit.

Convert T to Integer (Floating-point) FCTI: The operand T is converted from floating-point representation to two's complement integer representation, taking into account the specified precision of the floating-point operand. If the output precision is specified as single, the result is a 32-bit integer. If the output precision is specified as double, the result is a 64-bit integer.

Scale T to Integer by Q (Floating-point) FSTI: The operand T is converted from floating-point representation to two's complement integer representation, using the exponent of the floating-point operand Q as a scale factor and taking into account the specified precision of the floating-point operands. The unbiased exponent of the operand Q is added to the exponent of the operand T, permitting IEEE and DEC operands to be multiplied by any power of 2 and IBM operands by any power of 16, before the conversion is performed. If the output precision is specified as single, the result is a 32-bit integer. If the output precision is specified as double, the result is a 64-bit integer.

F' = (P' * Q') + T' (Floating-point) FMAC: The operands P' and Q' are multiplied, producing a double-precision product. This product is added to the operand T', taking into account any specified precision conversions. Instructions such as "P*Q – T", "T – P*Q", "ABS(P*Q) + ABS(T)" and "ABS (P*Q + T)" may be executed by setting the sign-change controls appropriately while executing this base operation.

Round T to Integral Value (Floating-point) FRND: The floating-point operand T is rounded to an integer-valued floating-point operand, using the specified rounding mode and taking into account any specified precision conversions. As an example, the operation converts a floating-point representation of Pi (3.14159...) to a floating-point representation of 3.0 or 4.0, depending on the rounding mode selected. The final result of the operation is a floating-point number.

Reciprocal Seed of P (Floating-point) FRCP: An approximation to the reciprocal of the operand P is evaluated, taking into account any specified precision conversions. The reciprocal seed comprises an accurate sign, a fully-accurate exponent and a mantissa which is accurate to only one place. This operation can be used as the initial step in performing Newton-Raphson division; optionally, an external seed look-up table can be used for faster convergence.

Convert T to Alternate Floating-point Format (Floating-point) FCTA: The floating-point operand T, assumed to be in the "primary" floating-point format is converted to a floating-point operand in the "Alternate" floating-point format, taking into account any specified precision conversions.

Convert T from Alternate Floating-point Format (Floating-point) FCFA: The floating-point operand T, assumed to be in the "Alternate" floating-point format is converted to a floating-point operand in the "primary" floating-point format, taking into account any specified precision conversions.

F = P (Integer) IPAS: The P-operand is passed through the ALU unchanged, except for any specified precision conversions. If the user specifies different input and output precisions, the operation may be used to perform single-to-double or double-to-single conversions. Instructions such as negation, absolute value extraction and sign transfer may be performed by setting the sign-change control appropriately while executing this base operation.

F = P + T (Integer) IADD: The two operands P and T are added, taking into account any specified precision conversions. Instructions such as subtraction, absolute-value-of-sum and absolute-value-of-difference may be performed by setting the sign-change controls appropriately while executing this base operation.

F = P * Q (Integer) IMUL: The two operands P and Q are multiplied, taking into account any specified precision conversions. Either 32-bit multiplication or 64-bit multiplication may be performed, and the user may select either the MSBs or the LSBs of the product as the final result. In addition, format-adjusting may be implemented if required, and the operands may be considered as signed (two's complement) or unsigned.

Compare P, T (Integer) ICMP: The two operands P and T are compared, taking into account any specified precision conversions. The output of the operation is the result of the subtraction (P - T). The flags are set appropriately to indicate the result of the comparison, one of three flags (greater than,

less than or equals) being set for any given compare operation.

Maximum P, T (Integer) IMAX: The two operands P and T are compared, taking into account any specified precision conversions. The most-positive operand is selected as the output. The "Winner" flag indicates which of the two operands is selected.

Minimum P, T (Integer) IMIN: The two operands P and T are compared, taking into account any specified precision conversions. The most-negative operand is selected as the output. The "Winner" flag indicates which of the two operands is selected.

Convert T to Floating-point (Integer) ICTF: The operand T is converted from two's complement integer representation to floating-point representation, taking into account the specified precision of the integer operand. If the output precision is specified as single, the result is a 32-bit floating-point operand. If the output precision is specified as double, the result is a 64-bit floating-point operand.

Scale T to Floating-point by Q (Integer) ISTF: The operand T is converted from two's complement integer representation to floating-point representation, using the exponent of the floating-point operand Q as a scale factor and taking into account the specified precision of the integer operand. The unbiased exponent of the operand Q is added to the exponent of the floating-point result, permitting IEEE and DEC operands to be multiplied by any power of 2, and IBM operands by any power of 16, after the conversion is performed. If the output precision is specified as single, the result is a 32-bit floating-point operand. If the output precision is specified as double, the result is a 64-bit floating-point operand.

F = P OR T (Integer) ILOR: The operand P is logically ORed with the operand T. Before the operation is performed, the inputs, if 32-bit, are sign-extended to 64-bits.

F = P AND T (Integer) IAND: The operand P is logically ANDed with the operand T. Before the operation is performed, the inputs, if 32-bit, are sign-extended to 64-bits.

F = P XOR T (Integer) IXOR: The operand P is logically exclusive-ORed with the operand T. Before the operation is performed, the inputs, if 32-bit, are sign-extended to 64-bits. This operation may be used to invert an operand by selecting the second operand to be the integer constant -1, so that all bits of this second operand are 1. Exclusive-ORing an operand with -1 is equivalent to inverting each bit in the operand.

Shift P Logical by Q Places (Integer) ILSH: This operation cannot be performed in mixed-precision mode. The precision of the result is the same as the precision of the input operand P. A two's-complement shift length in the range −64 to +63 (double-precision) or −32 to +31 (single-precision) is extracted from the LSBs of the operand Q. The operand P is logically right-shifted by the number of places specified by the shift length. A negative shift length therefore produces a left-shift. If a right-shift is performed, zeros fill vacated bit positions to the left of the input operand. If a left-shift is performed, zeros fill vacated bit positions to the right of the input operand.

Shift P Arithmetic by Q Places (Integer) IASH: This operation cannot be performed in mixed-precision mode. The precision of the result is the same as the precision of the input operand P. A two's-complement shift length in the range −64 to +63 (double-precision) or −32 to +31 (single-precision) is extracted from the LSBs of the operand Q. The operand P is arithmetically right-shifted by the number of places specified by the shift length. A negative shift length therefore produces a left-shift. If a right-shift is performed, the MSB (bit 63 or 31) is replicated to fill vacated bit positions to the left of the input operand. If a left-shift is performed, zeros fill vacated bit positions to the right of the input operand.

Funnel Shift PT by Q Places (Integer) IFSH: This operation cannot be performed in mixed-precision mode. The operand T is interpreted as having the same precision as the input operand P and the precision of the result is also the same as the precision of the input operand P. A two's-complement shift length in the range −64 to +63 (double-precision) or −32 to +31 (single-precision) is extracted from the LSBs of the operand Q. A triple-width operand (96-bit or 192-bit) is formed by concatenating the input operands into the arrangement P-T-P, with the 32-bit or 64-bit result field initially aligned with the T-operand. The triple-width operand is logically right-shifted by the number of places specified by the shift length. A negative shift length therefore produces a left-shift.

Move P (Format Independent) MOVE: The 64-bit operand P is passed unchanged through the ALU. No exceptions are detected or signalled.

Load Mode Register (Format Independent) LMRG: The 32-bit operand on the R_{31-0} input port is loaded into the mode register on the rising edge of the clock input. No exceptions are detected or signalled.

Operation Flags

For each operation, the ALU produces thirteen flags that indicate operation status. Of the flags produced, a maximum of seven are relevant to any given operation. The relevant flags are clocked into the status register, and the other flags are discarded.

The ALU flags are:

C — CARRY: Carry-out bit produced by integer addition, subtraction, or comparison.

I — INVALID OPERATION: Input operands are unsuitable for the operation specified (e.g., ∞ * 0).

R — RESERVED OPERAND: Reserved operand detected/generated.

S — SIGN: Result sign.

U — UNDERFLOW: Result underflowed the destination format.

V — OVERFLOW: Result overflowed the destination format.

W — WINNER: Indicates which of the two operands was selected when performing Max/Min operations.

X — INEXACT RESULT: Result had to be rounded to fit the destination format.

Z — ZERO: Zero result.

>, =, <, # — GREATER THAN, EQUAL, LESS THAN, UNORDERED: Used to report the result of a comparison operation.

Table 6 lists the flags reported for each operation.

TABLE 6. ORGANIZATION OF FLAGS APPLICABLE TO EACH BASE OPERATION

Format	Operation	Mnemonic	SIGN	Flag6	Flag5	Flag4	Flag3	Flag2	Flag1
IEEE	F' = P	FPAS	S	Z	X	U	V	R	I
IEEE	F' = P' + T'	FADD	S	Z	X	U	V	R	I
IEEE	F' = P' x Q'	FMUL	S	Z	X	U	V	R	I
IEEE	Compare P, T	FCMP	S	=	>	<	#	R	I
IEEE	Maximum P, T	FMAX	S	Z		W		R	I
IEEE	Minimum P, T	FMIN	S	Z		W		R	I
IEEE	Convert T to Integer	FCTI	S	Z	X		V	R	I
IEEE	Scale T to Integer	FSTI	S	Z	X		V	R	I
IEEE	F' = (P' x Q') + T'	FMAC	S	Z		U	V	R	I
IEEE	Round T to Integral Value	FRND	S	Z	X		V	R	I
IEEE	Reciprocal Seed of P	FRCP	S	Z		U	V	R	I
IEEE	Convert T to Alt FP Format	FCTA	S	Z	X	U	V	R	I
IEEE	Convert T from Alt FP Format	FCFA	S	Z	X	U	V	R	I
DEC D	F' = P	FPAS	S	Z	X		V	R	
DEC D	F' = P' + T'	FADD	S	Z	X	U	V	R	
DEC D	F' = P' x Q'	FMUL	S	Z	X	U	V	R	
DEC D	Compare P, T	FCMP	S	=	>	<	#	R	
DEC D	Maximum P, T	FMAX	S	Z		W		R	
DEC D	Minimum P, T	FMIN	S	Z		W		R	
DEC D	Convert T to Integer	FCTI	S	Z	X		V	R	I
DEC D	Scale T to Integer	FSTI	S	Z	X		V	R	I
DEC D	F' = (P' x Q') + T'	FMAC	S	Z		U	V	R	
DEC D	Round T to Integral Value	FRND	S	Z	X		V	R	
DEC D	Reciprocal Seed of P	FRCP	S	Z		U	V	R	I
DEC D	Convert T to Alt FP Format	FCTA	S	Z	X	U	V	R	I
DEC D	Convert T from Alt FP Format	FCFA	S	Z	X	U	V	R	I
DEC G	F' = P	FPAS	S	Z	X	U	V	R	
DEC G	F' = P' + T'	FADD	S	Z	X	U	V	R	
DEC G	F' = P' x Q'	FMUL	S	Z	X	U	V	R	
DEC G	Compare P, T	FCMP	S	=	>	<	#	R	
DEC G	Maximum P, T	FMAX	S	Z		W		R	
DEC G	Minimum P, T	FMIN	S	Z		W		R	
DEC G	Convert T to Integer	FCTI	S	Z	X		V	R	I
DEC G	Scale T to Integer	FSTI	S	Z	X		V	R	I
DEC G	F' = (P' x Q') + T'	FMAC	S	Z		U	V	R	
DEC G	Round T to Integral Value	FRND	S	Z	X		V	R	
DEC G	Reciprocal Seed of P	FRCP	S	Z		U	V	R	I
DEC G	Convert T to Alt FP Format	FCTA	S	Z	X	U	V	R	I
DEC G	Convert T from Alt FP Format	FCFA	S	Z	X	U	V	R	I
IBM	F' = P	FPAS	S	Z	X		V		
IBM	F' = P' + T'	FADD	S	Z	X	U	V		
IBM	F' = P' x Q'	FMUL	S	Z	X	U	V		
IBM	Compare P, T	FCMP	S	=	>	<			
IBM	Maximum P, T	FMAX	S	Z		W			
IBM	Minimum P, T	FMIN	S	Z		W			
IBM	Convert T to Integer	FCTI	S	Z	X		V		
IBM	Scale T to Integer	FSTI	S	Z	X		V		
IBM	F' = (P' x Q') + T'	FMAC	S	Z		U	V		
IBM	Round T to Integral Value	FRND	S	Z	X		V		
IBM	Reciprocal Seed of P	FRCP	S	Z			V		I
IBM	Convert T to Alt FP Format	FCTA	S	Z	X	U	V	R	
IBM	Convert T from Alt FP Format	FCFA	S	Z	X	U	V	R	I
Integer	F = P	IPAS	S	Z			V		
Integer	F = P + T	IADD	S	Z			V		C
Integer	F = P x Q	IMUL	S	Z			V		
Integer	Compare P, T	ICMP	S	=	>	<	V		C
Integer	Maximum P, T	IMAX	S	Z		W			
Integer	Minimum P, T	IMIN	S	Z		W			
Integer	Convert T to Floating-Point	ICTF	S	Z	X				
Integer	Scale T to Floating-Point	ISTF	S	Z	X	U	V	R	
Integer	F = P OR T	ILOR	S	Z					
Integer	F = P AND T	IAND	S	Z					
Integer	F = P XOR T	IXOR	S	Z					
Integer	Logical Shift P by Q Places	ILSH	S	Z					
Integer	Arithmetic Shift P by Q Places	IASH	S	Z			V		
Integer	Funnel Shift P T by Q Places	IFSH	S	Z					
	Move P	MOVE	S						
	Load Mode Register	LMRG							

Master/Slave Operation

Two Am29C327 processors can be tied together in master/slave configuration, with the slave checking the results produced by the master. All input and output signals of the slave, with the exception of $\overline{\text{SLAVE}}$ and MSERR, are tied to the corresponding signals of the master. The master is selected by asserting signal $\overline{\text{SLAVE}}$ LOW; the slave, by asserting signal $\overline{\text{SLAVE}}$ HIGH.

The slave processor, by comparing its outputs to the outputs of the master processor, performs a comprehensive check of the operation of the master processor. In addition, the slave processor may detect open circuits and other faults in the electrical path between the master processor and the system. Note that the master processor still performs the comparison between its outputs and its own internally generated results, and is therefore able to detect faults in its output drivers.

APPLICATIONS

Suggestions for Power and Ground Pin Connections

The Am29C327operates in an environment of fast signal rise times and substantial switching currents. Therefore, care must be exercised during circuit board design and layout, as with any high-performance component. The following is a suggested layout, but since systems vary widely in electrical configuration, an empirical evaluation of the intended layout is recommended.

The V_{CCO} and GNDO pins carry output driver switching currents and can be electrically noisy. The V_{CC} and GND pins, which supply the logic core of the device, tend to produce less noise, and the circuits they supply may be adversely affected by noise spikes on the V_{CC} plane. For this reason, it is best to provide isolation between the V_{CC} and V_{CCO} pins, as well as independent decoupling for each. Isolating the GND and GNDO pins is not required.

Printed Circuit-Board Layout Suggestions

1. Use of a multi-layer PC board with separate power, ground, and signal planes is highly recommended.

2. All V_{CC} and V_{CCO} pins should be connected to the V_{CC} plane. V_{CC} pins should be isolated from V_{CCO} pins by means of an isolation slot which is cut in the V_{CC} plane; see Figure 10. By physically separating the V_{CC} and V_{CCO} pins, coupled noise will be reduced.

3. All GND and GNDO pins should be connected directly to the ground plane.

4. The V_{CCO} pins should be decoupled to ground with a 0.1-μF ceramic capacitor and a 10-μF electrolytic capacitor, placed as closely to the Am29C327 as is practical. V_{CC} pins should be decoupled to ground in a similar manner.

A suggested layout is shown in Figure 10.

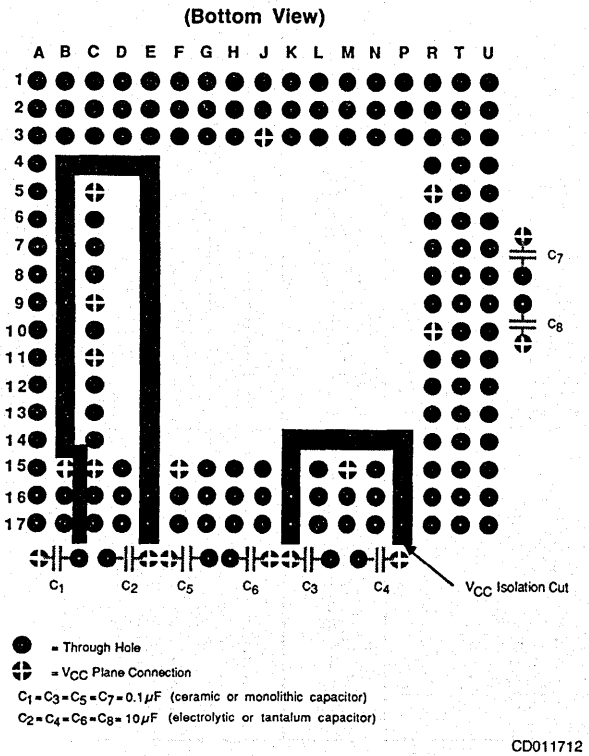


Figure 10. Suggested Printed Circuit-Board Layout

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Case Temperature Under Bias ..... −55 to +125°C
Supply Voltage to
Ground Potential Continuous ..... −0.3 V to +7.0 V
DC Voltage Applied to Outputs for
HIGH Output State ..... −0.3 V to +V_{CC} +0.3 V
DC Input Voltage ..... −0.3 V to +V_{CC} +0.3 V
DC Output Current, Into LOW Outputs ..... 30 mA
DC Input Current ..... −10 mA to +10 mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Case Temperature (T_C) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.75 V to +5.25 V

Military* (M) Devices
Case Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... +4.5 V to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

*Military Product 100% tested at T_C = +25°C, +125°C, and −55°C.

Thermal Resistance (Typical)

Symbol	CGX169	Unit
θ_{JA}	20	°C/W
θ_{JC}	4	°C/W

DC CHARACTERISTICS over operating ranges unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions (Note 1)			Min.	Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC} = Min. V_{IN} = V_{IH} or V_{IL}		I_{OH} = −4.0 mA	2.4		V
V_{OL}	Output LOW Voltage	V_{CC} = Min. V_{IN} = V_{IH} or V_{IL}		I_{OL} = 4.0 mA		0.5	V
V_{IH}	Input Logical HIGH Voltage (Note 2)				2.0		V
V_{IL}	Input Logical LOW Voltage (Note 2)					0.8	V
$V_{IH(F)}$	Guaranteed Input Logical HIGH Voltage (Note 2)	F Bus, Slave Operation only			V_{CC} − 0.5		V
$V_{IL(F)}$	Guaranteed Input Logical LOW Voltage (Note 2)	F Bus, Slave Operation only				0.5	V
I_{IL}	Input LOW Current	V_{CC} = Max. V_{IN} = 0.5 V				−10	μA
I_{IH}	Input HIGH Current	V_{CC} = Max. V_{IN} = V_{CC} − 0.5 V				10	μA
I_{OZH}	Off State (High Impedance) Output Current	V_{CC} = Max. V_O = 2.4 V				10	μA
I_{OZL}	Off State (High Impedance) Output Current	V_{CC} = Max. V_O = 0.5 V				−10	μA
I_{CC} Static	Static Power Supply Current	V_{CC} = Max. I_O = 0 μA	COM'L T_C = 0 to +70°C	(Note 3) CMOS V_{IN} = V_{CC} or GND		240	mA
				(Note 3) TTL V_{IN} = 0.5 V or 2.4 V		275	
			MIL T_C = −55 to +125°C	(Note 3) CMOS V_{IN} = V_{CC} or GND		285	
				(Note 3) TTL V_{IN} = 0.5 V or 2.4 V		335	
C_{PD}	Power Dissipation Capacitance (Note 4)	V_{CC} = Max. No Load				12,500	pF

Advance Information

Notes: 1. V_{CC} conditions shown as Min. or Max. refer to the applicable device type Operating Range.
2. These input levels provide zero-noise immunity and should only be statically tested in a noise-free environment (not functionally tested).
3. Use CMOS Static I_{CC} when the device is driven by CMOS circuits and TTL Static I_{CC} when the device is driven by TTL circuits.
4. C_{PD} determines the dynamic current consumption:

I_{CC} (Total) = I_{CC} (Static) + $(C_{PD} + nCL)\frac{f}{2}$, where f is the clock frequency, CL output load capacitance, and n number of loads.
This is tested on a sample basis.

CAPACITANCE*

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
C_I	Input Capacitance	f = 1 MHz		12	pF
C_O	Output Capacitance			20	
$C_{I/O}$	I/O Pin Capacitance			20	

*These capacitances are tested on a sample basis.

SWITCHING CHARACTERISTICS over COMMERCIAL operating operating range unless otherwise specified

No.	Parameter Description	Test Conditions	Am29C327		Am29C327-1		Am29C327-2		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	CLOCK Period	(Note 1)							
	Flow-Through Mode								
	Multiply-Accumulate		240	DC	200	DC	160	DC	ns
	All Other Operations		190	DC	160	DC	130	DC	ns
	Single-Pipelined Mode								
	Multiply-Accumulate		180	DC	160	DC	130	DC	ns
	All Other Operations		125	DC	100	DC	80	DC	ns
	Double-Pipelined Mode								
	Multiply-Accumulate		125	DC	100	DC	80	DC	ns
2	CLOCK LOW Time		15		12		10		ns
3	CLOCK HIGH Time		15		12		10		ns
4	Instruction Setup Time	(Note 2)	19		17		16		ns
5	Instruction Hold Time	(Note 2)	0		0		0		ns
6	Data Setup Time	(Note 3)	19		17		16		ns
7	Data Hold Time	(Note 3)	0		0		0		ns
8	Control Lines Setup Time	(Note 4)	18		17		16		ns
9	Control Lines Hold Time	(Note 4)	0		0		0		ns
10	F_{31-0} CLOCK-to-Output-Valid F Register Clocked			22		18		15	ns
11	$FLAG_{6-1}$ SIGN CLOCK-to-Output-Valid Status Register Clocked			17		15		13	ns
12	F_{31-0} CLOCK-to-Output-Valid F Register Transparent								
	Flow-Through Mode								
	Multiply-Accumulate			250		215		172	ns
	All Other Operations			200		170		136	ns
	Single-Pipelined Mode								
	Multiply-Accumulate			190		170		136	ns
	All Other Operations			135		110		88	ns
	Double-Pipelined Mode								
	Multiply-Accumulate			135		110		88	ns
13	$FLAG_{6-1}$ SIGN CLOCK-to-Output-Valid Status Register Transparent								
	Flow-Through Mode								
	Multiply-Accumulate			250		210		168	ns
	All Other Operations			195		170		136	ns
	Single-Pipelined Mode								
	Multiply-Accumulate			190		160		128	ns
	All Other Operations			115		100		80	ns
	Double-Pipelined Mode								
	Multiply-Accumulate			115		100		80	ns
14	$\overline{OEF}$, $\overline{OES}$, Disable Time HIGH to Z			18		16		14	ns
15	$\overline{OEF}$, $\overline{OES}$, Disable Time LOW to Z			18		16		14	ns
16	$\overline{OEF}$, $\overline{OES}$, Enable Time Z to HIGH			22		19		16	ns
17	$\overline{OEF}$, $\overline{OES}$, Enable Time Z to LOW			22		19		16	ns
18	FSEL to F_{31-0}			22		18		15	ns
19	MSERR Data-to-Valid Delay			29		27		25	ns

Notes: 1. CLOCK switching characteristics are made relative to 1.5 V.
2. Instruction signals include S/$\overline{DR}$, S/$\overline{DS}$, S/$\overline{DF}$, RM_{2-0}, $PSEL_{3-0}$, $QSEL_{3-0}$, $TSEL_{3-0}$, and I_{13-0}.
3. Data signals include R_{31-0} and S_{31-0}.
4. Control signals include $\overline{ENR}$, $\overline{ENS}$, $\overline{ENF}$, $\overline{ENRF}$, $RFSEL_{2-0}$, and $\overline{ENI}$.

Conditions: A. All inputs/outputs except CLOCK are TTL-compatible for V_{IH}, V_{IL}, and V_{OL}.
B. All outputs are driving 80 pF unless otherwise noted.
C. All setup, hold, and delay times are measured relative to CLOCK at $V_{CC}/2$ volts unless otherwise noted.

PRELIMINARY
Advance Information

SWITCHING CHARACTERISTICS over MILITARY operating range unless otherwise specified

No.	Parameter Description	Test Conditions	Am29C327		Am29C327-1		Am29C327-2		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	CLOCK Period	(Note 1)							
	Flow-Through Mode								
	Multiply-Accumulate		290	DC	245	DC	200	DC	ns
	All Other Operations		230	DC	195	DC	160	DC	ns
	Single-Pipelined Mode								
	Multiply-Accumulate		220	DC	195	DC	160	DC	ns
	All Other Operations		150	DC	120	DC	100	DC	ns
	Double-Pipelined Mode								
	Multiply-Accumulate		150	DC	120	DC	100	DC	ns
2	CLOCK LOW Time		18		15		12		ns
3	CLOCK HIGH Time		18		15		12		ns
4	Instruction Setup Time	(Note 2)	23		21		17		ns
5	Instruction Hold Time	(Note 2)	0		0		0		ns
6	Data Setup Time	(Note 3)	23		21		17		ns
7	Data Hold Time	(Note 3)	0		0		0		ns
8	Control Lines Setup Time	(Note 4)	22		21		17		ns
9	Control Lines Hold Time	(Note 4)	0		0		0		ns
10	F_{31-0} CLOCK-to-Output-Valid F Register Clocked			27		23		19	ns
11	$FLAG_{6-1}$ SIGN CLOCK-to-Output-Valid Status Register Clocked			21		19		16	ns
12	F_{31-0} CLOCK-to-Output-Valid F Register Transparent								
	Flow-Through Mode								
	Multiply-Accumulate			300		260		215	ns
	All Other Operations			240		205		170	ns
	Single-Pipelined Mode								
	Multiply-Accumulate			230		205		170	ns
	All Other Operations			165		135		110	ns
	Double-Pipelined Mode								
	Multiply-Accumulate			165		135		110	ns
13	$FLAG_{6-1}$ SIGN CLOCK-to-Output-Valid Status Register Transparent								
	Flow-Through Mode								
	Multiply-Accumulate			300		255		210	ns
	All Other Operations			235		205		170	ns
	Single-Pipelined Mode								
	Multiply-Accumulate			230		195		160	ns
	All Other Operations			140		125		100	ns
	Double-Pipelined Mode								
	Multiply-Accumulate			140		125		100	ns
14	$\overline{OEF}$, $\overline{OES}$, Disable Time HIGH to Z			22		20		16	ns
15	$\overline{OEF}$, $\overline{OES}$, Disable Time LOW to Z			22		20		16	ns
16	$\overline{OEF}$, $\overline{OES}$, Enable Time Z to HIGH			27		23		19	ns
17	$\overline{OEF}$, $\overline{OES}$, Enable Time Z to LOW			27		23		19	ns
18	FSEL to F_{31-0}			27		22		18	ns
19	MSERR Data-to-Valid Delay			34		33		27	ns

Notes: 1. CLOCK switching characteristics are made relative to 1.5 V.
2. Instruction signals include S/$\overline{DR}$, S/$\overline{DS}$, S/$\overline{DF}$, RM_{2-0}, $PSEL_{3-0}$, $QSEL_{3-0}$, $TSEL_{3-0}$, and I_{13-0}.
3. Data signals include R_{31-0} and S_{31-0}.
4. Control signals include $\overline{ENR}$, $\overline{ENS}$, $\overline{ENF}$, $\overline{ENRF}$, $RFSEL_{2-0}$, and $\overline{ENI}$.

Conditions: A. All inputs/outputs except CLOCK are TTL-compatible for V_{IH}, V_{IL}, and V_{OL}.
B. All outputs are driving 80 pF unless otherwise noted.
C. All setup, hold, and delay times are measured relative to CLOCK at $V_{CC}/2$ volts unless otherwise noted.

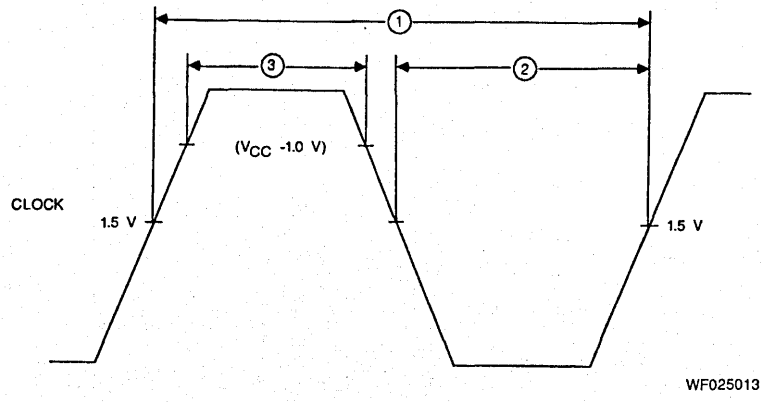
Advance Information

SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
———	MUST BE STEADY	WILL BE STEADY
———		
/\/\/\/	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
/\/\/\/	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
XXXXX	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
———	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

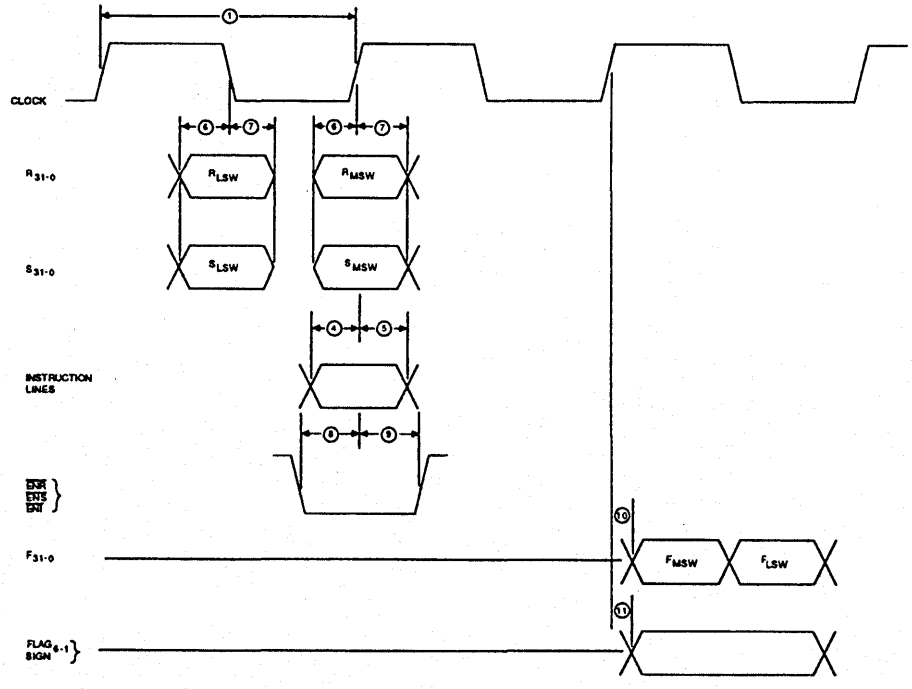
KS000010



WF025013

Input Clock Timing

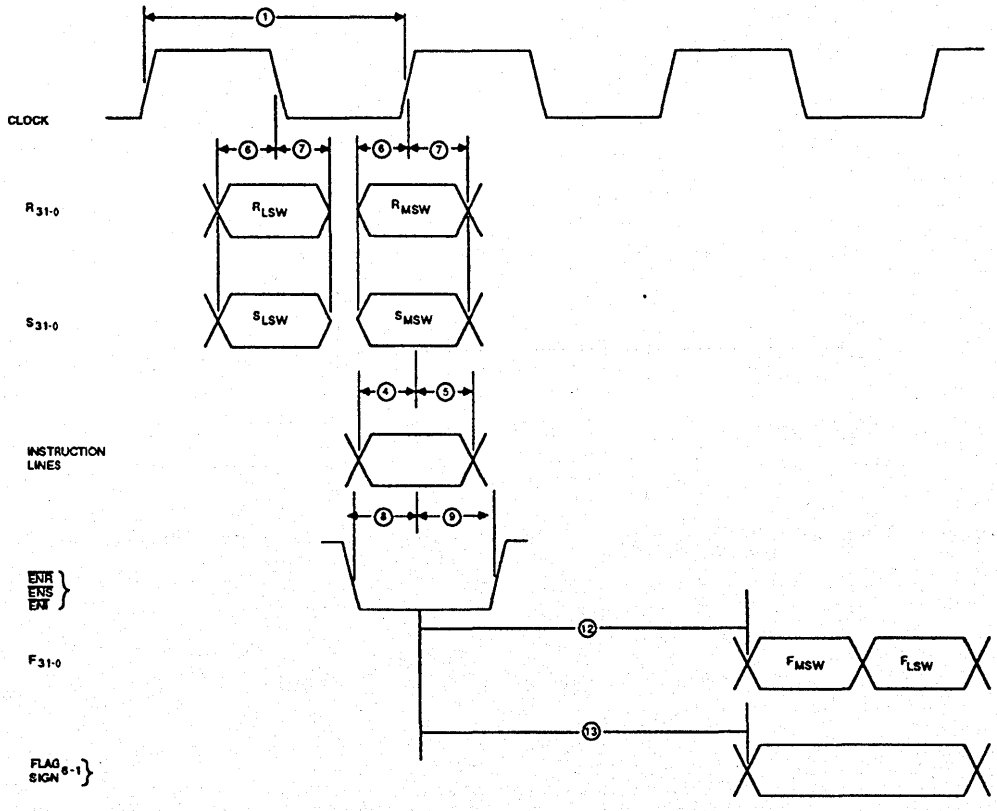
SWITCHING WAVEFORMS (Cont'd.)



WF025023

Timing of Operations with F Register and Status Clocked. Assumes 32-Bit Bus, Single-Cycle, LSW-First Input Mode and Flow-Through Operation

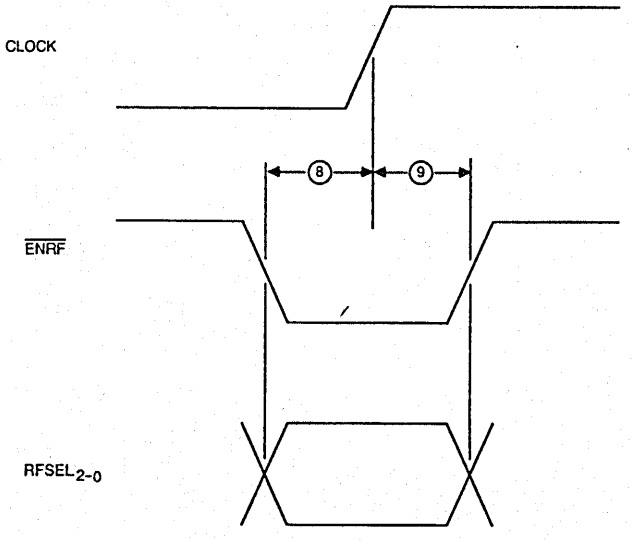
SWITCHING WAVEFORMS (Cont'd.)



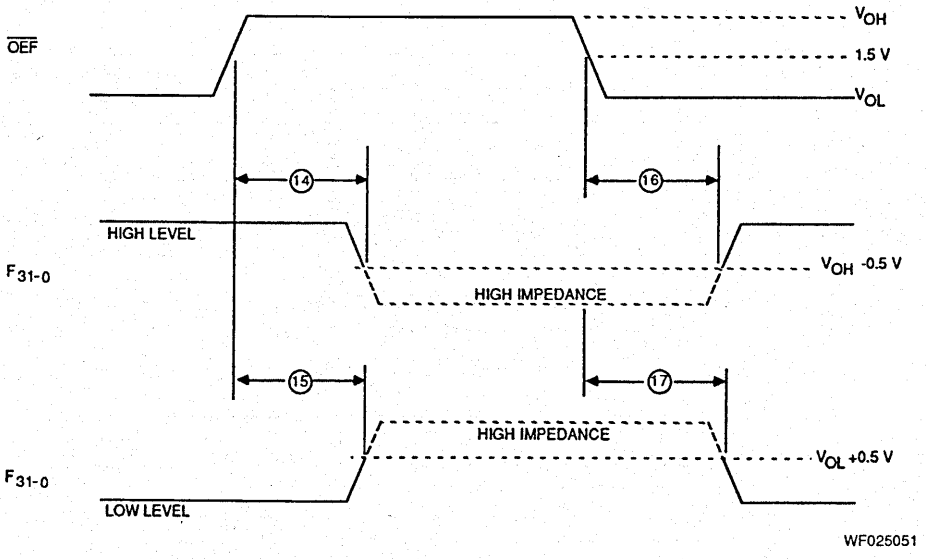
WF025033

Timing of Operations with F-Register and Status Register in Feedthrough Mode. Assumes 32-Bit Bus, Single-Cycle, LSW-First Input Mode and Flow-Through Operation.

SWITCHING WAVEFORMS (Cont'd.)

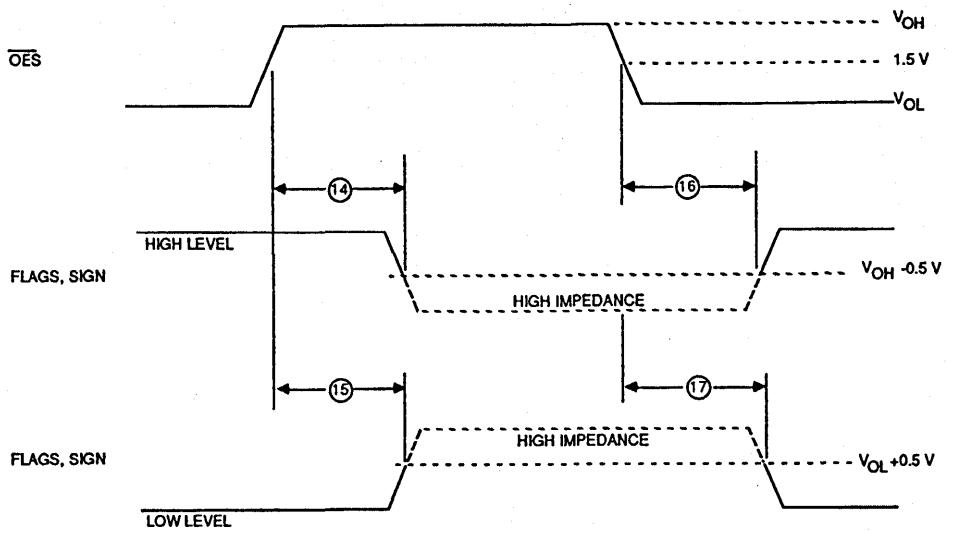


Register File Control Timing

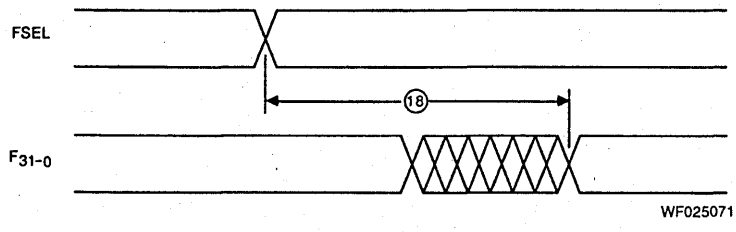


Enable/Disable Timing for F_{0-31}

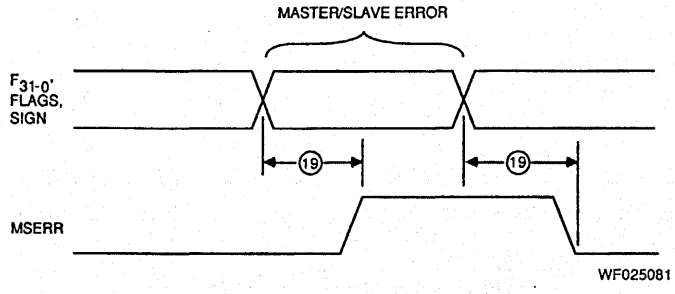
SWITCHING WAVEFORMS (Cont'd.)



Enable/Disable Timing for $FLAG_{1-6}$ and SIGN

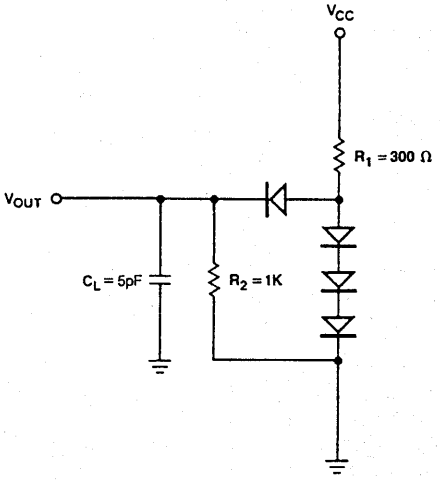


Output Selection Timing



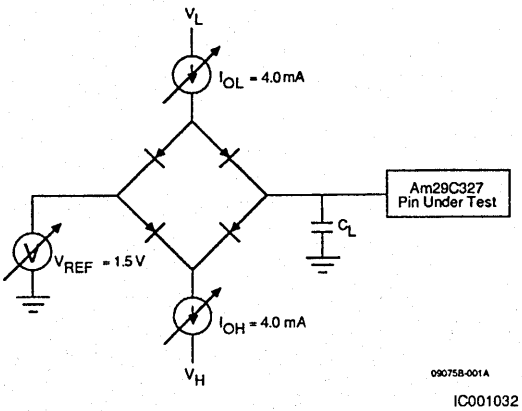
Master/Slave Timing (Assumes SLAVE Mode)

SWITCHING TEST CIRCUITS



TCR01334

A. Three-State Outputs

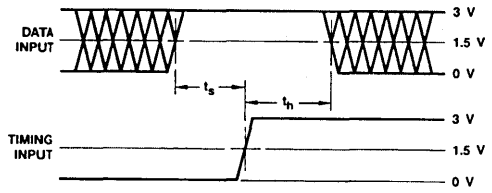


IC001032

B. All Other Outputs

2

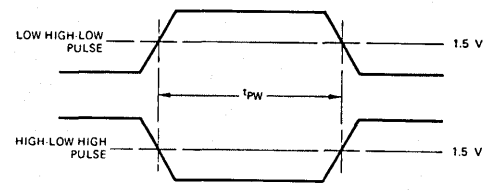
SWITCHING TEST WAVEFORMS



WFR02970

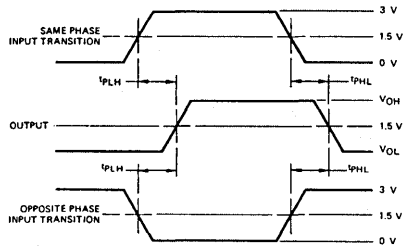
Notes: 1. Diagram shown for HIGH data only. Output transition may be opposite sense.
2. Cross-hatched area is don't care condition.

Setup, Hold, and Release Times



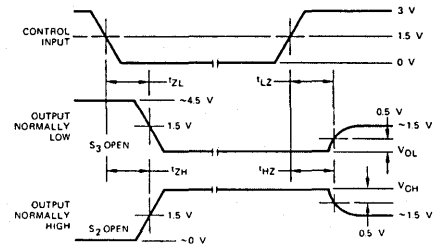
WFR02790

Pulse Width



WFR02980

Propagation Delay



WFR02660

Notes: 1. Diagram shown for Input Control Enable-LOW and Input Control Disable-HIGH.
2. S_1, S_2 and S_3 of Load Circuit are closed except where shown.

Enable and Disable Times

TEST PHILOSOPHY AND METHODS

The following eight points describe AMD's philosophy for high volume, high speed automatic testing.

1. Ensure that the part is adequately decoupled at the test head. Large changes in V_{CC} current as the device switches may cause erroneous function failures due to V_{CC} changes.
2. Do not leave inputs floating during any tests, as they may start to oscillate at high frequency.
3. Do not attempt to perform threshold tests at high speed. Following an output transition, ground current may change by as much as 400 mA in 5-8 ns. Inductance in the ground cable may allow the ground pin at the device to rise by hundreds of millivolts momentarily.
4. Use extreme care in defining point input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins and they may not actually reach V_{IL} or V_{IH} until the noise has settled. AMD recommends using $V_{IL} \leqslant 0$ V and $V_{IH} \geqslant 3.0$ V for AC tests.
5. To simplify failure analysis, programs should be designed to perform DC, Function, and AC tests as three distinct groups of tests.
6. Capacitive Loading for AC Testing

   Automatic testers and their associated hardware have stray capacitance that varies from one type of tester to another, but is generally around 50 pF. This, of course, makes it impossible to make direct measurements of parameters which call for smaller capacitive load than the associated stray capacitance. Typical examples of this are the so-called "float delays," which measure the propagation delays into the high-impedance state and are usually specified at a load capacitance of 5.0 pF. In these cases, the test is performed at the higher load capacitance (typically 50 pF), and engineering correlations based on data taken with a bench setup are used to predict the result at the lower capacitance.

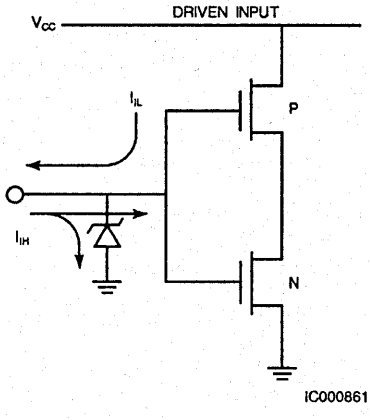
   Similarly, a product may be specified at more than one capacitive load. Since the typical automatic tester is not capable of switching loads in mid-test, it is impossible to make measurements at both capacitances even though they may both be greater than the stray capacitance. In these cases, a measurement is made at one of the two capacitances. The result at the other capacitance is predicted from engineering correlations based on data taken with a bench setup and the knowledge that certain DC measurements (I_{OH}, I_{OL} for example) have already been taken and are within spec. In some cases, special DC tests are performed in order to facilitate this correlation.
7. Threshold Testing

   The noise associated with automatic testing (due to the long, inductive cables) and the high gain of the tested device when in the vicinity of the actual device threshold, frequently give rise to oscillations when testing high speed circuits. These oscillations are not indicative of a reject device, but instead of an overtaxed test system. To minimize this problem, thresholds are tested at least once for each input pin. Thereafter, "hard" high and low levels are used for other tests. Generally this means that function and AC testing are performed at "hard" input levels rather than at V_{IL} Max. and V_{IH} Min.
8. AC Testing

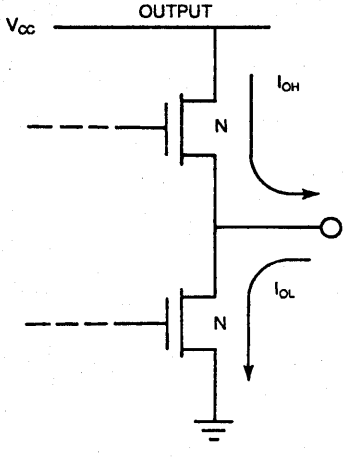
   Occasionally, parameters are specified that cannot be measured directly on automatic testers because of tester limitations. Data input hold times often fall into this category. In these cases, the parameter in question is guaranteed by correlating these tests with other AC tests that have been performed. These correlations are arrived at by the cognizant engineer by using precise bench measurements in conjunction with the knowledge that certain DC parameters have already been measured and are within spec.

   In some cases, certain AC tests are redundant, since they can be shown to be predicted by some other tests which have already been performed. In these cases, the redundant tests are not performed.

INPUT/OUTPUT CIRCUIT DIAGRAMS



IC000861



IC000870

APPENDICES

APPENDIX A — DATA FORMATS

The following data formats are supported: 32-bit integer, 64-bit integer, IEEE single-precision, IEEE double-precision, DEC F, DEC D, DEC G, IBM single-precision, and IBM double-precision.

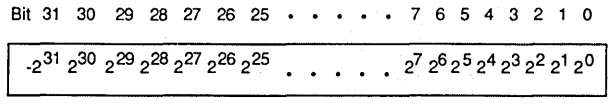
The primary and alternate floating-point formats are selected by mode register bits M3 to M0. The user may select between floating-point operations and integer operations by means of instruction bit I5.

The nine supported formats are described below:

Integer Formats

32-Bit Integer

The 32-bit integer word is arranged as follows:

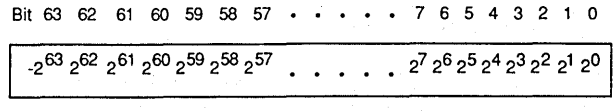


TB001030

The 32-bit word is interpreted as a two's-complement integer. For integer multiplications, the user has the option of interpreting integers as unsigned. An unsigned single-precision integer has a format similar to that of the two's-complement integer, but with an MSB weight of 2^{31}.

64-Bit Integer

The 64-bit integer word is arranged as follows:



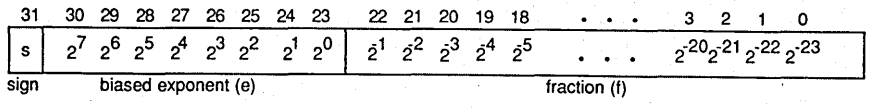
TB001040

The 64-bit word is interpreted as a two's-complement integer. For integer multiplications, the user has the option of interpreting integers as unsigned. An unsigned double-precision integer has a format similar to that of the two's-complement integer, but with an MSB weight of 2^{63}.

IEEE Formats

IEEE Single-Precision

The IEEE single-precision word is 32 bits wide and is arranged in the format as follows:



TB001050

The floating-point word is divided into three fields: a single-bit sign, an 8-bit biased exponent, and a 23-bit fraction.

The sign bit is 0 for positive numbers and 1 for negative numbers. Zero may have either sign.

The biased exponent is an 8-bit unsigned integer representing a multiplicative factor of some power of two. The bias value is 127. If, for example, the multiplicative value for a floating-point number is to be 2^a, the value of the biased exponent is a + 127, where "a" is the true exponent.

The fraction is a 23-bit unsigned fractional field containing the 23 least-significant bits of the floating-point number's 24-bit mantissa. The weight of the fraction's most-significant bit is 2^{-1}. The weight of the least-significant bit is 2^{-23}.

An IEEE floating-point number is evaluated or interpreted as follows:

If e = 255 and f ≠ 0 ..... value = NaN	Not-a-Number
If e = 255 and f = 0 ..... value = $(-1)^S\infty$	Infinity
If 0 < e < 255..... value = $(-1)^S 2^{e-127}(1.f)$	Normalized number
If e = 0 and f ≠ 0 ..... value = $(-1)^S 2^{-126}(0.f)$	Denormalized number
If e = 0 and f = 0..... value = $(-1)^S 0$	Zero

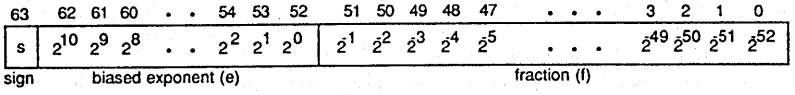
Infinity: Infinity can have either a positive or negative sign. The interpretation of infinities is determined by the Affine/Projective select input AFF/$\overline{\text{PROJ}}$.

NaN: A NaN is interpreted as a signal or symbol. NaNs are used to indicate invalid operations, and as a means of passing process status through a series of calculations. They arise in two ways: either generated by the Am29C327 to indicate an invalid operation, or provided by the user as an input. A signaling NaN has the MSB of its fraction set to 0 and at least one of the remaining fraction bits set to 1. A quiet NaN has the MSB of its fraction set to 1.

The IEEE format is fully described in ANSI/IEEE Standard 754-1985.

IEEE Double-Precision

The IEEE double-precision word is 64 bits wide and is arranged in the format shown below:



TB001060

The floating-point word is divided into three fields: a single-bit sign, an 11-bit biased exponent, and a 52-bit fraction.

The sign bit is 0 for positive numbers and 1 for negative numbers; zero may have either sign.

The biased exponent is an 11-bit unsigned integer representing a multiplicative factor of some power of two. The bias value is 1023. If, for example, the multiplicative value for a floating-point number is to be 2^a, the value of the biased exponent is a + 1023, where "a" is the true exponent.

The fraction is a 52-bit unsigned fractional field containing the 52 least-significant bits of the floating-point number's 53-bit mantissa. The weight of the fraction's most-significant bit is 2^{-1}. The weight of the least-significant bit is 2^{-52}.

An IEEE floating-point number is evaluated or interpreted as follows:

If e = 2047 and f ≠ 0	value = Reserved operand	Not-a-Number
If e = 2047 and f = 0	value = $(-1)^s\infty$	Infinity
If 0 < e < 2047	value = $(-1)^s 2^{e-1023}(1.f)$	Normalized number
If e = 0 and f ≠ 0	value = $(-1)^s 2^{-1022}(0.f)$	Denormalized number
If e = 0 and f = 0	value = $(-1)^s 0$	Zero

Infinity: Infinity can have either a positive or negative sign. The interpretation of infinities is determined by the Affine/Projective select input AFF/$\overline{\text{PROJ}}$.

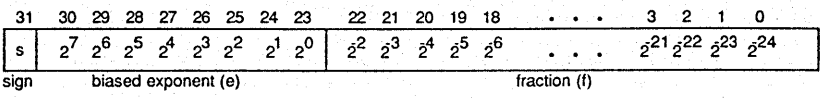
NaN: A NaN is interpreted as a signal or symbol. NaNs are used to indicate invalid operations, and as a means of passing process status through a series of calculations. They arise in two ways: either generated by the Am29C327 to indicate an invalid operation, or provided by the user as an input. A signaling NaN has the MSB of its fraction set to 0 and at least one of the remaining fraction bits set to 1. A quiet NaN has the MSB of its fraction set to 1.

The IEEE format is fully described in ANSI/IEEE Standard 754-1985.

DEC Formats

DEC F

The DEC F word is 32 bits wide and is arranged in the format shown below:



TB001070

The floating-point word is divided into three fields: a single-bit sign, an 8-bit biased exponent, and a 23-bit fraction.

The sign bit is 0 for positive numbers and 1 for negative numbers; zero has a positive sign.

The biased exponent is an 8-bit unsigned integer representing a multiplicative factor of some power of two. The bias value is 128. If, for example, the multiplicative value for a floating-point number is to be 2^a, the value of the biased exponent is a + 128, where "a" is the true exponent.

The fraction is a 23-bit unsigned fractional field containing the 23 least-significant bits of the floating-point number's 24-bit mantissa. The weight of the fraction's most-significant bit is 2^{-2}. The weight of the least-significant bit is 2^{-24}.

A DEC F floating-point number is evaluated or interpreted as follows:

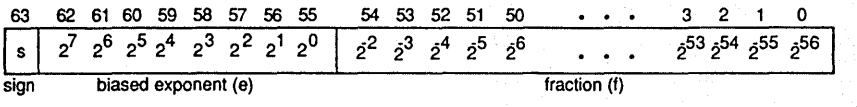
If e ≠ 0	value ≠ $(-1)^s 2^{e-128}(0.1f)$
If s = 0 and e = 0	value = 0
If s = 1 and e = 0	value = DEC-Reserved Operand

DEC-Reserved Operand: A DEC-Reserved Operand is interpreted as a signal or symbol. DEC-Reserved Operands are used to indicate invalid operations and operations whose results have overflowed the destination format. They may also be used to pass symbolic information from one calculation to another.

The DEC formats are fully described in the VAX Architecture Manual.

DEC D

The DEC D word is 64 bits wide and is arranged in the format shown below:



TB001080

The floating-point word is divided into three fields: a single-bit sign, an 8-bit biased exponent, and a 55-bit fraction.

The sign bit is 0 for positive numbers and 1 for negative numbers; zero has a positive sign.

The biased exponent is an 8-bit unsigned integer representing a multiplicative factor of some power of two. The bias value is 128. If, for example, the multiplicative value for a floating-point number is to be 2^a, the value of the biased exponent is a + 128, where "a" is the true exponent.

The fraction is a 55-bit unsigned fractional field containing the 55 least-significant bits of the floating-point number's 56-bit mantissa. The weight of the fraction's most-significant bit is 2^{-2}. The weight of the least-significant bit is 2^{-56}.

A DEC D floating-point number is evaluated or interpreted as follows:

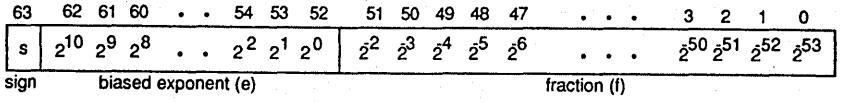
If $e \neq 0$ ..... value = $(-1)^s 2^{e-128}(0.1f)$
If s = 0 and e = 0..... value = 0
If s = 1 and e = 0..... value = DEC-Reserved Operand

DEC-Reserved Operand: A DEC-Reserved Operand is interpreted as a signal or symbol. DEC-Reserved Operands are used to indicate invalid operations and operations whose results have overflowed the destination format. They may also be used to pass symbolic information from one calculation to another.

The DEC formats are fully described in the VAX Architecture Manual.

DEC G

The DEC G word is 64 bits wide and is arranged in the format shown below:



TB001090

The floating-point word is divided into three fields: a single-bit sign, an 11-bit biased exponent, and a 52-bit fraction.

The sign bit is 0 for positive numbers and 1 for negative numbers; zero has a positive sign.

The biased exponent is an 11-bit unsigned integer representing a multiplicative factor of some power of two. The bias value is 1024. If, for example, the multiplicative value for a floating-point number is to be 2^a, the value of the biased exponent is a + 1024, where "a" is the true exponent.

The fraction is a 52-bit unsigned fractional field containing the 52 least-significant bits of the floating-point number's 53-bit mantissa. The weight of the fraction's most-significant bit is 2^{-2}. The weight of the least-significant bit is 2^{-53}.

A DEC G floating-point number is evaluated or interpreted as follows:

If $e \neq 0$ ..... value = $(-1)^s 2^{e-1024}(0.1f)$
If s = 0 and e = 0..... value = 0
If s = 1 and e = 0..... value = DEC-Reserved Operand

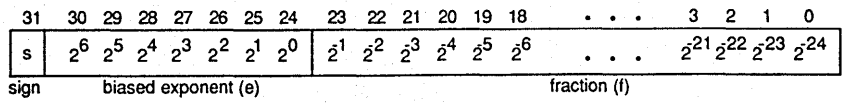
DEC-Reserved Operand: A DEC-Reserved Operand is interpreted as a signal or symbol. DEC-Reserved Operands are used to indicate invalid operations and operations whose results have overflowed the destination format. They may also be used to pass symbolic information from one calculation to another.

The DEC formats are fully described in the VAX Architecture Manual.

IBM Formats

IBM Single-Precision

The IBM single-precision word is 32 bits wide and is arranged in the format shown below:



TB001100

The floating-point word is divided into three fields: a single-bit sign, a 7-bit biased exponent, and a 24-bit fraction.

The sign bit is 0 for positive numbers and 1 for negative numbers; a True-zero has a positive sign.

The biased exponent is a 7-bit unsigned integer representing a multiplicative factor of some power of 16. The bias value is 64. If, for example, the multiplicative value for a floating-point number is to be 16^a, the value of the biased exponent is a + 64, where "a" is the true exponent.

The fraction is a 24-bit unsigned fractional field containing the 24 least-significant bits of the floating-point number's 25-bit mantissa. The weight of the fraction's most-significant bit is 2^{-1}. The weight of the least-significant bit is 2^{-24}.

An IBM floating-point number is evaluated or interpreted as follows:

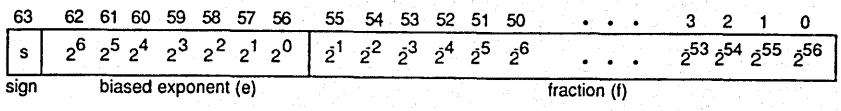
$$\text{value} = (-1)^s 16^{e-64}(0.f)$$

Zero: There are two classes of zero. If the sign, biased exponent and fraction are all zero, the operand is known as a "True Zero." If the fraction is zero, but the sign and biased exponent are not both zero, the operand is known as a "Floating-point Zero."

The IBM format is fully described in the IBM System/370 Principles of Operation Manual.

IBM Double-Precision

The IBM double-precision word is 64 bits wide and is arranged in the format shown below:



TB001110

The floating-point word is divided into three fields: a single-bit sign, a 7-bit biased exponent, and a 56-bit fraction.

The sign bit is 0 for positive numbers and 1 for negative numbers; a True-zero has a positive sign.

The biased exponent is a 7-bit unsigned integer representing a multiplicative factor of some power of 16. The bias value is 64. If, for example, the multiplicative value for a floating-point number is to be 16^a, the value of the biased exponent is a + 64, where "a" is the true exponent.

The fraction is a 56-bit unsigned fractional field containing the 56 least-significant bits of the floating-point number's 57-bit mantissa. The weight of the fraction's most-significant bit is 2^{-1}. The weight of the least-significant bit is 2^{-56}.

An IBM floating-point number is evaluated or interpreted as follows:

$$\text{value} = (-1)^s 16^{e-64}(0.f)$$

Zero: There are two classes of zero. If the sign, biased exponent and fraction are all zero, the operand is known as a "True Zero." If the fraction is zero, but the sign and biased exponent are not both zero, the operand is known as a "Floating-point Zero."

The IBM format is fully described in the IBM System/370 Principles of Operation Manual.

APPENDIX B — ROUNDING MODES

The Am29C327 provides six rounding modes for floating-point operations, and for integer multiplication. The rounding mode for an operation is selected by the input pins RM_2-RM_0.

RM_2	RM_1	RM_0	Round Mode
0	0	0	Round to Nearest (IEEE)
0	0	1	Round to Minus Infinity
0	1	0	Round to Plus Infinity
0	1	1	Round to Zero
1	0	0	Round to Nearest (DEC)
1	0	1	Round Away From Zero
1	1	X	Illegal Value

Round to Nearest (IEEE)

The infinitely precise result of an operation is rounded to the closest representable value in the destination format. If the infinitely precise result is exactly halfway between two representations, it is rounded to the representation having a least-significant bit of zero.

Round to Minus Infinity (IEEE)

The infinitely precise result of an operation is rounded to the closest representable value in the destination format that is less than or equal to the infinitely precise result. This rounding mode conforms to the "round to minus infinity" mode described in the IEEE Floating-Point Standard.

The IEEE standard specifies that all four "IEEE" modes be available so that the user may select the mode most appropriate for the algorithm being executed. The DEC standard specifies that two rounding modes be available - Round-to-Nearest (DEC) and Round-to-Zero. The IBM standard specifies that all operations be performed using the Round-to-Zero mode.

It should be noted, however, that the Am29C327 permits any of the supported rounding modes to be selected, regardless of the format of the operation. It is permissable to use one of the IEEE rounding modes with an IBM operation, or DEC rounding with an IEEE operation, or any other possible combination. For those integer operations where rounding is performed, any rounding mode may be chosen. This flexibility allows the user to select the mode most appropriate for the arithmetic environment in which the processor is operating.

Round to Plus Infinity (IEEE)

The infinitely precise result of an operation is rounded to the closest representable value in the destination format that is greater than or equal to the infinitely precise result.

Round to Zero (IEEE)

The infinitely precise result of an operation is rounded to the closest representable value in the destination format whose magnitude is less than or equal to the infinitely precise result.

Round to Nearest (DEC)

The infinitely precise result of an operation is rounded to the closest representable value in the destination format. If the infinitely precise result is exactly halfway between two representations, it is rounded to the representation having the greater magnitude.

Round Away from Zero

The infinitely precise result of an operation is rounded to the closest representable value in the destination format whose magnitude is greater than or equal to the infinitely precise result.

A graphical representation of these rounding modes is shown in Figures B1-1 and B1-2.

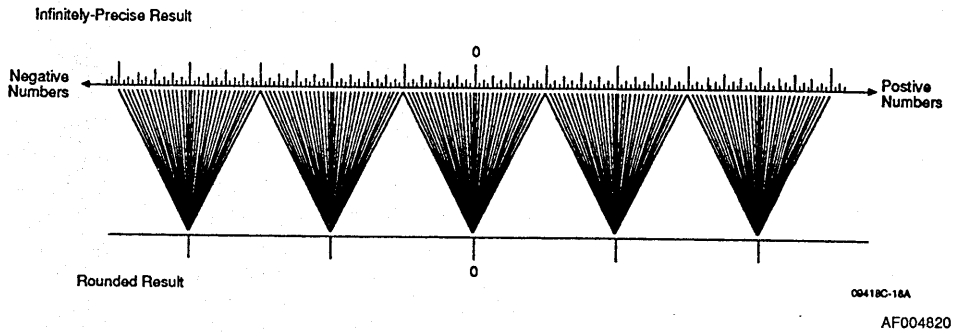


Figure B1. Illustration of IEEE Round-to-Nearest Rounding Mode

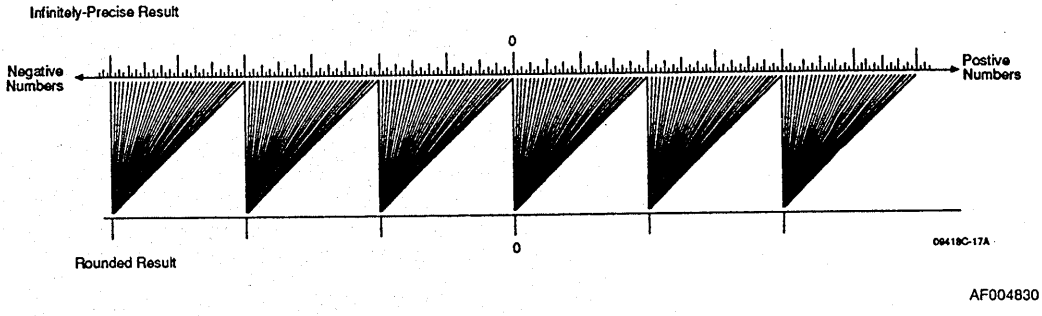


Figure B2. Illustration of IEEE Round-to-Minus-Infinity Rounding Mode

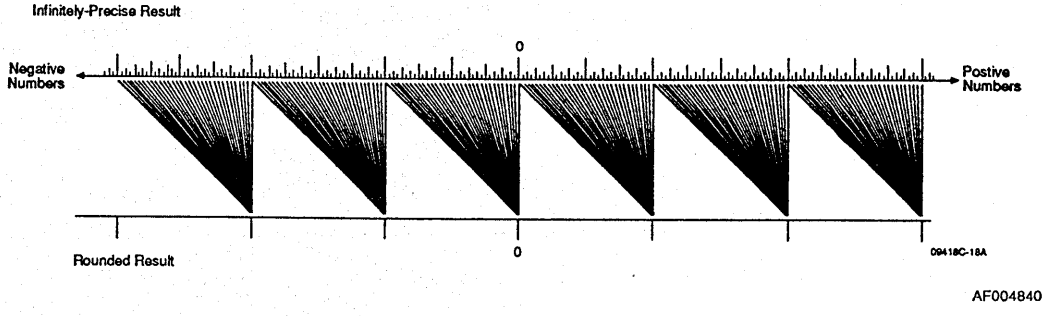


Figure B3. Illustration of IEEE Round-to-Plus-Infinity Rounding Mode

2

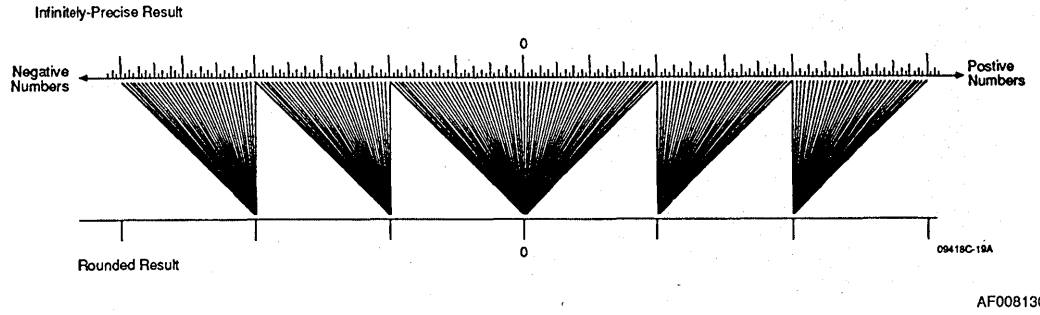


Figure B4. Illustration of IEEE Round-to-Zero Rounding Mode

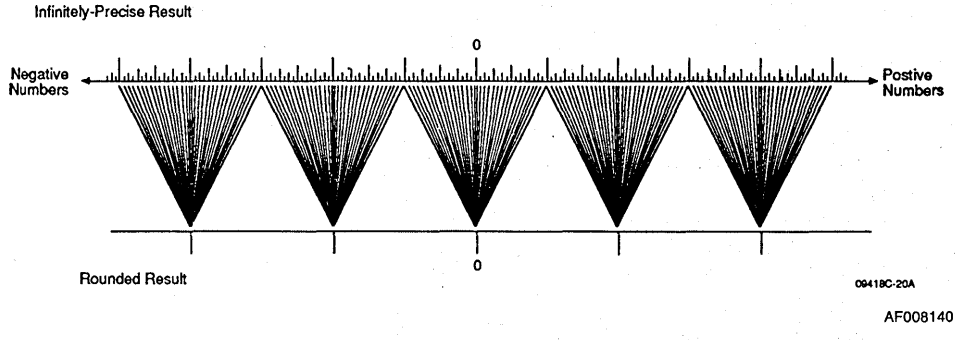


Figure B5. Illustration of DEC Round-to-Nearest Rounding Mode

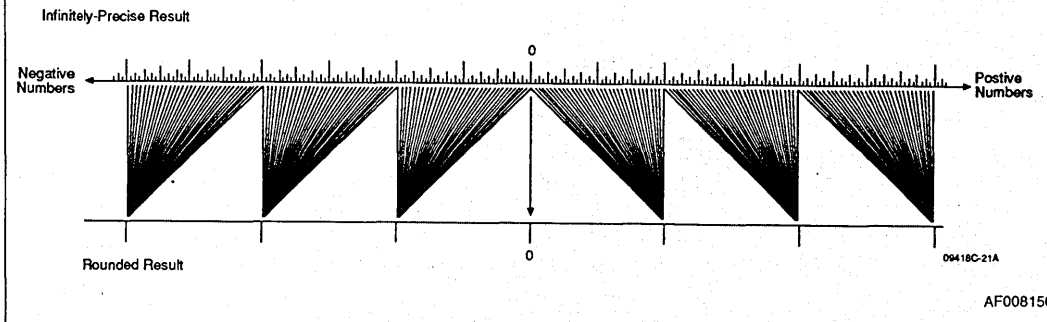


Figure B6. Illustration of Round-Away-From-Zero Rounding Mode

APPENDIX C — DEVIATIONS FROM FLOATING-POINT STANDARDS

There are several cases in which the implementation of the IEEE, DEC and IBM floating-point standards in the Am29C327 differs from the formal definitions of those standards. This appendix describes these deviations from the standards.

Deviations from the IEEE Standard

Section 7.3 of the IEEE-754 standard specifies that "Trapped overflow on conversion from a binary floating-point format shall deliver to the trap handler a result in that or a wider format, possibly with the exponent bias adjusted, but rounded to the destination's precision."

According to the IEEE standard, then, if a double-to-single IEEE operation overflows while traps are enabled, the result is a double-precision operand, rounded to single-precision width (23-bit fraction), together with a correctly-adjusted (double-precision) exponent and the appropriate flags for a trapped overflow.

In the case of an overflow in any IEEE operation, the Am29C327 returns a result in the destination format specified by the user, rounded to that destination format.

In the case of the double-to-single overflow described above, the result from the Am29C327 is a single-precision operand, together with a correctly-adjusted (single-precision) exponent and the appropriate flags for a trapped overflow.

A simple example serves to illustrate the discrepancy, by describing the conversion of the double-precision IEEE number 52B123456789ABCD to single-precision, with traps enabled and the round-to-nearest rounding mode selected. This number is too large to be represented in single-precision format.

According to the IEEE standard, the result of this operation is the double-precision number 52B1234560000000, comprising the double-precision exponent of the input and a fraction truncated to 23 bits, together with flags V and X.

When the operation is performed in the Am29C327, however, using the " F' = P' " operation with appropriate precision controls, the result is the single-precision number 75891A2B, comprising the single-precision (overflowed) exponent reduced by 192 (decimal) and a single-precision fraction, together with flags V and X.

It should be noted that trapped operation is an optional part of the IEEE standard. Full adherence to the IEEE specification of trapped operation is therefore not necessary to ensure compliance with IEEE-754.

Deviations from the DEC F, DEC D and DEC G Standards

The DEC F, DEC D and DEC G standards, as implemented in the Am29C327, differ from the implementations in a VAX only in the way in which the sub-fields of the floating-point word are arranged. The differences are listed in Table C1.

TABLE C1. DIFFERENCES IN Am29C327 AND DEC FLOATING-POINT FORMATS

	Am29C327 Arrangement	VAX Arrangement
DEC F	sign: bit 31 exponent: bits 30-23 fraction: bits 22-0	sign: bit 15 exponent: bits 14-7 fraction: bits 6-0, bits 31-16
DEC D	sign: bit 63 exponent: bits 62-55 fraction: bits 54-0	sign: bit 15 exponent: bits 14-7 fraction: bits 6-0, bits 31-16, bits 47-32, bits 63-48
DEC G	sign: bit 63 exponent: bits 62-52 fraction: bits 51-0	sign: bit 15 exponent: bits 14-4 fraction: bits 3-0, bits 31-16, bits 47-32, bits 63-48

The discrepancies are shown graphically in Figure C1. Within each exponent and fraction field, the shading illustrates the weighting of the bits, from the MSB to LSB.

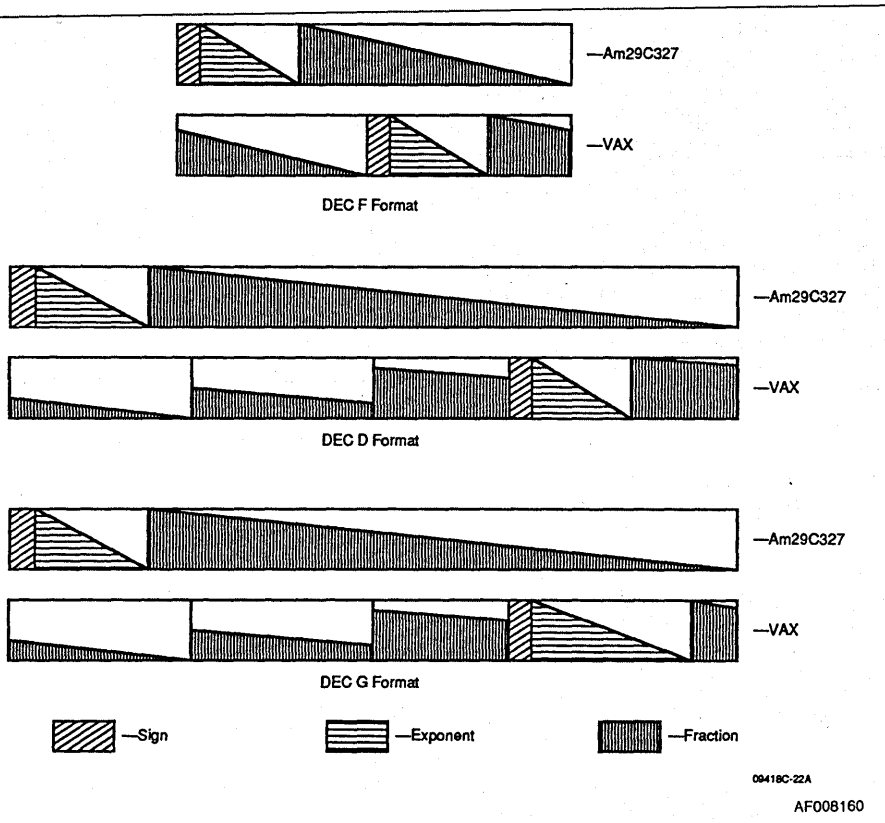
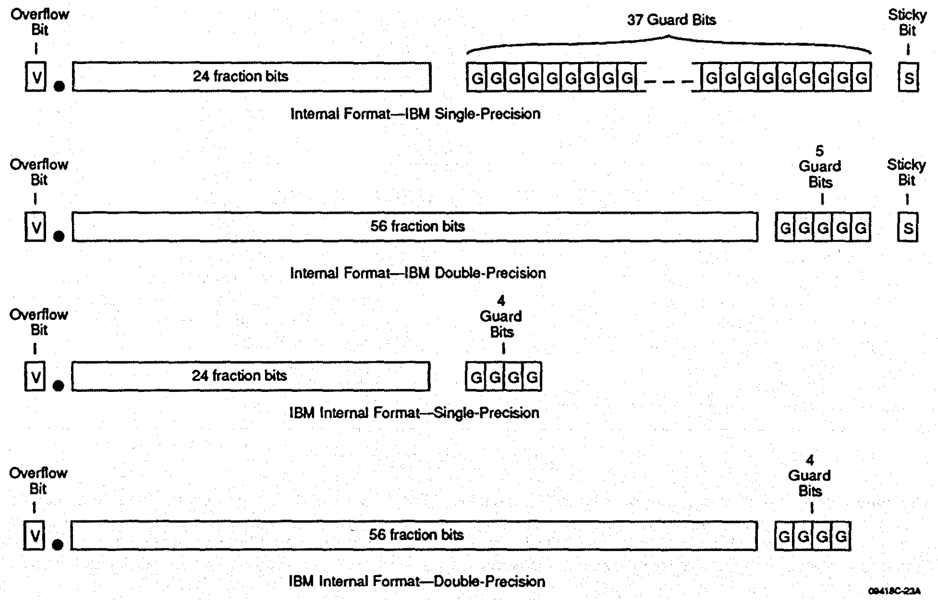


Figure C1. Differences in DEC Representations Between the Am29C327 and the VAX

Deviations from the IBM Standard

The Am29C327's deviations from the IBM standard may be summarized as follows, assuming that the user has selected the round-to-nearest rounding mode:

1. The Am29C327 provides more guard bits in its internal format than specified by the IBM standard. With certain combinations of input operands, the Am29C327 produces more accurate results than a standard IBM processor for instructions based on addition operations and comparisons (i.e. IBM instructions with an op-code, I4-0, of 00001 or 00011).
2. The discrepancies are much larger for single-precision operations than double-precision operations, because the difference in the number of guard bits is much greater (33 more for single, one more for double).
3. There is no universal rule for determining whether a given set of input operands will result in a discrepancy. Provided the conditions in (1) above are met, the user must examine each operation on a case-by-case basis, taking into account the input operands and the internal formats discussed in this section.
4. The Am29C327 does not produce unnormalized results from additions. The results of all addition operations are renormalized.



TB001230

Figure C2. Differences in Internal Mantissa Formats of an IBM CPU and the Am29C327

It should be mentioned that the discrepancies due to the above effects, for both additions and multiplications, are typically insignificant when compared to the magnitude of the result itself. They become an important issue only when the user requires exact compatibility with an existing implementation of the IBM standard.

The term "accuracy" in this discussion is used to describe the difference between the final result of an operation and its infinitely-precise result. In the context of adherence to a standard, however, "accuracy" might better be defined as the difference between the final result of an operation executed in the Am29C327 and the final result of the same operation executed in a CPU which exactly meets that standard.

2

Am9513A

System Timing Controller

FINAL

DISTINCTIVE CHARACTERISTICS

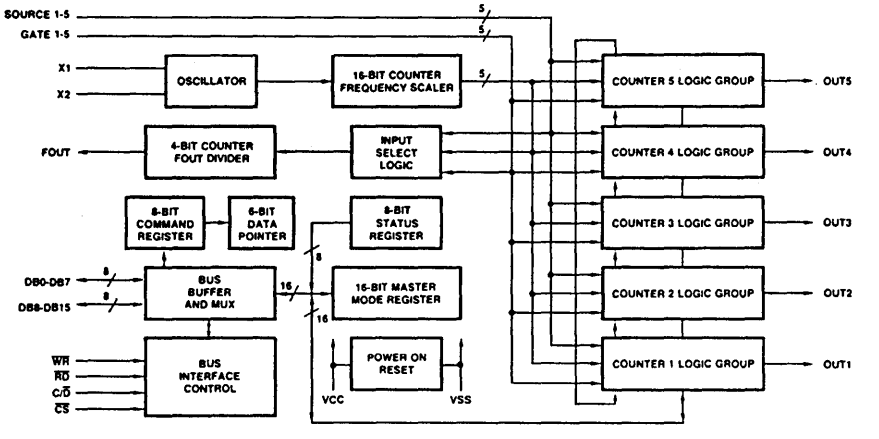
- Five independent 16-bit counters
- High speed counting rates
- Up/down and binary/BCD counting
- Internal oscillator frequency source
- Tapped frequency scaler
- Programmable frequency output
- 8-bit or 16-bit bus interface
- Time-of-day option
- Alarm comparators on counters 1 and 2
- Complex duty cycle outputs
- One-shot or continuous outputs
- Programmable count/gate source selection
- Programmable input and output polarities
- Programmable gating functions
- Retriggering capability
- +5 volt power supply
- Standard 40-pin package
- SMD/DESC qualified

GENERAL DESCRIPTION

The Am9513A System Timing Controller is an LSI circuit designed to service many types of counting, sequencing and timing applications. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital one-shots, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis, etc. A variety of programmable operating modes and control features allows the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide pulses or levels and can be active-high or active-low. The counters can be programmed to count up or down in either binary or BCD. The host processor may read an accumulated count at any time without disturbing the counting process. Any of the counters may be internally concatenated to form any effective counter length up to 80 bits.

BLOCK DIAGRAM

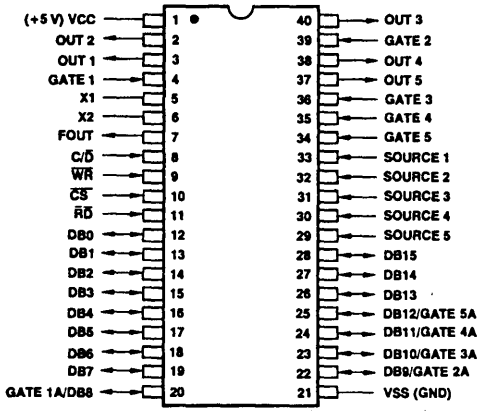


BD003381

Publication # 01731 Rev. F Amendment /0
Issue Date: August 1989

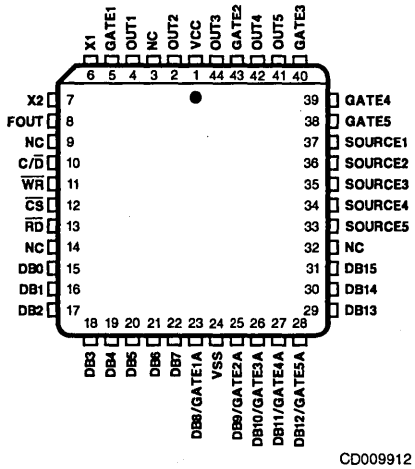
CONNECTION DIAGRAMS
Top View

DIP



CD005212

LCC & PLCC



CD009912

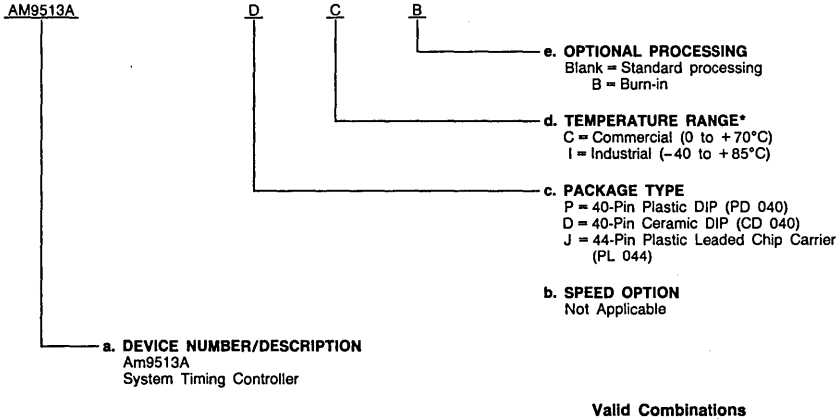
Note: Pin 1 is marked for orientation.

2

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9513A	PC, DC, DCB, DIB, JC

*This device is also available in Military temperature range.

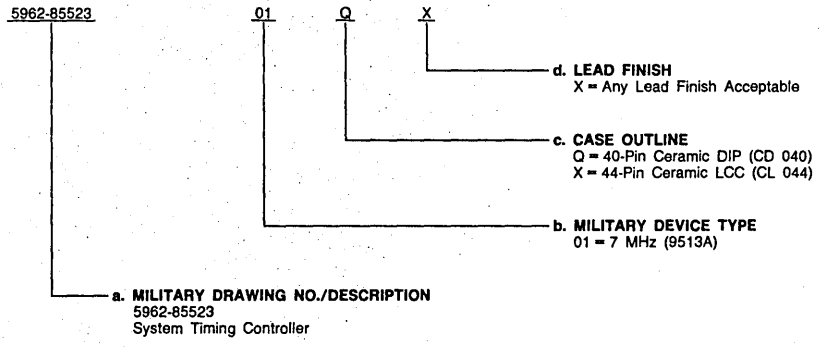
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION (continued)

Standard Military Drawing (SMD)/DESC Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. Standard Military Drawing (SMD)/DESC products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for SMD/DESC products is formed by a combination of: **a. Military Drawing Part Number**
b. Device Type
c. Case Outline
d. Lead Finish



Valid Combinations	
5962-8552301	QX, XX

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

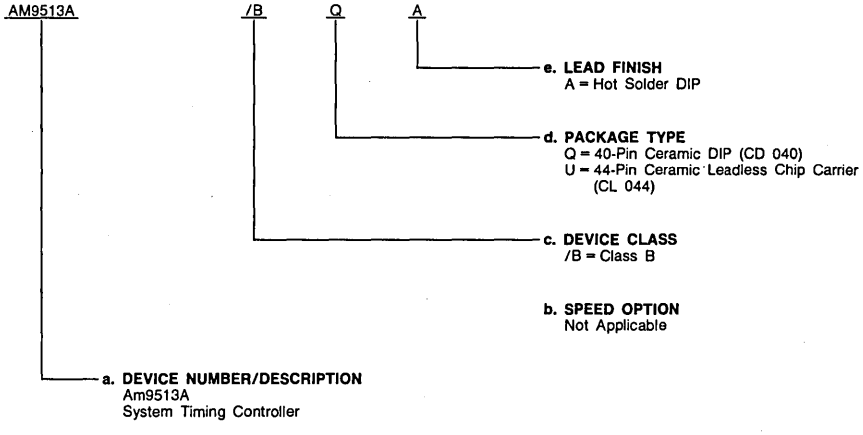
Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

ORDERING INFORMATION (continued)

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM9513A	/BQA, /BUA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of
Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1	VCC		+5 V Power Supply.
21	VSS		Ground.
5, 6	X1, X2	O, I	(Crystal). X1 and X2 are the connections for an external crystal used to determine the frequency of the internal oscillator. The crystal should be a parallel-resonant, fundamental-mode type. An RC or LC or other reactive network may be used instead of a crystal. For driving from an external frequency source, X1 should be left open and X2 should be connected to a TTL source and a pull-up resistor.
7	FOUT	O	(Frequency Out). The FOUT output is derived from a 4-bit counter that may be programmed to divide its input by any integer value from 1 through 16 inclusive. The input to the counter is selected from any of 15 sources, including the internal scaled oscillator frequencies. FOUT may be gated on and off under software control and when off will exhibit a low impedance to ground. Control over the various FOUT options resides in the Master Mode register. After power-up, FOUT provides a frequency that is 1/16 that of the oscillator. The input source on power-up is F1.
4, 39, 36 – 34	GATE1 – GATE5	I	(Gate). The Gate inputs may be used to control the operations of individual counters by determining when counting may proceed. The same Gate input may control up to three counters. Gate pins may also be selected as count sources for any of the counters and for the FOUT divider. The active polarity for a selected Gate input is programmed at each counter. Gating function options allow level-sensitive gating or edge-initiated gating. Other gating modes are available including one that allows the Gate input to select between two counter output frequencies. All gating functions may also be disabled. The active Gate input is conditioned by an auxiliary input when the unit is operating with an external 8-bit data bus. See Data Bus description. Schmitt-trigger circuitry on the GATE inputs allows slow transition times to be used.
33 – 29	SRC1 – SRC5	I	(Source).The Source inputs provide external signals that may be counted by any of the counters. Any Source line may be routed to any or all of the counters and the FOUT divider. The active polarity for a selected SRC input is programmed at each counter. Any duty cycle waveform will be accepted as long as the minimum pulse width is at least half the period of the maximum specified counting frequency for the part. Schmitt-trigger circuitry on the SRC inputs allows slow transition times to be used.
3, 2, 40, 38, 37	OUT1 – OUT5	O	(Counter). Each 3-state OUT signal is directly associated with a corresponding individual counter. Depending on the counter configuration, the OUT signal may be a pulse, a square wave, or a complex duty cycle waveform. OUT pulse polarities are individually programmable. The output circuitry detects the counter state that would have been all bits zero in the absence of a reinitialization. That information is used to generate the selected waveform type. An optional output mode for Counters 1 and 2 overrides the normal output mode and provides a true OUT signal when the counter contents match the contents of an Alarm register.
12 – 19, 20, 22 – 28	DB0 – DB7, DB8 – DB15	I/O	(Data Bus). The 16 bidirectional Data Bus lines are used for information exchanges with the host processor. HIGH on a Data Bus line corresponds to one and LOW corresponds to zero. These lines act as inputs when $\overline{WR}$ and $\overline{CS}$ are active and as outputs when $\overline{RD}$ and $\overline{CS}$ are active. When $\overline{CS}$ is inactive, these pins are placed in a high-impedance state. After power-up or reset, the data bus will be configured for 8-bit width and will use only DB0 through DB7. DB0 is the least significant and DB7 is the most significant bit position. The data bus may be reconfigured for 16-bit width by changing a control bit in the Master Mode register. This is accomplished by writing an 8-bit command into the low-order DB lines while holding the DB13 – DB15 lines at a logic high level. Thereafter, all 16 lines can be used, with DB0 as the least significant and DB15 as the most significant bit position. When operating in the 8-bit data bus environment, DB8 – DB15 will never be driven active by the Am9513A. DB8 through DB12 may optionally be used as additional Gate inputs (see Figure 2). If unused, they should be held HIGH. When pulled LOW, a GATENA signal will disable the action of the corresponding counter N gating. DB13 – DB15 should be held HIGH in 8-bit bus mode whenever $\overline{CS}$ and $\overline{WR}$ are simultaneously active.
10	$\overline{CS}$	I	(Chip Select). The active-low Chip Select input enables Read and Write operations on the data bus. When Chip Select is HIGH, the Read and Write inputs are ignored. The first Chip Select signal after power-up is used to clear the power-on reset circuitry. If Chip Select is tied to ground permanently, the power-on reset circuitry may not function. In such a configuration, the software reset command must be issued following power-up to reset the Am9513A.
11	$\overline{RD}$	I	(Read). The active-low Read signal is conditioned by Chip Select and indicates that internal information is to be transferred to the data bus. The source will be determined by the port being addressed and, for Data Port reads, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
9	$\overline{WR}$	I	(Write). The active-low Write signal is conditioned by Chip Select and indicates that data bus information is to be transferred to an internal location. The destination will be determined by the port being addressed and, for Data Port writes, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
8	C/$\overline{D}$	I	(Control/Data). The Control/Data signal selects source and destination locations for Read and Write operations on the data bus. Control Write operations load the Command register and the Data Pointer. Control Read operations output the Status register. Data Read and Data Write transfers communicate with all other internal registers. Indirect addressing at the data port is controlled internally by the Data Pointer register.

Signal	Abbreviation	Type	Pins
+ 5 Volts	VCC	Power	1
Ground	VSS	Power	1
Crystal	X1, X2	O, I	2
Read	$\overline{RD}$	Input	1
Write	$\overline{WR}$	Input	1
Chip Select	$\overline{CS}$	Input	1
Control/Data	C/$\overline{D}$	Input	1
Source N	SRC	Input	5
Gate N	GATE	Input	5
Data Bus	DB	I/O	16
Frequency Out	FOUT	Output	1
Out N	OUT	Output	5

Figure 1. Interface Signal Summary

Figure 1 summarizes the interface signals and their abbreviations for the STC.

Package Pin	Data Bus Width (MM14)	
	16 Bits	8 Bits
12	DB0	DB0
13	DB1	DB1
14	DB2	DB2
15	DB3	DB3
16	DB4	DB4
17	DB5	DB5
18	DB6	DB6
19	DB7	DB7
20	DB8	GATE 1A
22	DB9	GATE 2A
23	DB10	GATE 3A
24	DB11	GATE 4A
25	DB12	GATE 5A
26	DB13	(VIH)
27	DB14	(VIH)
28	DB15	(VIH)

Figure 2. Data Bus Assignments

Interface Considerations

All of the input and output signals for the Am9513A are specified with logic levels compatible with those of standard TTL circuits. In addition to providing TTL compatible voltage levels, other output conditions are specified to help configure non-standard interface circuitry. The logic level specifications take into account all worst-case combinations of the three variables that affect the logic level thresholds: ambient temperature, supply voltage and processing parameters. A change in any of these toward nominal values will improve the actual operating margins and will increase noise immunity.

Unprotected open gate inputs of high quality MOS transistors exhibit very high resistances on the order of perhaps 10^{14} ohms. It is easy, therefore, in some circumstances, for charge to enter the gate node of such an input faster than it can be discharged and consequently, for the gate voltage to rise high enough to break down the oxides and destroy the transistor.

All inputs to the Am9513A include protection networks to help prevent damaging accumulations of static charge. The protection circuitry is designed to slow the transitions of incoming current surges and to provide low-impedance discharge paths for voltages beyond the normal operating levels. Note, however, that input energy levels can nonetheless be too high to be successfully absorbed. Conventional design, storage, and handling precautions should be observed so that the protection networks themselves are not overstressed.

Within the limits of normal operation, the input protection circuitry is inactive and may be modeled as a lumped series RC as shown in Figure 3(a). The functionally active input connection during normal operation is the gate of a MOS transistor. No active sources or drains are connected to the inputs so that neither transient nor steady-state currents are impressed on the driving signals other than the charging or discharging of the input capacitance and the accumulated leakage associated with the protection network and the input circuit.

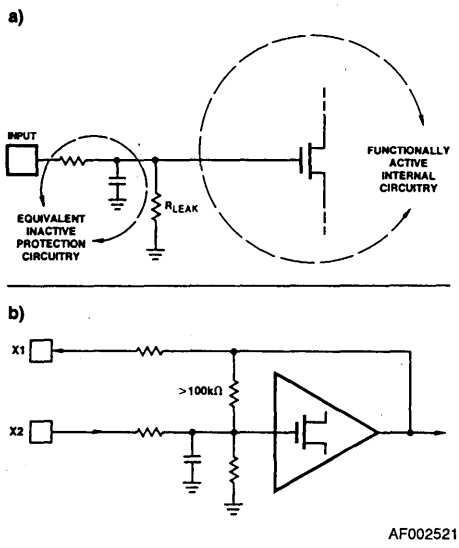


Figure 3. Input Circuitry

The only exception to the purely capacitive input case is the X2 crystal input. As shown in Figure 3(b) an internal resistor connects X1 and X2 in addition to the protection network. The resistor is a modestly high value of more than 100kohms.

Fanout from the driving circuitry into the Am9513A inputs will generally be limited by transition time considerations rather than DC current limitations when the loading is dominated by conventional MOS circuits. In an operating environment, all inputs should be terminated so they do not float and therefore will not accumulate stray static charges. Unused inputs should be tied directly to Ground or VCC, as appropriate. An input in use will have some type of logic output driving it, and termination during operation will not be a problem. Where inputs are driven from logic external to the card containing this chip, however, on-board termination should be provided to protect the chip when the board is unplugged (the input would otherwise float). A pull-up resistor or a simple inverter or gate will suffice.

DETAILED DESCRIPTION

The Am9513A System Timing Controller (STC) is a support device for processor oriented systems that is designed to enhance the available capability with respect to counting and timing operations. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital timing functions, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis and many more. A variety of programmable operating modes and control features allow the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide either pulses or levels. The counters can be programmed to count up or down in either binary or BCD. The accumulated count may be read without disturbing the counting process. Any of the counters may be internally concatenated to form an effective counter length of up to 80 bits.

The Am9513A block diagrams indicate the interface signals and the basic flow of information. Internal control lines and the internal data bus have been omitted. The control and data registers are all connected to a common internal 16-bit bus. The external bus may be 8- or 16-bits wide; in the 8-bit mode, the internal 16-bit information is multiplexed to the low order data bus pins DB0 through DB7.

An internal oscillator provides a convenient source of frequencies for use as counter inputs. The oscillator's frequency is controlled at the X1 and X2 interface pins by an external reactive network such as a crystal. The oscillator output is divided by the Frequency Scaler to provide several sub-frequencies. One of the scaled frequencies (or one of ten input signals) may be selected as an input to the FOUT divider and then comes out of the chip at the FOUT interface pin.

The STC is addressed by the external system as two locations: a Control port and a Data port. The Control port provides direct access to the Status and Command registers, as well as allowing the user to update the Data Pointer register. The Data port is used to communicate with all other addressable internal locations. The Data Pointer register controls the Data port addressing.

Among the registers accessible through the Data port are the Master Mode register and five Counter Mode registers, one for each counter. The Master Mode register controls the programmable options that are not controlled by the Counter Mode registers.

Each of the five general-purpose counters is 16-bits long and is independently controlled by its Counter Mode register. Through this register, a user can software select one of 16 sources as the counter input, a variety of gating and repetition modes, up or down counting in binary or BCD and active-high or active-low input and output polarities.

Associated with each counter are a Load register and a Hold register, both accessible through the Data port. The Load register is used to automatically reload the counter to any predefined value, thus controlling the effective count period. The Hold register is used to save count values without disturbing the count process, permitting the host processor to read intermediate counts. In addition, the Hold register may be used as a second Load register to generate a number of complex output waveforms.

All five counters have the same basic control logic and control registers. Counters 1 and 2 have additional Alarm registers and comparators associated with them, plus the extra logic necessary for operating in a 24-hour time-of-day mode. For real-time operation, the time-of-day logic will accept 50Hz, 60Hz or 100Hz input frequencies.

Each general counter has a single dedicated output pin. It may be turned off when the output is not of interest or may be configured in a variety of ways to drive interrupt controllers, Darlington buffers, bus drivers, etc. The counter inputs, on the other hand, are specifically not dedicated to any given interface line. Considerable versatility is available for configuring both the input and the gating of individual counters. This not only permits dynamic reassignment of inputs under software control, but also allows multiple counters to use a single input and a single gate pin to control more than one counter. Indeed, a single pin can be the gate for one counter and, at the same time, the count source for another.

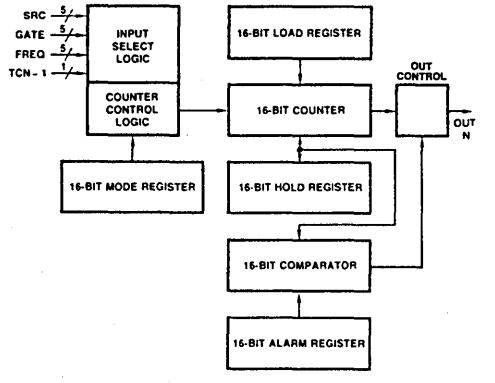


Figure 4. Counter Logic Groups 1 and 2

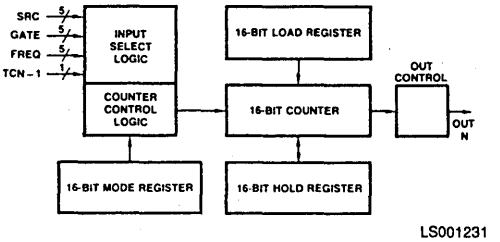


Figure 5. Counter Logic Groups 3, 4 and 5

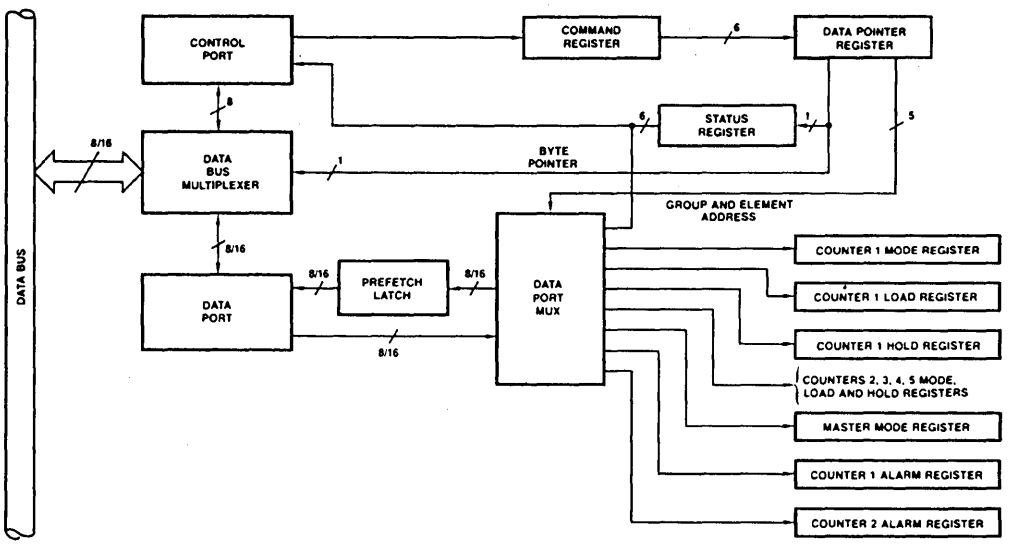


Figure 6. Am9513A Register Access

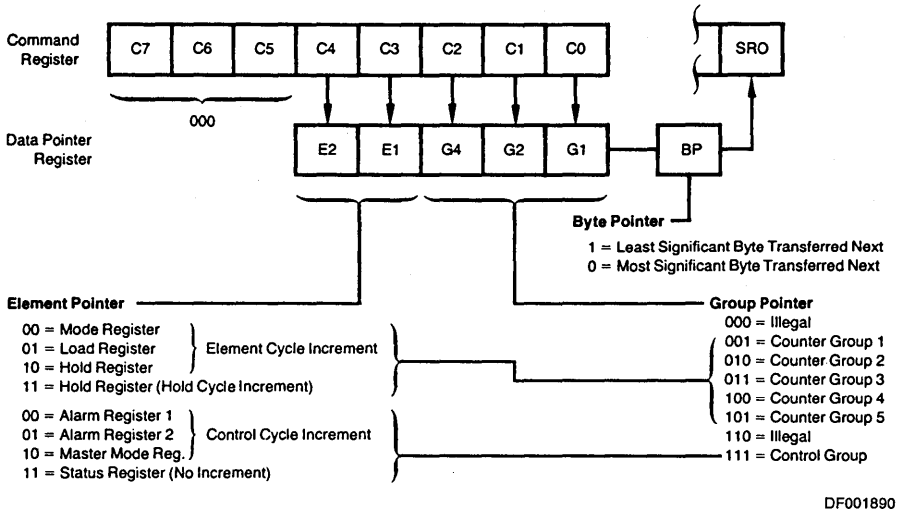


Figure 7. Data Pointer Register

	Element Cycle			Hold Cycle
	Mode Register	Load Register	Hold Register	Hold Register
Counter 1	FF01	FF09	FF11	FF19
Counter 2	FF02	FF0A	FF12	FF1A
Counter 3	FF03	FF0B	FF13	FF1B
Counter 4	FF04	FF0C	FF14	FF1C
Counter 5	FF05	FF0D	FF15	FF1D
Master Mode Register = FF17 Alarm 1 Register = FF07 Alarm 2 Register = FF0F Status Register = FF1F				

Notes:
1. All codes are in hex.
2. When used with an 8-bit bus, only the two low order hex digits should be written to the command port; the "FF" prefix should be used only for a 16-bit data bus interface.

Figure 8. Load Data Pointer Commands

Sequencing is enabled by clearing Master Mode bit 14 (MM14) to zero. As shown in Figure 9 several types of sequencing are available depending on the data bus width being used and the initial Data Pointer value entered by command.

When E1 = 0 or E2 = 0 and G4, G2, G1 points to a Counter Group, the Data Pointer will proceed through the Element cycle. The Element field will automatically sequence through the three values 00, 01 and 10 starting with the value entered. When the transition from 10 to 00 occurs, the Group field will also be incremented by one. Note that the Element field in this case does not sequence to a value of 11. The Group field circulates only within the five Counter Group codes.

If E2, E1 = 11 and a Counter Group are selected, then only the Group field is sequenced. This is the Hold cycle. It allows the Hold registers to be sequentially accessed while bypassing the Mode and Load registers. The third type of sequencing is the Control cycle. If G4, G2, G1 = 111 and E2, E1 ≠ 11, the Element Pointer will be incremented through the values 00, 01 and 10, with no change to the Group Pointer.

When G4, G2, G1 = 111 and E2, E1 = 11, no incrementing takes place and only the Status register will be available through the Data port. Note that the Status register can also always be read directly through the Control port.

For all these auto-sequencing modes, if an 8-bit data bus is used, the Byte pointer will toggle after every data transfer to allow the least and most significant bytes to be transferred before the Element or Group fields are incremented.

Prefetch Circuit

To minimize the read access time to internal Am9513A registers, a prefetch circuit is used for all read operations through the Data port. Following each read or write operation through the Data port, the Data Pointer register is updated to point to the next register to be accessed. Immediately following this update, the new register data is transferred to a special prefetch latch at the interface pad logic. When the user performs a subsequent read of the Data port, the data bus drivers are enabled, outputting the prefetched data on the bus. Since the internal data register is accessed prior to the start of the read operation, its access time is transparent to the user. To keep the prefetched data consistent with the Data Pointer, prefetches are also performed after each write to the Data port and after execution of the "Load Data Pointer" command. The following rules should be kept in mind regarding Data port Transfers.

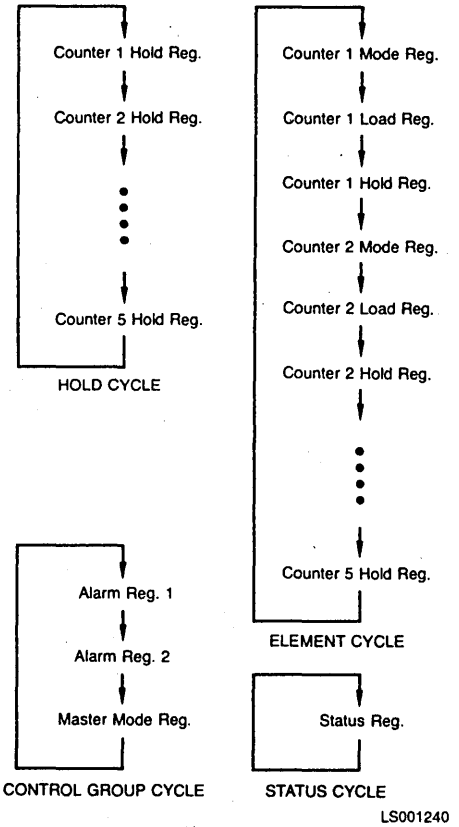


Figure 9. Data Pointer Sequencing

1. The Data Pointer register should always be reloaded before reading from the Data port if a command, other than "Load Data Pointer," was issued to the Am9513A following the last Data port read or write. The Data Pointer does not have to be loaded again if the first Data port transaction after a command entry is a write, since the Data port write will automatically cause a new prefetch to occur.
2. Operating modes N, O, Q, R and X allow the user to save the counter contents in the Hold register by applying an active-going gate edge. If the Data Pointer register had been pointing to the Hold register in question, the prefetched value will not correspond to the new value saved in the Hold Register. To avoid reading an incorrect value, a new "Load Data Pointer" command should be issued before attempting to read the saved data. A Data port write (to another register) will also initiate a prefetch; subsequent reads will access the recently saved Hold register data. Many systems will use the "saving" gate edge to interrupt the host CPU. In systems such as this, the interrupt service routine should issue a "Load Data Pointer" command prior to reading the saved data.

Status Register

The 8-bit read-only Status register indicates the state of the Byte Pointer bit in the Data Pointer register and the state of the

OUT signal for each of the general counters. See Figures 10 and 17. The OUT signals reported are those internal to the chip after the polarity-select logic and just before the three-state interface buffer circuitry. Bits SR6 and SR7 may be 0 or 1.

The Status register OUT bit reflects an active-high or active-low TC output or a TC Toggled output, as programmed in the Output Control Field of the Counter Mode register. That is, it reflects the exact state of the OUT pin. When the low-impedance to Ground Output option (CM2-CM0 = 000) is selected, the Status register will reflect an active-high TC Output. When a high-impedance Output option (CM2-CM0 = 100) is selected, the Status register will reflect an active-low TC output.

For Counters 1 and 2, the OUT pin will reflect the comparator output if the comparators are enabled. The Status register bit and OUT pin are active-high if CM2 = 0 and active-low if CM2 = 1. When the high-impedance option is selected and the comparator is enabled, the status register bit will reflect an active-high comparator output. When the low-impedance to Ground option is selected and the comparator is enabled, the status register bit will reflect an active-low comparator output.

The Status register is normally accessed by reading the Control port (see Figure 6) but may also be read via the Data port as part of the Control Group.

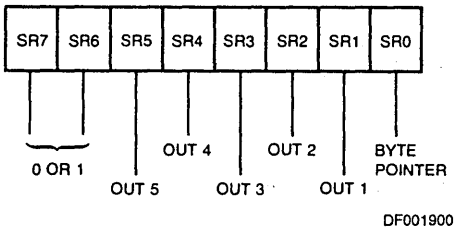


Figure 10. Status Register Bit Assignments

DATA PORT REGISTERS

Counter Logic Groups

As shown in Figures 4 and 5, each of the five Counter Logic Groups consists of a 16-bit general counter with associated control and output logic, a 16-bit Load register, a 16-bit Hold register and a 16-bit Mode register. In addition, Counter Groups 1 and 2 also include 16-bit Comparators and 16-bit Alarm registers. The comparator/alarm functions are controlled by the Master Mode register. The operation of the Counter Mode registers is the same for all five counters. The host CPU has both read and write access to all registers in the Counter Logic Groups through the Data port. The counter itself is never directly accessed.

Load Register

The 16-bit read/write Load register is used to control the effective length of the general counter. Any 16-bit value may be written into the Load register. That value can then be transferred into the counter each time the Terminal Count (TC) occurs. "Terminal Count" is defined as that period of time when the counter contents would have been zero if an external value had not been transferred into the counter. Thus, the terminal count frequency can be the input frequency divided by the value in the Load register. In all operating modes, either the Load or Hold register will be transferred into the counter when TC occurs. In cases where values are being accumulated in the counter, the Load register action can become transparent by filling the Load register with all zeros.

Hold Register

The 16-bit read/write Hold register is dual-purpose. It can be used in the same way as the Load register, thus offering an alternate source for module definition for the counter. The Hold register may also be used to store accumulated counter values for later transfer to the host processor. This allows the count to be sampled while the counting process proceeds without interruption. Transfer of the counter contents into the Hold register is accomplished by the hardware interface in some operating modes or by software commands at any time.

Counter Mode Register

The 16-bit read/write Counter Mode register controls the gating, counting, output and source select functions within each Counter Logic Group. The "Counter Mode Control Options" section of this document describes the detailed control options available. Figure 16 shows the bit assignments for the Counter Mode registers.

Alarm Registers and Comparators

Added functions are available in the Counter Logic Groups for Counters 1 and 2 (see Figure 4). Each contains a 16-bit Alarm register and a 16-bit Comparator. When the value in the counter reaches the value in the Alarm register, the Comparator output will go true. The Master Mode register contains control bits to individually enable/disable the comparators. When enabled, the comparator output appears on the OUT pin of the associated counter in place of the normal counter output. The output will remain true as long as the comparison is true, that is, until the next input causes the count to change. The polarity of the Comparator output will be active-high if the Output Control field of the Counter Mode register is 001 or 010 and active-low if the Output Control field is 101.

MASTER MODE CONTROL OPTIONS

The 16-bit Master Mode (MM) register is used to control those internal activities that are not controlled by the individual Counter Mode registers. This includes frequency control, Time-of-Day operation, comparator controls, data bus width and data pointer sequencing. Figure 11 shows the bit assignments for the Master Mode register. This section describes the use of each control field.

Master Mode register bits MM12, MM13 and MM14 can be individually set and reset using commands issued to the Command register. In addition, they can all be changed by writing directly to the Master Mode register.

After power-on reset or a Master Reset command, the Master Mode register is cleared to an all zero condition. This results in the following configuration:

Time-of-Day disabled
Both Comparators disabled
FOUT Source is frequency F1
FOUT Divider set for divide-by-16
FOUT gated on
Data Bus 8 bits wide
Data Pointer Sequencing enabled
Frequency Scaler divides in binary

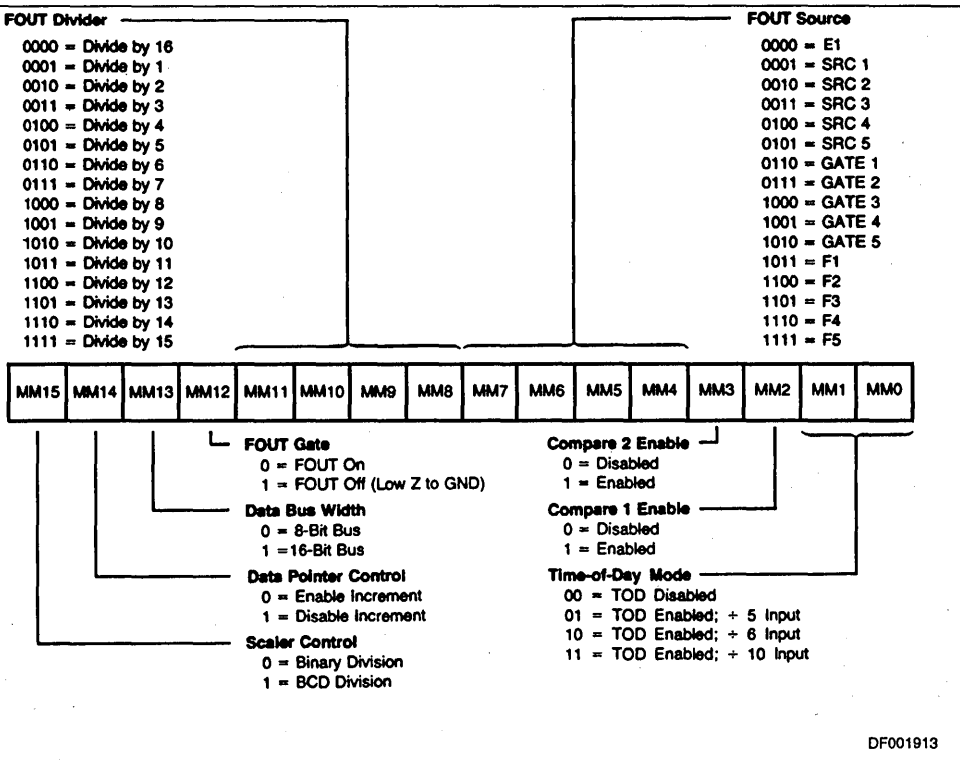


Figure 11. Master Mode Register Bit Assignments

Time-of-Day

Bits MM0 and MM1 of the Master Mode register specify the Time-of-Day (TOD) options. When MM0 = 0 and MM1 = 0, the special logic used to implement TOD is disabled, and Counters 1 and 2 will operate in exactly the same way as Counters 3, 4 and 5. When MM0 = 1 or MM1 = 1, additional counter decoding and control logic is enabled on Counters 1 and 2, which causes their decades to turn over at the counts that generate appropriate 24-hour TOD accumulations. For additional information, see the Time-of-Day chapter in the 9513A System timing controller technical manual.

Comparator Enable

Bits MM2 and MM3 control the Comparators associated with Counters 1 and 2. When a Comparator is enabled, its output is substituted for the normal counter output on the associated OUT1 or OUT2 pin. The comparator output will be active-high if the output control field of the Counter Mode register is 001 or 010 and active-low for a code of 101. Once the compare output is true, it will remain so until the count changes and the comparison therefore goes false.

The two Comparators can always be used individually in any operating mode. One special case occurs when the Time-of-Day option is revoked and both Comparators are enabled. The operation of Comparator 2 will then be conditioned by Comparator 1 so that a full 32-bit compare must be true in order to generate a true signal on OUT2. OUT1 will continue, as usual, to reflect the state of the 16-bit comparison between Alarm 1 and Counter 1.

FOUT Source

Master Mode bits MM4 through MM7 specify the source input for the FOUT divider. Fifteen inputs are available for selection, and they include the five Source pins, the five Gate pins and the five internal frequencies derived from the oscillator. The 16th combination of the four control bits (all zeros) is used to assure that an active frequency is available at the input to the FOUT divider following reset.

FOUT Divider

Bits MM8 through MM11 specify the dividing ratio for the FOUT Divider. The FOUT source (selected by bits MM4 through MM7) is divided by an integer value between 1 and 16, inclusive, and is then passed to the FOUT output buffer. After power-on or reset, the FOUT divider is set to divide-by-16.

FOUT Gate

Master Mode bit MM12 provides a software gating capability for the FOUT signal. When MM12 = 1, FOUT is off and in a low-impedance state to ground. MM12 may be set or cleared in conjunction with the loading of the other bits in the Master Mode register; alternatively, there are commands that allow MM12 to be individually set or cleared directly without changing any other Master Mode bits. After power-up or reset, FOUT is gated on.

When changing the FOUT divider ratio or FOUT source, transient pulses as short as half the period of the FOUT source may appear on the FOUT pin. Turning the FOUT gate on or off can also generate a transient. This should be considered when using FOUT as a system clock source.

Bus Width

Bit MM13 controls the multiplexer at the data bus interface in order to configure the part for an 8-bit or 16-bit external bus. The internal bus is always 16-bits wide. When MM13 = 1, 16-bit data is transferred directly between the internal bus and all 16 of the external bus lines. In this configuration, the Byte Pointer bit in the Data Pointer register remains set at all times. When MM13 = 0, 16-bit internal data is transferred a byte at a time to and from the eight low-order external data bus lines. The Byte Pointer bit toggles with each byte transfer in this mode.

When the Am9513A is set to operate with an 8-bit data bus width, pins DB8 through DB15 are not used for the data bus and are available for other functions. Pins DB13 through DB15 should be tied high. Pins DB8 through DB12 are used as auxiliary gating inputs and are labeled GATE1A through GATE5A respectively. The auxiliary gate pin, GATENA, is logically ANDed with the gate input to Counter N, as shown in Figure 12. The output of the AND gate is then used as the gating signal for Counter N.

Data Pointer Sequencing

Bit MM14 controls the Data Pointer logic to enable or disable the automatic sequencing functions. When MM14 = 1, the contents of the Data Pointer can be changed only directly by entering a command. When MM14 = 0, several types of automatic sequencing of the Data Pointer are available. These are described in the Data Pointer register section of this document.

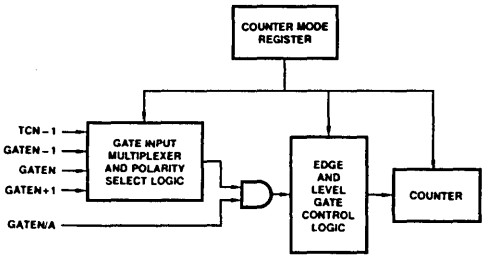
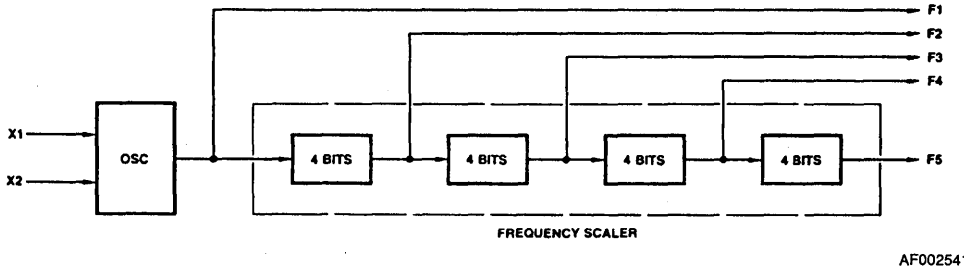


Figure 12. Gating Control

Thus, the host processor, by controlling MM14, may repetitively read/write a single internal location, or may sequentially read/write groups of locations. Bit MM14 can be loaded by writing to the Master Mode register or can be set or cleared by software command.

Scaler Ratios

Master Mode bit MM15 controls the counting configuration of the Frequency Scaler counter. When MM15 = 0, the Scaler divides the oscillator frequency in binary steps so that each subfrequency is 1/16 of the preceding frequency. When MM15 = 1, the Scaler divides in BCD steps so that adjacent frequencies are related by ratios of 10 instead of 16 (see Figure 13).



Frequency	BCD Scaling MM15 = 1	Binary Scaling MM15 = 0
F1	OSC	OSC
F2	F1 ÷ 10	F1 ÷ 16
F3	F1 ÷ 100	F1 ÷ 256
F4	F1 ÷ 1,000	F1 ÷ 4,096
F5	F1 ÷ 10,000	F1 ÷ 65,536

Figure 13. Frequency Scaler Ratios

Counter Mode	A	B	C	D	E	F	G	H	I	J	K	L
Special Gate (CM7)	0	0	0	0	0	0	0	0	0	0	0	0
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15–CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm	X	X	X									
Count to TC twice, then disarm							X	X	X			
Count to TC repeatedly without disarming				X	X	X				X	X	X
Gate input does not gate counter input	X			X			X			X		
Count only during active gate level		X			X			X			X	
Start count on active gate edge and stop count on next TC			X			X						
Start count on active gate edge and stop count on second TC									X			X
No hardware retriggering	X	X	X	X	X	X	X	X	X	X	X	X
Reload counter from Load register on TC	X	X	X	X	X	X						
Reload counter on each TC, alternating reload source between Load and Hold registers							X	X	X	X	X	X
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.												
On active gate edge transfer counter into Hold register and then reload counter from Load register												

Counter Mode	M	N	O	P	Q	R	S	T	U	V	W	X
Special Gate (CM7)	1	1	1	1	1	1	1	1	1	1	1	1
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15–CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm		X	X									
Count to TC twice, then disarm							X					
Count to TC repeatedly without disarming					X	X				X		X
Gate input does not gate counter input							X			X		
Count only during active gate level		X			X							
Start count on active gate edge and stop count on next TC			X			X						X
Start count on active gate edge and stop count on second TC												
No hardware retriggering							X			X		X
Reload counter from Load register on TC		X	X		X	X						X
Reload counter on each TC, alternating reload source between Load and Hold registers.												
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.							X			X		
On active gate edge transfer counter into Hold register and then reload counter from Load register		X	X		X	X						
On active gate edge transfer counter into Hold register, but counting continues												X

Notes: 1. Counter modes M, P, T, U and W are reserved and should not be used.
2. Mode X is available for Am9513A only.

Figure 14. Counter Mode Operating Summary

COUNTER MODE DESCRIPTIONS

Counter Mode register bits CM15–CM13 and CM7–CM5 select the operating mode for each counter (see Figure 14). To simplify references to a particular mode, each mode is assigned a letter from A through X. Representative waveforms for the counter modes are illustrated in Figures 15a through 15v. (Because the letter suffix in the figure number is keyed to the mode, Figures 15m, 15p, 15t, 15u and 15w do not exist.) The figures assume down counting on rising source edges. Those modes which automatically disarm the counter (CM5 = 0) are shown with the $\overline{WR}$ plus entering the required ARM command; for modes which count repetitively (CM5 = 1), the ARM command is omitted. The retriggering modes (N, O, Q and R) are shown with one retrigger operation. Both a TC output waveform and a TC Toggled output waveform are shown for each mode. The symbols L and H are used to represent count values equal to the Load and Hold register contents, respectively. The symbols K and N represent arbitrary count values. For each mode, the required bit pattern in the Counter Mode register is shown; "don't care" bits are marked "X." These figures are designed to clarify the mode descriptions; the Am9513A Electrical Specification should be used as the authoritative reference for timing relationships between signals.

To keep the following mode descriptions concise and to the point, the phrase "source edges" is used to refer to active-going source edges only, not to inactive-going edges. Similarly, the phrase "gate edges" refers only to active-going gate edges. Also, again to avoid verbosity and euphuism, the descriptions of some modes state that a counter is stopped or disarmed "on a TC, inhibiting further counting." As is fully explained in the TC section of this document, for these modes the counter is actually stopped or disarmed following the active-going source edge which drives the counter out of TC. In other words, since a counter in the TC state always counts, irrespective of its gating or arming status, the stopping or disarming of the count sequence is delayed until TC is terminated.

MODE A

Software-Triggered Strobe with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode A, shown in Figure 15a, is one of the simplest operating modes. The counter will be available for counting source edges when it is issued an ARM command. On each TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume when a new ARM command is issued.

MODE B

Software-Triggered Strobe with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode B, shown in Figure 15b, is identical to Mode A except that source edges are counted only when the assigned Gate is active. The counter must be armed before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. On each TC the counter will reload from the Load register and automatically disarm itself, inhibiting further counting unitl a new ARM command is issued.

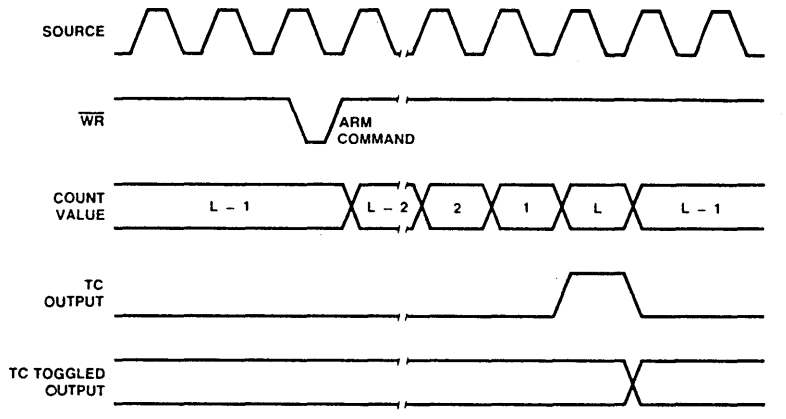


Figure 15a. Mode A Waveforms

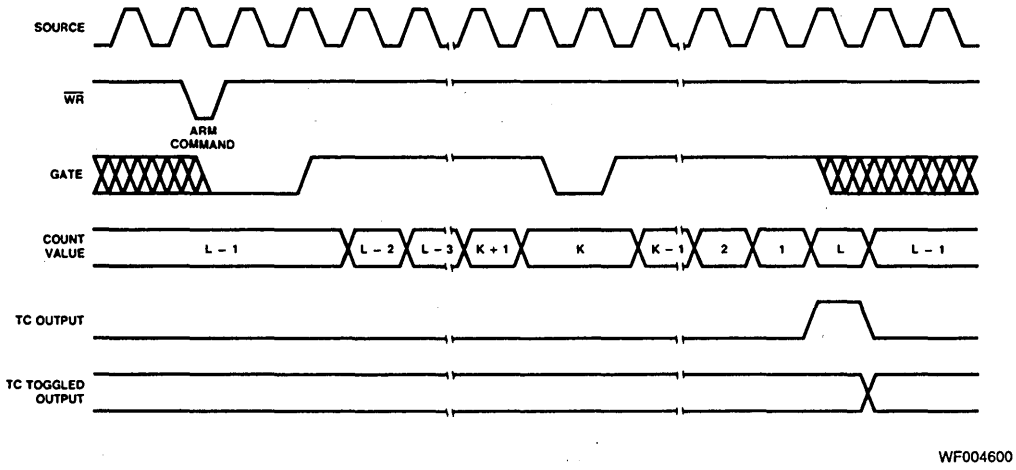


Figure 15b. Mode B Waveforms

MODE C

Hardware-Triggered Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode C, shown in Figure 15c, is identical to Mode A, except that counting will not begin until a Gate edge is applied to the armed counter. The counter must be armed before application of the triggered Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting on the first source edge after the triggering Gate edge and will continue counting until TC. At TC, the counter will reload from the Load register and automatically disarm itself. Counting will then remain inhibited until a new ARM command and a new Gate edge are applied in that order. Note that after application of a triggered Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode B, where the Gate can be modulated throughout the count cycle to stop and start the counter.

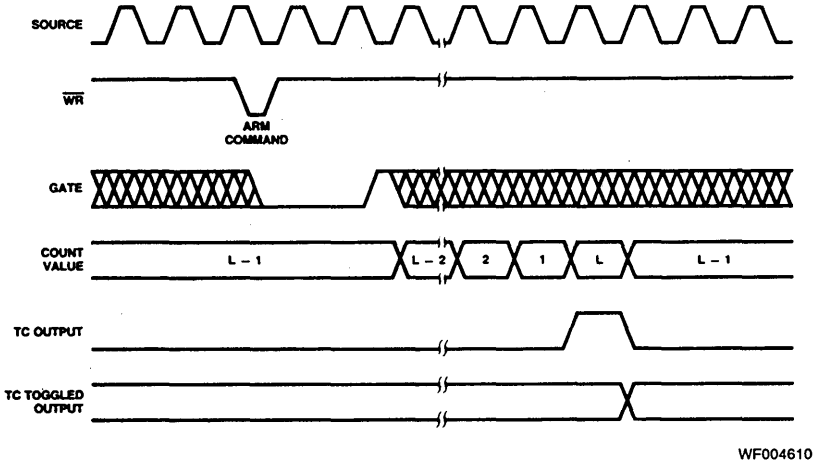


Figure 15c. Mode C Waveforms

MODE D

Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode D, shown in Figure 15d, is typically used in frequency generation applications. In this mode, the Gate input does not affect counter operation. Once armed, the counter will count to TC repetitively. On each TC, the counter will reload itself from the Load register; hence, the Load register value determines the time between TCs. A square wave rate generator may be obtained by specifying the TC Toggled output mode in the Counter Mode register.

MODE E

Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode E, shown in Figure 15e, is identical to Mode D, except the counter will only count those source edges which occur while the Gate input is active. This feature allows the counting process to be enabled and disabled under hardware control. A square wave rate generator may be obtained by specifying the TC Toggled output mode.

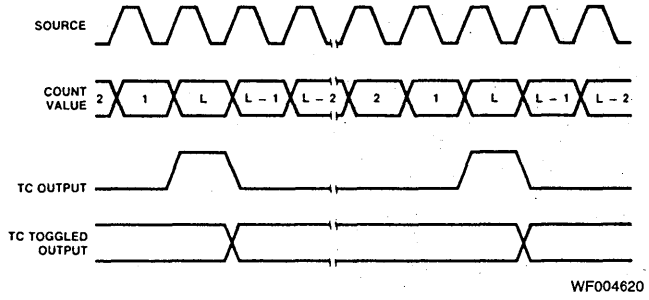


Figure 15d. Mode D Waveforms

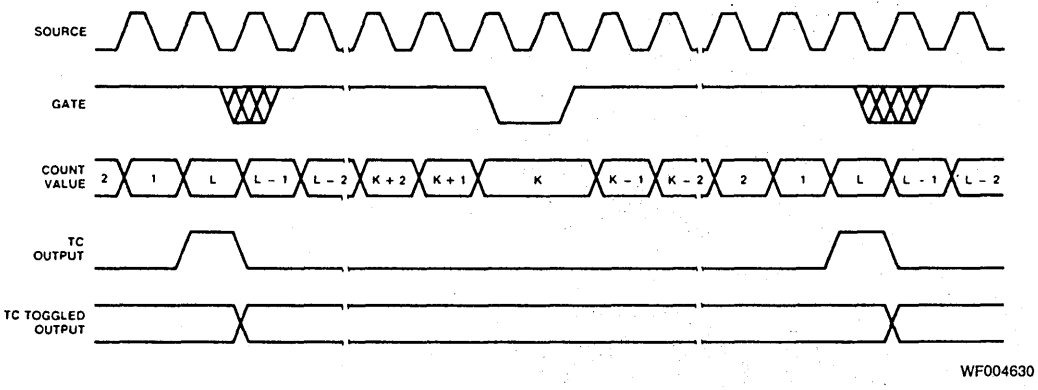


Figure 15e. Mode E Waveforms

MODE F

Non-Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode F, shown in Figure 15f, provides a non-retriggerable one-shot timing function. The counter must be armed before it will function. Application of a Gate edge to the armed counter will enable counting. When the counter reaches TC, it will reload itself from the Load register. The counter will then stop counting, awaiting a new Gate edge. Note that unlike Mode C, a new ARM command is not needed after TC, only a new Gate edge. After application of a triggering Gate edge, the Gate input is disregarded until TC.

MODE G

Software-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

In Mode G, the Gate does not affect the counter's operation. Once armed, the counter will count to TC twice and then automatically disarm itself. For most applications, the counter will initially be loaded from the Load register either by a LOAD command or by the last TC of an earlier timing cycle. Upon counting to the first TC, the counter will reload itself from the Hold register. Counting will proceed until the second TC, when the counter will reload itself from the Load register and automatically disarm itself, inhibiting further counting. Counting can be resumed by issuing a new ARM command. A software-triggered delayed pulse one-shot may be generated by specifying the TC Toggled output mode in the Counter Mode register. The initial counter contents control the delay from the ARM command until the output pulse starts. The Hold register contents control the pulse duration. Mode G is shown in Figure 15g.

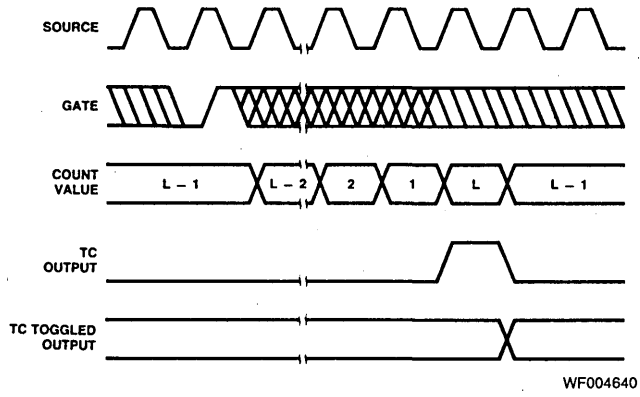
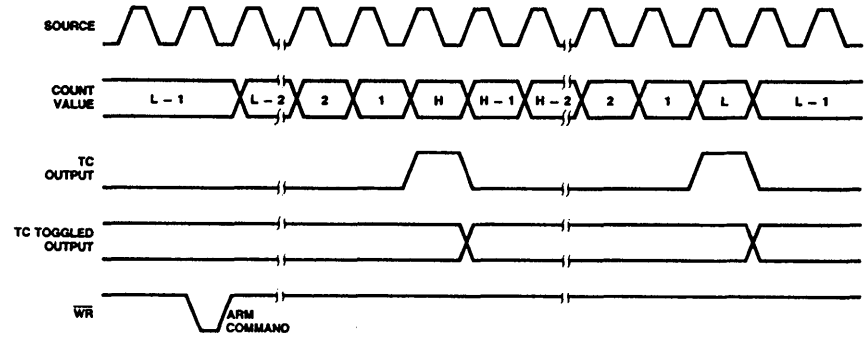


Figure 15f. Mode F Waveforms



WF004650

Figure 15g. Mode G Waveforms

MODE H

Software-Triggered Delayed Pulse One-Shot with Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode H, shown in Figure 15h, is identical to Mode G except that the Gate input is used to qualify which source edges are to be counted. The counter must be armed for counting to occur. Once armed, the counter will count all source edges that occur while the Gate is active. This permits the Gate to turn the count process on and off. As with Mode G, the counter will be reloaded from the Hold register on the first TC and reloaded from the Load register and disarmed on the second TC. This mode allows the Gate to control the extension of both the initial output delay time and the pulse width.

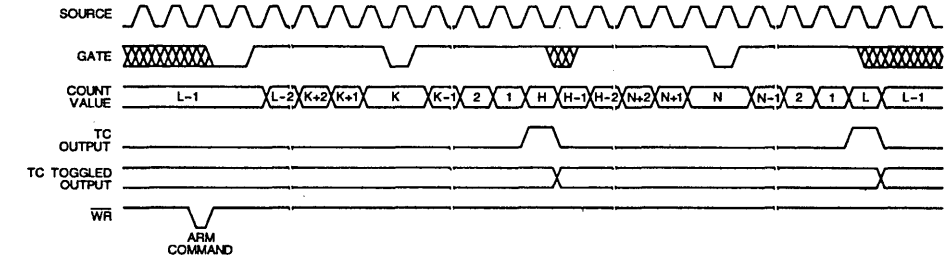
MODE I

Hardware-Triggered Delayed Pulse Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode I, shown in Figure 15i, is identical to Mode G, except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. An armed counter will start counting on the first source edge after the triggering Gate edge. Counting will then proceed in the same manner as in Mode G. After the second TC, the counter will disarm itself. An ARM command and Gate edge must be issued in this order to restart counting. Note that after application of a triggering Gate edge, the Gate input will be disregarded until the second TC. This differs from Mode H, where the Gate can be modulated throughout the count cycle to stop and start the counter.



WF004661

Figure 15h. Mode H Waveforms

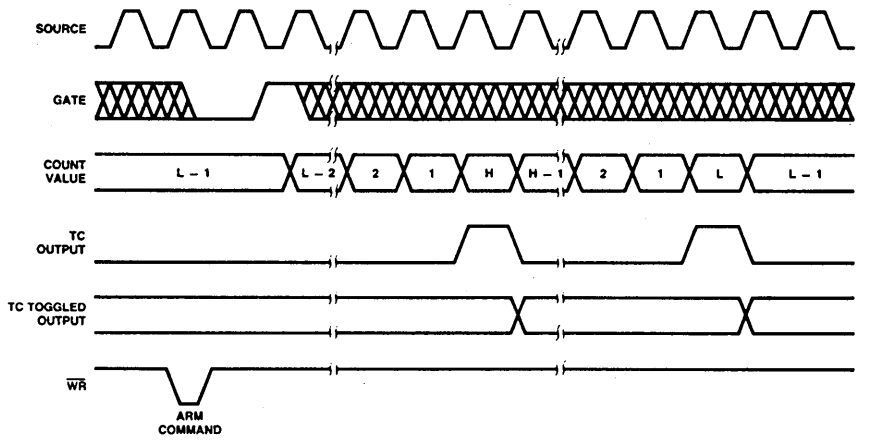


Figure 15i. Mode I Waveforms

MODE J

Variable Duty Cycle Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode J, shown in Figure 15j, will find the greatest usage in frequency generation applications with variable duty cycle requirements. Once armed, the counter will count continuously until it is issued a DISARM command. On the first TC, the counter will be reloaded from the Hold register. Counting will then proceed until the second TC at which time the counter will be reloaded from the Load register. Counting will continue, with the reload source alternating on each TC, until a DISARM command is issued to the counter. (The third TC reloads from the Hold register, the fourth TC reloads from the Load register, etc.) A variable duty cycle output can be generated by specifying the TC Toggled output in the Counter Mode register. The Load and Hold values then directly control the output duty cycle, with high resolution available when relatively high count values are used.

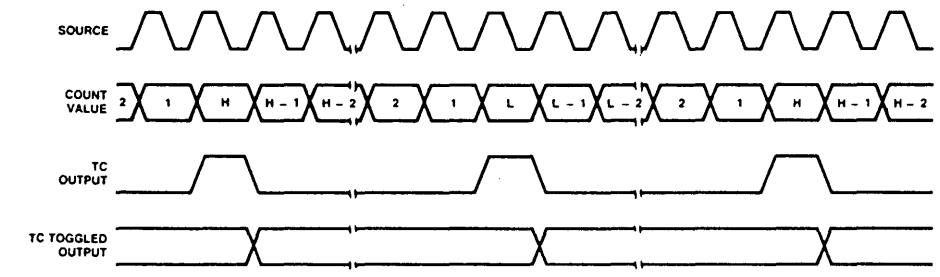
MODE K

Variable Duty Cycle Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

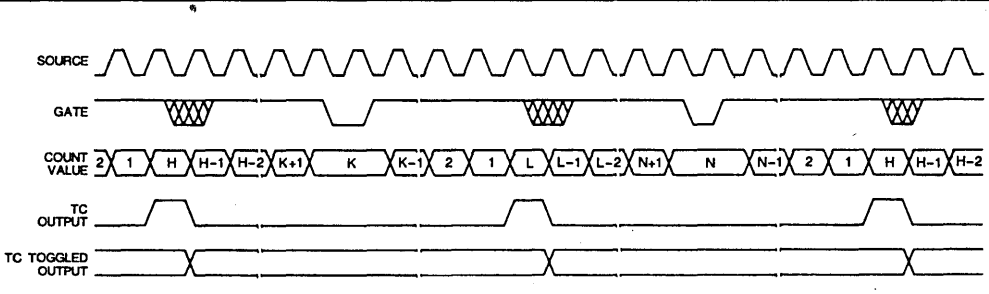
CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode K, shown in Figure 15k, is identical to Mode J, except that source edges are only counted when the Gate is active. The counter must be armed for counting to occur. Once armed, the counter will count all source edges which occur while Gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode J, the reload source used will alternate on each TC, starting with the Hold register on the first TC after any ARM command. When the TC Toggled output is used, this mode allows the Gate to modulate the duty cycle of the output waveform. It can affect both the HIGH and LOW portions of the output waveform.



WF004680

Figure 15j. Mode J Waveforms



WF004691

Figure 15k. Mode K Waveforms

MODE L

Hardware-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode L, shown in Figure 15l, is similar to Mode J except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting source edges after the triggering Gate edge, and counting will proceed until the second TC. Note that after application of a triggering Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode K, where the gate can be modulated throughout the count cycle to stop and start the counter. On the first TC after application of the triggering Gate edge, the counter will be reloaded from the Hold register. On the second TC, the counter will be reloaded from the Load register, and counting will stop until a new gate edge is issued to the counter. Note that unlike Mode K, new Gate edges are required after every second TC to continue counting.

MODE N

Software-Triggered Strobe with Level Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode N, shown in Figure 15n, provides a software-triggered strobe with level gating that is also hardware retriggerable. The counter must be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of the ARM command and the application of an active Gate, the counter will count to TC. Upon reaching TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume upon the issuance of a new ARM command. All active-going Gate edges issued to an armed counter will cause a retrigger operation. Upon application of the Gate edge, the counter contents will be saved in the Hold register. On the first qualified source edge after application of the retriggering gate edge, the contents of the Load register will be transferred into the counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

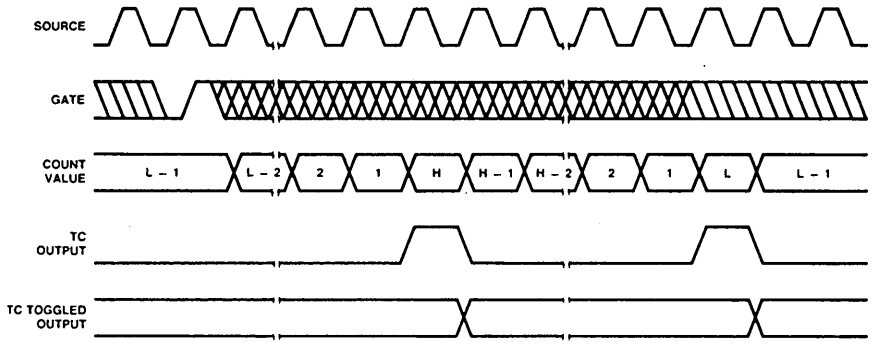


Figure 15l. Mode L Waveforms

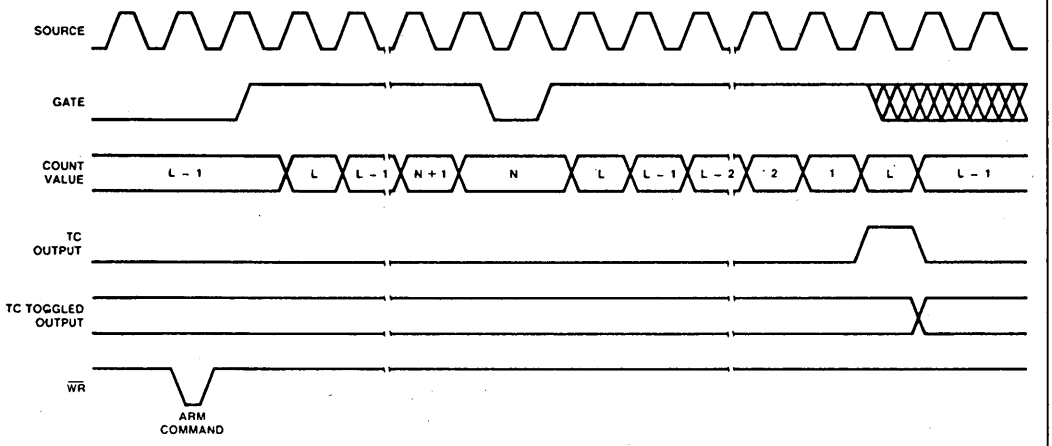


Figure 15n. Mode N Waveforms

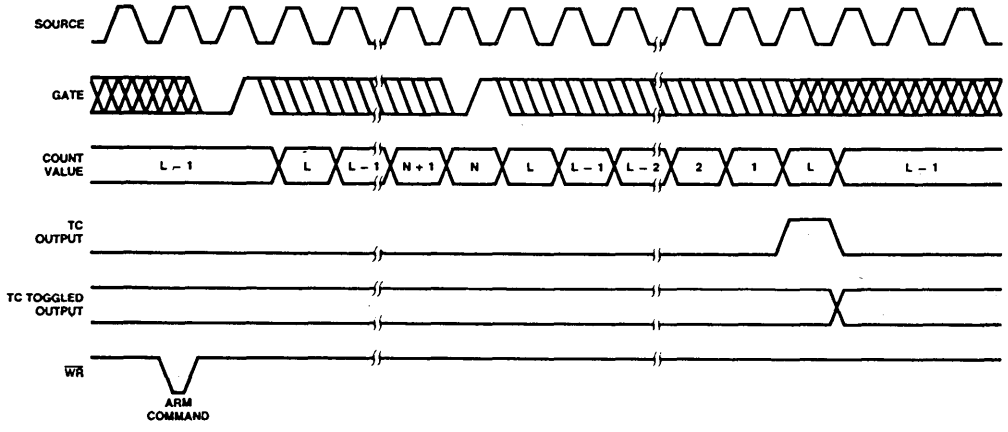
MODE O

Software-Triggered Strobe with Edge Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode O, shown in Figure 15o, is similar Mode N, except that counting will not begin until an active-going Gate edge is applied to an armed counter and the Gate level is not used to modulate counting. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. Irrespective of the Gate level, the counter will count all source edges after the triggering Gate edge until the first TC. On the first TC,the counter will be reloaded from the Load register and disarmed. A new ARM command and a new Gate edge must be applied in that order to initiate a new counting cycle. Unlike Modes C, F, I and L, which disregard the Gate input once counting starts, in Mode O the count process will be retriggered on all active-going Gate edges, including the first Gate edge used to start the counter. On each retriggering Gate edge, the counter contents will be transferred into the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second-source edge after a retrigger.



WF004720

Figure 15o. Mode O Waveforms

MODE Q

Rate Generator with Synchronization (Event Counter with Auto-Read/Reset)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode Q, shown in Figure 15q, provides a rate generator with synchronization or an event counter with auto-read/reset. The counter must first be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which cccur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of an ARM command and the application of an active Gate, the counter will count to TC repetitively. On each TC,the counter will reload itself from the Load register. The counter may be retriggered at any time by presenting an active-going Gate edge to the Gate input. The retriggering Gate edge will transfer the contents of the counter into the Hold register. The first qualified source edge after the retriggering Gate edge will transfer the contents of the Load register into the Counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

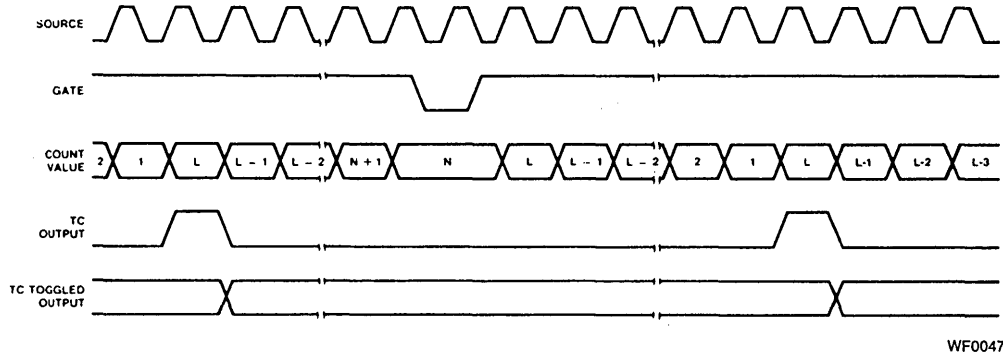
MODE R

Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

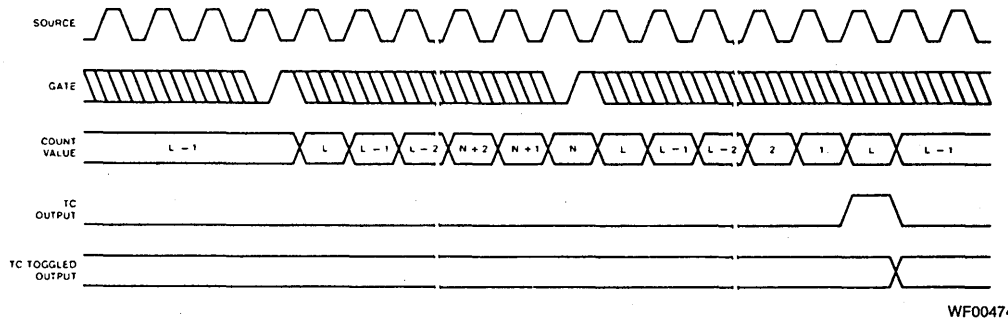
CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode R, shown in Figure 15r, is similar to Mode Q, except that edge gating rather than level gating is used. In other words, rather than use the Gate level to qualify which source edges to count, Gate edges are used to start the counting operation. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. After application of a Gate edge, an armed counter will count all source edges until TC, irrespective of the Gate level. On the first TC, the counter will be reloaded from the Load register and stopped. Subsequent counting will not occur until a new Gate edge is applied. All Gate edges applied to the counter, including the first used to trigger counting, initiate a retrigger operation. Upon application of a Gate edge, the counter contents are saved in the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second source edge after the retriggering Gate edge.



WF004730

Figure 15q. Mode Q Waveforms



WF004740

Figure 15r. Mode R Waveforms

MODE S

RELOAD SOURCE

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	0	X	X	X	X	X

In this mode, the reload source for LOAD commands (irrespective of whether the counter is armed or disarmed) and for TC-initiated reloads is determined by the Gate input. The Gate input in Mode S is used only to select the reload source, not to start or modulate counting. When the Gate is Low, the Load register is used; when the Gate is High, the Hold register is used. Note the Low-Load, High-Hold mnemonic convention. Once armed, the counter will count to TC twice and then disarm itself. On each TC, the counter will be reloaded from the reload source selected by the Gate. Following the second TC, an ARM command is required to start a new counting cycle. Mode S is shown in Figure 15s.

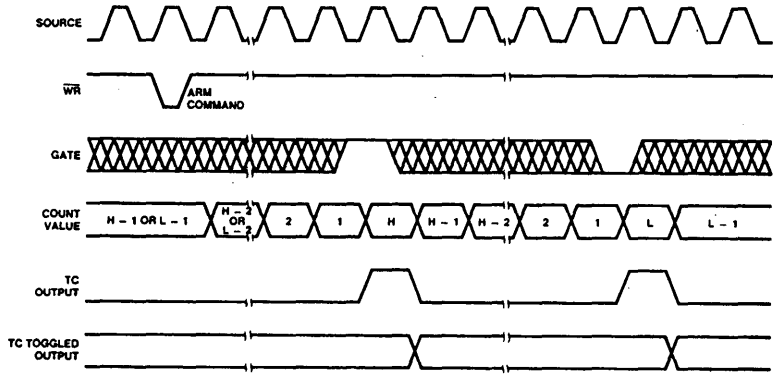
MODE V

Frequency-Shift Keying

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

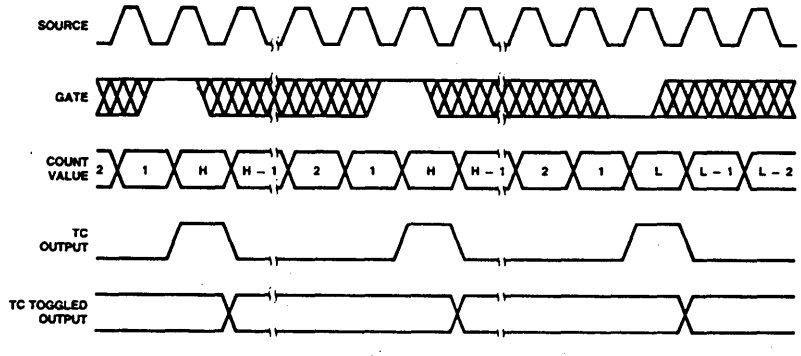
CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode V, shown in Figure 15v, provides frequency-shift keying modulation capability. Gate operation in this mode is identical to that in Mode S. If the Gate is Low, a LOAD command or a TC-induced reload will reload the counter from the Load register. If the Gate is HIGH, LOADs and reloads will occur from the Hold register. The polarity of the Gate only selects the reload source; it does not start or modulate counting. Once armed, the counter will count repetitively to TC. On each TC, the counter will reload itself from the register determined by the polarity of the Gate. Counting will continue in this manner until a DISARM command is issued to the counter. Frequency shift keying may be obtained by specifying a TC Toggled output mode in the Counter Mode register. The switching of frequencies is achieved by modulating the Gate.



WF004751

Figure 15s. Mode S Waveforms



WF004760

Figure 15v. Mode V Waveforms

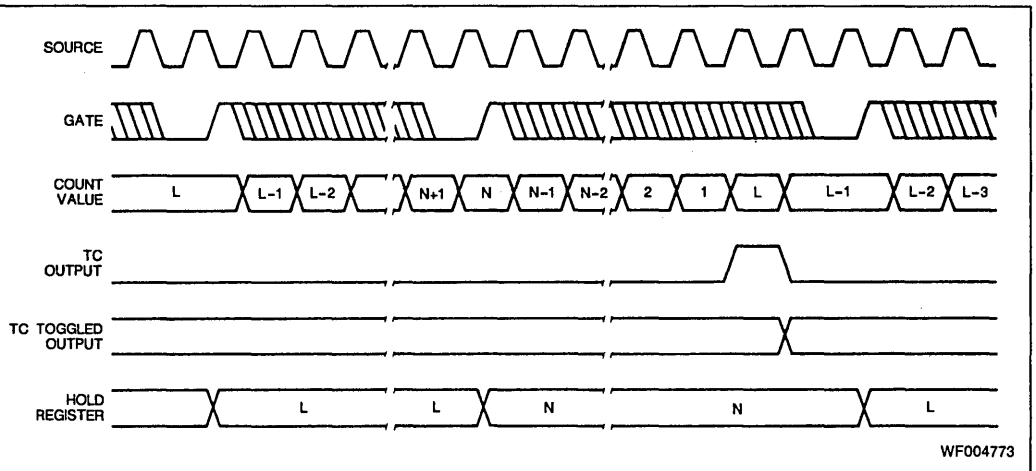


Figure 15x. Mode X Waveforms

MODE X

Hardware Save (available in Am9513A only)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
Edge			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode X, as shown in Figure 15x, provides a hardware sampling of the counter contents without interrupting the count. A LOAD AND ARM command or a LOAD command followed by an ARM command is required to initialize the counter. Once armed, a Gate edge starts the counting operation; Gate edges applied to a disarmed counter are disregarded. After application of the Triggering Gate edge, the counter will count all qualified source edges until the first TC, irrespective of the gate level. All gate edges applied during the counting sequence will store the current count in the Hold register, but they will not interrupt the counting sequence. On each TC, the counter will be reloaded from the Load register and stopped. Subsequent counting requires a new triggering Gate edge; counting resumes on the first source edge following the triggering Gate edge.

Note: Mode X is only available in the Am9513A devices.

COUNTER MODE CONTROL OPTIONS

Each Counter Logic Group includes a 16-bit Counter Mode (CM) register used to control all of the individual options available with its associated general counter. These options include output configuration, count control, count source and gating control. Figure 16 shows the bit assignments for the Counter Mode registers. This section describes the control options in detail. Note that generally each counter is independently configured and does not depend on information outside its Counter Logic Group. The Counter Mode register should be loaded only when the counter is Disarmed. Attempts to load the Counter Mode register when the counter is armed may result in erratic counter operation.

After power-on reset or a Master Reset command, the Counter Mode registers are initialized to a preset condition. The value entered is 0B00 hex and results in the following control configuration:

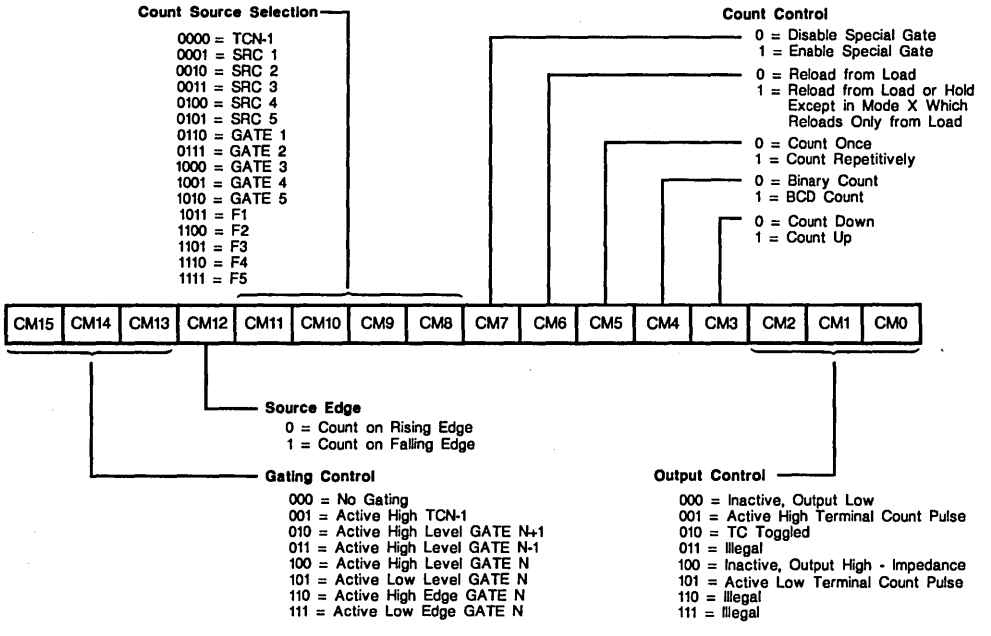
Output low-impedance to ground
Count down
Count binary
Count once
Load register selected
No retriggering
F1 input source selected
Positive-true input polarity
No gating

Output Control

Counter mode bits CM0 through CM2 specify the output control configuration. Figure 17 shows a schematic representation of the output control logic. The OUT pin may be off (a high-impedance state), or it may be inactive with a low-impedance to ground. The three remaining valid combinations represent the active-high, active-low or TC Toggle output waveforms.

One output form available is called Terminal Count (TC) and represents the period in time that the counter reaches an equivalent value of zero. TC will occur on the next count when the counter is at 0001 for down counting, at 9999 (BCD) for BCD up counting or at FFFF (hex) for binary up counting. Figure 18 shows a Terminal Count pulse and an example context that generated it. The TC width is determined by the period of the counting source. Regardless of any gating input or whether the counter is Armed or Disarmed, the terminal count will go active for only one clock cycle. Figure 18 assumes active-high source polarity, counter armed, counter decrementing and an external reload value of K.

The counter will always be loaded from an external location when TC occurs; the user can choose the source location and the value. If a non-zero value is picked, the counter will never really attain a zero state, and TC will indicate the counter state that would have been zero had no parallel transfer occurred.



Note: See Figure 15 for restrictions on Count Control and Gating Control bit combinations.

Figure 16. Counter Mode Register Bit Assignments

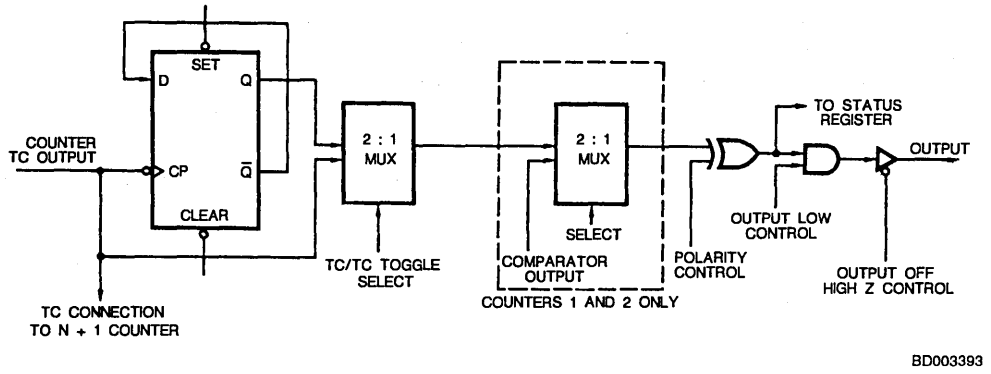


Figure 17. Output Control Logic

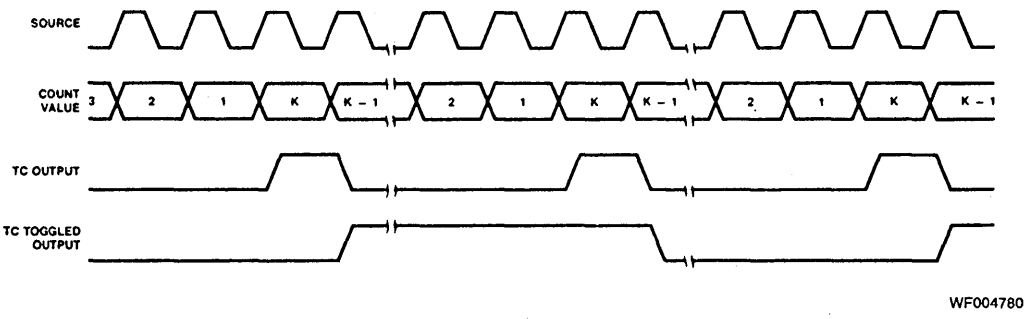


Figure 18. Counter Output Waveforms

The other output form, TC Toggled, uses the trailing edge of TC to toggle a flip-flop to generate an output level instead of a pulse. The toggle output is half the frequency of TC. The TC Toggled output will frequently be used to generate variable duty-cycle square waves in Operating Modes G through K.

In Mode L the TC Toggled output can be used to generate a one-shot function, with the delay to the start of the output pulse and the width of the output pulse separately programmable. With selection of the minimum delay to the start of the pulse, the output will toggle on the second source pulse following application of the triggering Gate edge.

Note that the TC Toggled output form contains no implication about whether the output is active-high or active-low. Unlike the TC output, which generates a transient pulse which can clearly be active-high or active-low, the TC Toggled output waveform only flips the state of the output on each TC. The sole criterion of whether the TC Toggled output is active-high or active-low is the level of the output at the start of the count cycle. This can be controlled by the Set and Clear Output commands. (See Figure 19.)

TC (Terminal Count)

On each Terminal Count (TC), the counter will reload itself from the Load or Hold register. TC is defined as that period of time when the counter contents would have been zero had no reload occurred. Some special conditions apply to counter operation immediately before and during TC.

1. In the clock cycle before TC, an internal signal is generated that commits the counter to go to TC on the next count, and retriggering by a hardware Gate edge (Modes N, O, Q and R) or a software LOAD or LOAD AND ARM command will not extend the time to TC. Note that the "next count" driving the counter to TC can be caused by the application of a count source edge (in level gating modes, the edge must occur while the gate is active, or it will be disregarded), by the application of a LOAD or LOAD AND ARM command (see 2 below) or by the application of a STEP command.
2. If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will immediately go to TC. If these commands are issued during TC, the TC state will immediately terminate.
3. When TC is active, the counter will always count the next source edge issued to it, even if it is disarmed or gated off during TC. This means that TC will never be active for longer than one count period and it may, in fact, be shorter if a STEP command or a LOAD or LOAD AND ARM command is applied during TC (see item 2 above). This also means that a counter that is disarmed or stopped on TC is actually disarmed/stopped immediately following TC.

This may cause count sequences different from what a user might expect. Since the counter is always reloaded at the start of TC and since it always counts at the end of TC, the counter contents following TC will differ by one from the reloaded value, irrespective of the operating mode used.

If the reloaded value was 0001 for down counting, 9999 (BCD) for BCD up counting or FFFF (hex) for binary up counting, the count at the end of TC will drive the counter into TC again regardless of whether the counter is gated off or disarmed. As long as these values are reloaded, the TC output will stay active. If a TC Toggled output is selected, it will toggle on each count. Execution of a LOAD, LOAD AND ARM or STEP command with these counter contents will act the same as application of a source pulse, causing TC to remain active and a TC Toggled output to toggle.

Count Control

Counter Mode bits CM3 through CM7 specify the various options available for direct control of the counting process. CM3 and CM4 operate independently of the others and control up/down and BCD/binary counting. They may be combined freely with other control bits to form many types of counting configurations. The other three bits and the Gating Control field interact in complex ways. Bit CM5 controls the repetition of the count process. When CM5 = 1, counting will proceed in the specified mode until the counter is disarmed. When CM5 = 0, the count process will proceed only until one full cycle of operation occurs. This may occur after one or two TC events. The counter is then disarmed automatically. The single or double TC requirement will depend on the state of other control bits. Note that even if the counter is automatically disarmed upon a TC, it always counts the count source edge which generates the trailing TC edge.

When TC occurs, the counter is always reloaded with a value from either the Load register of the Hold register. Bit CM6 specifies the source options for reloading the counter. When CM6 = 0, the contents of the Load register will be transferred into the counter at every occurrence of TC. When CM6 = 1, the counter reload location will be either the Load or Hold Register. The reload location in this case may be controlled externally by using a Gate pin (Modes S and V) or may alternate on each TC (Modes G through L). With alternating sources and with the TC Toggled output selected, the duty cycle of the output waveform is controlled by the relative Load and Hold values and very fine resolution of duty cycles ratios may be achieved.

Bit CM7 controls the special gating functions that allow retriggering and the selection of Load or Hold sources for counter reloading. The use and definition of CM7 will depend

on the status of the Gating Control field and bits CM5 and CM6.

Hardware Retriggering

Whenever hardware retriggering is enabled (Modes N, O, Q, and R), all active-going Gate edges initiate retrigger operations. On application of the Gate edge, the counter contents will be transferred to the Hold register. On the first qualified source edge after application of the retriggering Gate edge, the Load register contents will be transferred into the counter. (Qualified source edges are edges which occur while the counter is gated on and Armed.)

This means that, if level gating is used, the edge occurring on active-going gate transitions will initiate a retrigger. Similarly, when edge gating is enabled, an edge used to start the counter will also initiate a retrigger. The first count source edge applied after the Gate edge will not increment/decrement the counter but retrigger it.

If a LOAD, LOAD AND ARM, or a STEP Command occurs between the retriggering Gate edge and the first qualified source edge, it will be interpreted as a source edge and transfer the Load register contents into the counter. Thereafter, the counter will count all qualified source edges.

When some form of Gating is specified, CM7 controls hardware retriggering. In this case, when CM7 = 0, hardware retriggering does not occur; when CM7 = 1, the counter is retriggered any time an active-going Gate edge occurs. Retriggering causes the counter value to be saved in the Hold register and the Load register contents to be transferred into the counter.

When No Gating is specified, the definition of CM7 changes. In this case, when CM7 = 0, the Gate input has no effect on the counting; when CM7 = 1, the Gate input specifies the source (selecting either the Load or Hold register) used to reload the counter when TC occurs. Figure 14 shows the various available control combinations for these interrelated bits.

Count Source Selection

Counter Mode bits CM8 through CM12 specify the source used as input to the counter and the active edge that is counted. Bit CM12 controls the polarity for all the sources; logic zero counts rising edges and logic one counts falling edges. Bits CM8 through CM11 select 1 of 16 counting sources to route to the counter input. Five of the available inputs are internal frequencies derived from the internal oscillator (see Figure 13 for frequency assignments). Ten of the available inputs are interface pins; five are labeled SRC and five are labeled GATE.

The 16th available input is the TC output from the adjacent lower-numbered counter. (The Counter 5 TC wraps around to the Counter 1 input.) This option allows internal concatenating that permits very long counts to be accumulated. Since all five counters may be concatenated, it is possible to configure a counter that is 80-bits long on one Am9513A chip. When TCN-1 is the source, the count ripples between the connected counters. External connections can also be made, and can use the toggle bit for even longer counts. This is easily accomplished by selecting a TC Toggled output mode and wiring OUTN to one of the SRC inputs.

Gating Control

Counter Mode bits CM15, CM14, CM13 specify the hardware gating options. When "no gating" is selected (000), the counter will proceed unconditionally as long as it is armed. For any other gating mode, the count process is conditioned by the specified gating configuration.

For a code of 100 in this field, counting can proceed only when the pin labeled GATEN associated with Counter N is at a logic high level. When it goes LOW, counting is simply suspended until the Gate goes HIGH again. A code of 101 performs the same function with an opposite active polarity. Codes 010 and 011 offer the same function as 100, but specify alternate input pins as Gating Sources. This allows any of three interface pins to be used as gates for a given counter. On Counter 4, for example, pin 34, pin 35 or pin 36 may be used to perform the gating function. This also allows a single Gate pin to simultaneously control up to three counters. Counters 1 and 5 are considered adjacent when using TCN-1 (001), Gate N + 1 (010) and Gate N – 1 (011) controls.

For codes of 110 or 111 in this field, counting proceeds after the specified active Gate edge until one or two TC events occur. Within this interval, the Gate input is ignored, except for the retriggering option. When repetition is selected, a cycle will be repeated as soon as another Gate edge occurs. With repetition selected, any Gate edge applied after TC goes active will start a new count cycle. Edge gating is useful when implementing a digital single-shot since the gate can serve as a convenient firing trigger.

A 001 code in this field selects the TC (not TOGGLE) output from the adjacent lower-numbered counter as the gate. This is useful for synchronous counting when adjacent counters are concatenated.

COMMAND DESCRIPTIONS

The command set for the Am9513A allows the host processor to customize and manage the operating modes and features for particular applications, to initialize and update both the internal data and control information, and to manipulate operating bits during operation. Commands are entered directly into the 8-bit Command register by writing into the Control port (see Figure 6).

All available commands are described in the following text. Figure 19 summarizes the command codes and includes a brief description of each function. Figure 20 shows all the unused code combinations; unused codes should not be entered into the Command register since undefined activities may occur.

Six of the command types are used for direct software control of the counting process and they each contain a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of five general counters (S1 = Counter 1, S2 = Counter 2, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation occurs for the corresponding counter. This type of command format has three basic advantages. It saves host software by allowing any combination of counters to be acted on by a single command. It allows simultaneous action on multiple counters where synchronization of commands is important. It allows counter-specific service routines to control individual counters without needing to be aware of the operating context of other counters.

Three of the commands use a 3-bit binary code (N4, N2, N1) to identify the affected counter (a 001 programs counter 1, etc.). Unlike the previously mentioned commands, these commands allow you to program only one counter at a time.

Command Code								Command Description
C7	C6	C5	C4	C3	C2	C1	C0	
0	0	0	E2	E1	G4	G2	G1	Load Data Pointer register with contents of E and G fields. (G ≠ 000, G ≠ 110)
0	0	1	S5	S4	S3	S2	S1	Arm counting for all selected counters
0	1	0	S5	S4	S3	S2	S1	Load contents of specified source into all selected counters
0	1	1	S5	S4	S3	S2	S1	Load and Arm all selected counters*
1	0	0	S5	S4	S3	S2	S1	Disarm and Save all selected counters
1	0	1	S5	S4	S3	S2	S1	Save all selected counters in Hold register
1	1	0	S5	S4	S3	S2	S1	Disarm all selected counters
1	1	1	0	1	N4	N2	N1	Set Toggle out (HIGH) for counter N (001 ≤ N ≤ 101)
1	1	1	0	0	N4	N2	N1	Clear Toggle out (LOW) for counter N (001 ≤ N ≤ 101)
1	1	1	1	0	N4	N2	N1	Step counter N (001 ≤ N ≤ 101)
1	1	1	0	1	0	0	0	Set MM14 (Disable Data Pointer Sequencing)
1	1	1	0	1	1	1	0	Set MM12 (Gate off FOUT)
1	1	1	0	1	1	1	1	Set MM13 (Enter 16-bit bus mode)
1	1	1	0	0	0	0	0	Clear MM14 (Enable Data Pointer Sequencing)
1	1	1	0	0	1	1	0	Clear MM12 (Gate on FOUT)
1	1	1	0	0	1	1	1	Clear MM13 (Enter 8-bit bus mode)
1	1	1	1	1	0	0	0	Enable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	0	0	1	Disable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	1	1	1	Master reset

*Not to be used for asynchronous operations.

Figure 19. Am9513A Command Summary

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	0	1	1	1
0	0	0	X	X	1	1	0
0	0	0	X	X	0	0	0
*1	1	1	1	1	X	X	X

*Unused except when XXX = 111, 001 or 000.

Figure 20. Am9513A Unused Command Codes

Arm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be enabled for counting. A counter must be armed before counting can commerce. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. This command can only arm or do nothing for a given counter; a zero in the S field does not disarm the counter.

ARM and DISARM commands can be used to gate counter operation on and off under software control. DISARM commands entered while a counter is in the TC state will not take effect until the counter leaves TC. This ensures that the counter never latches up in a TC state. (The counter may leave the TC state because of application of a count source edge, execution of a LOAD or LOAD AND ARM command, or execution of a STEP command.)

In modes which alternate reload sources (Modes G – L), the ARMing operation is used as a reset for the logic which determines which reload source to use on the upcoming TC. Following each ARM or LOAD AND ARM command, a counter in one of these modes will reload from the Hold register on the first TC and alternate reload sources thereafter (reload from the Load register on the second TC, the Hold register on the third, etc.).

Load Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be loaded with previously entered values. The source of information for each counter will be either the associated Load register or the associated Hold register, as determined by the operating configuration in the Mode register. The Load/Hold contents are not changed. This command will cause a transfer independent of any current operating configuration for the counter. It will often be used as a software retrigger or as counter initialization prior to active hardware gating.

If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will go immediately to TC. This occurs because the LOAD operation is performed by generating a pseudo-count pulse internal to the Am9513A, and the Am9513A is expecting to go into TC on the next count pulse. The reload source used to reload the counter will be the same as that which would have been used if the TC were generated by a source edge rather than by the LOAD operation.

Execution of a LOAD or LOAD AND ARM command while the counter is in TC will cause the TC to end. For Armed counters in all modes except S or V, the LOAD source used will be that to be used for the upcoming TC. (The LOADing operation will not alter the selection of reload source for the upcoming TC.) For Disarmed counters in modes except S or V, the reload sources used will be the LOAD register. For modes S or V, the reload source will be selected by the GATE input, regardless of whether the counter is Armed or Disarmed.

Special considerations apply when modes with alternating reload sources are used (Modes G – L). If a LOAD command drives the counter to TC in these modes, the reload source for the next TC will be from the opposite reload location. In other words, the LOAD-generated TC will cause the reload sources to alternate just as a TC generated by a source edge would. Note that if a second LOAD command is issued during the LOAD-generated TC (or during any other TC, for that matter), the second LOAD command will terminate the TC and cause a reload from the source designated for use with the next TC. The second LOAD will not alter the reload source for the next TC since the second LOAD does not generate a TC; reload sources alternate on TCs only, not on LOAD commands.

Load and Arm Counters*

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be first loaded and then armed. This command is equivalent to issuing a LOAD command and then an ARM command.

A LOAD AND ARM command which drives a counter to TC generates the same sequence of operations as execution of a LOAD command and then an ARM command. In modes which disarm on TC (Modes A – C and N – O, and Modes G – I and S if the current TC is the second in the cycle), the ARM part of the LOAD AND ARM command will re-enable counting for another cycle. In modes which alternate reload sources (Modes G – L), the ARMing operation will cause the next TC to reload from the HOLD register, irrespective of which reload source the current TC used.

*This command should not be used during asynchronous operations.

Disarm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disabled from counting. A disarmed counter will cease all counting independent of other conditions. The only exception to this is that a counter in the TC state will always count once, in order to leave TC, before DISARMing. This count may be generated by a source edge, by a LOAD or LOAD AND ARM command (the LOAD AND ARM command will negate the DISARM command) or by a STEP command. A disarmed counter may be updated using the LOAD command and may be read using the SAVE command. A count process may be resumed using an ARM command. See the ARM command description for further details.

Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by S field, will have their contents transferred into their associated Hold register. The transfer takes place without interfering with any counting that may be underway. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Disarm and Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disarmed, and the contents of the counter will be transferred into the associated Hold registers. This command is identical to issuing a DISARM command followed by a SAVE command.

Set TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: The initial output level for TC Toggle mode is set (HIGH) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 17) but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Clear TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: The initial output level for TC Toggle mode is Cleared (LOW) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 17) but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Step Counter

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: Counter N is incremented or decremented by one, depending on its operating configuration. If the Counter Mode register associated with the selected counter has its CM3 bit cleared to zero, this command will cause the counter to decrement by one. If CM3 is set to a logic high, this command will increment the counter by one. The STEP command will take effect even on a disarmed counter.

Load Data Pointer Register

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	E2	E1	G4	G2	G1

(G4, G2, G1 $\neq$ 000, $\neq$ 110)

Description: Bits in the E and G fields will be transferred into the corresponding Element and Group fields of the Data Pointer register as shown in Figure 7. The Byte Pointer bit in the Data Pointer register is set. Transfers into the Data Pointer only occur for G field values of 001, 010, 011, 100, 101 and 111. Values of 000 and 110 for G should not be used. See the "Setting the Data Pointer Register" section of this document for additional details.

Disable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	0	0	0

Description: This command sets Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Disabling the sequencing allows repetitive host processor access to a given internal location without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register.

Enable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	0	0	0

Description: This command clears Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Enabling the sequencing allows sequential host processor access to several internal locations without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register. See the "Data Pointer Register" section of this document for additional information on Data Pointer sequencing

Enable 16-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	1

Description: This command sets Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is set, no multiplexing takes place and all 16 external data bus lines are used to transfer information into and out of the STC. MM13 may also be controlled by loading the full Master Mode register in parallel.

Enable 8-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	1

Description: This command clears Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is cleared, the multiplexer is enabled and 16-bit internal information is transferred eight bits at a time to the eight low-order external data bus lines. MM13 may also be controlled by loading the full Master Mode register in parallel.

Gate Off FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	0

Description: This command sets Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output state of the FOUT signal. When gated off, the FOUT line will exhibit a low-impedance to ground. MM12 may also be controlled by loading the full Master Mode register in parallel.

Gate On FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	0

Description: This command clears Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output status of the FOUT signal. When MM12 is cleared, FOUT will become active and will drive out the selected and divided FOUT signal. MM12 may also be controlled by loading the full Master Mode register in parallel. When FOUT is gated on or off, a transient pulse may be generated on the FOUT signal.

Disable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	1

Description: This command disables the prefetch circuitry during Write operations (if does not affect Read operations). This reduces the write recovery time and allows the user to use block move instructions for initialization of the Am9513A registers. Once prefetch is disabled for writing, an Enable Prefetch for Write or a Reset command is necessary to re-enable the prefetch circuitry for writing.
Note: This command is only available in Am9513A devices; it is an illegal command in the "non-A" Am9513 device.

Enable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	0

Description: This command re-enables the prefetch circuitry for Write operations. It is used only to terminate the Disable Prefetch Command.
Note: This command is only available in Am9513A devices; it is an illegal command in the "non-A" Am9513 device.

Master Reset

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	1	1	1

Description: The Master Reset command duplicates the action of the power-on reset circuitry. It disarms all counters, enters 0000 in the Master Mode, Load and Hold registers and enters 0B00 (hex) in the Counter Mode registers.

Following either a power-up or software reset, the LOAD command should be applied to all the counters to clear any that may be in a TC state. The Data Pointer register should also be set to a legal value, since reset does not initialize it. A complete reset operation follows.

1. Using the procedure given in the "Command Initiation" section of this document, enter the FF (hex) command to perform a software reset.
2. Using the "Command Initiation" procedure, enter the LOAD command for all counters, opcode 5F (hex).
3. Using the procedure given in the "Setting the Data Pointer Register" section of this document, set the Data Pointer to a valid code. The legal Data Pointer codes are given in Figure 8.

The Master Mode, Counter Mode, Load and Hold registers can now be initialized to the desired values.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
VCC with Respect to VSS ..... −0.5 V to +7.0 V
All Signal Voltages
with Respect to VSS ..... −0.5 V to +7.0 V
Power Dissipitation (Package Limitation) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

Military (M) Devices
Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... 5 V ±5%

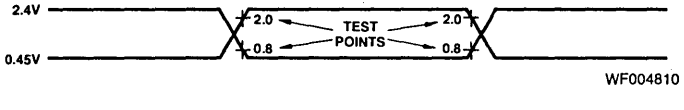
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified.

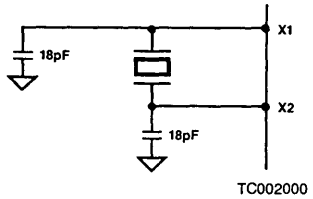
Parameters	Description		Test Conditions	Min	Max		Units
VIL	Input Low Voltage	All Inputs Except X2		VSS − 0.5	0.8		Volts
		X2 Input		VSS − 0.5	0.8		
VIH	Input High Voltage	All Input Except X2		2.2 V	VCC		Volts
		X2 Input		3.8	VCC		
VITH	Input Hysteresis (SRC and GATE Inputs Only)			0.2			Volts
VOL	Output Low Voltage		IOL = 3.2 mA		0.4		Volts
VOH	Output High Voltage		IOH = −200 μA	2.4			Volts
IIX	Input Load Current (Except X2)		VSS ≤ VIN ≤ VCC		±10		μA
IIX	Input Load Current X2		VSS ≤ VIN ≤ VCC		±100		μA
IOZ	Output Leakage Current (Except X1)		VSS + 0.4 ≤ VOUT ≤ VCC High-Impedance State		±25		μA
ICC	VCC Supply Current (Steady State)				255	275	mA
CIN	Input Capacitance		f = 1 MHz, T_A = +25°C. All pins not under test at 0 V.		10*	20*	pF
COUT	Output Capacitance				15*	20*	
CIO	IN/OUT Capacitance				20*	20*	

* Guaranteed by design.

SWITCHING TEST INPUT/OUTPUT WAVEFORMS



WF004810



TC002000

Crystal is fundamental mode parallel resonant 32 pF load capacitance less than 100 Ω ESR C_o less than 100 pF.

The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = HIGH
L = LOW
V = VALID
X = Unknown or Don't care
Z = High-Impedance

2. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.

3. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write and the counter may be off by one count.

4. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.

5. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation and the counter may be off by one count.

6. This parameter applies to cases where the write operation causes a change in the output bit.

7. The enabled count source is one of F1 – F5, TCN-1 SRC1 – SRC5 or GATE1 – GATE 5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.

8. This parameter applies to edge gating (CM15 – CM13 = 110 or 111) and gating when both CM7 = 1 and CM15 – CM13 ≠ 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.

9. This parameter applies to both edge and level gating (CM15 – CM13 = 001 through 111 and CM7 = 0). This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge and the counter may be off by one count.

10. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.

11. Signals F1 – F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals. F1 = X2.

12. This timing specification assumes that $\overline{CS}$ is active whenever $\overline{RD}$ or $\overline{WR}$ are active. $\overline{CS}$ may be held active indefinitely.

13. This parameter assumes X2 is driven from an external gate with a square wave.

14. This parameter assumes that the write operation is to the command register.

15. This timing specification applies to single-action commands only (e.g., LOAD, ARM, SAVE, etc.). For double-action commands such as LOAD AND ARM and DISARM AND SAVE, TWHEH minimum = 700 ns.

16. In short data write mode, TWHRL and TWHWL minimum = 1000 ns.

17. This parameter applies to the hardware retrigger/save modes N, O, Q, R, and X (CM7 = 1 and CM15 – CM13 < > 000). This parameter ensures that the gating pulse initiates a hardware retrigger/save operation.

18. This parameter applies to hardware load source select modes S and V (CM7 = 1 and CM15 – CM13 = 000). This parameter represents the minimum hold time to ensure that the GATE input selects the correct load source on the active source edge.

SWITCHING CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (Note 1)

Parameters	Description		Figure	Am9513A Min	Am9513A Max	Unit
TAVRL	C/$\overline{D}$ Valid to Read Low		21	25		ns
TAVWH	C/$\overline{D}$ Valid to Write High		21	170		ns
TCHCH	X2 High to X2 High (X2 Period) (Note 13)		22	145		ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width) (Note 13)		22	70		ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width) (Note 13)		22	70		ns
TDVWH	Data In Valid to Write High		21	80		ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 7)		22	145		ns
TEHEL TELEH	Count Source Pulse Duration (Note 7)		22	70		ns
TEHFV	Count Source High to FOUT Valid (Note 7)		22		500	ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 7, 9, 10)		22	10		ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 2, 7)		21	190		ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 3, 7)		21	−100		ns
TEHYV	Count Source High to Out Valid (Note 7)	TC Output	22		300	ns
		Immediate or Delayed Toggle Output	22		300	
		Comparator Output	22		350	
TFN	FN High to FN + 1 Valid (Note 11)		22		75	ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 7, 9, 10)		22	100		ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 8, 10)		22	145		ns
TGVWH	Gate Valid to Write High (Notes 3, 10)		21	−100		ns
TRHAX	Read High to C/$\overline{D}$ Don't Care		21	0		ns
TRHEH	Read High to Count Source High (Notes 4, 7)		21	0		ns
TRHQX	Read High to Data Out Invalid		21	10		ns
TRHQZ	Read High to Data Out at High-Impedance (Data Bus Release Time)		21		85	ns
TRHRL	Read High to Read Low (Read Recovery Time)		21	1000		ns
TRHSH	Read High to $\overline{CS}$ High (Note 12)		21	0		ns
TRHWL	Read High to Write Low (Read Recovery Time)		21	1000		ns
TRLQV	Read Low to Data Out Valid		21		110	ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)		21	20		ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 12)		21	160		ns
TSLRL	$\overline{CS}$ Low to Read Low (Note 12)		21	20		ns
TSLWH	$\overline{CS}$ Low to Write High (Note 12)		21	170		ns
TWHAX	Write High to C/$\overline{D}$ Don't Care		21	20		ns
TWHDX	Write High to Data In Don't Care		21	20		ns
TWHEH	Write High to Count Source High (Notes 5, 7, 14, 15)		21	550		ns
TWHGV	Write High to Gate Valid (Notes 5, 10, 14)		21	475		ns
TWHRL	Write High to Read Low (Write Recovery Time) (Note 16)		21	1500*		ns
TWHSH	Write High to $\overline{CS}$ High (Note 12)		21	20		ns
TWHWL	Write High to Write Low (Write Recovery Time) (Note 16)		21	1500*		ns
TWHYV	Write High to Out Valid (Notes 6, 14)		21		650	ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 12)		21	150		ns
TGVEH2	Gate Valid to Count Source High (Special Gate) (Notes 10, 13, 17)		22	200		ns
TEHGV2	Count Source High to Gate Valid (Special Gate) (Notes 10, 13, 18)		22	80		ns

Notes:

1. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

   A (Address) = C/$\overline{D}$
   C (Clock) = X2
   D (Data In) = DB0-DB15
   E (Enabled counter source input) = SRC1 – SRC5, GATE1 – GATE5, F1 – F5,TCN-1
   F = FOUT
   G (Counter gate input) = GATE1 – GATE5, TCN-1
   Q (Data Out) = DB0 – DB15
   R (Read) = $\overline{RD}$
   S (Chip Select) = $\overline{CS}$
   W (Write) = $\overline{WR}$
   Y (Output) = OUT1 – OUT5

2

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for SMD/DESC and APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

Parameter Symbol	Description		Am9513A Min.	Am9513A Max.	Unit
TAVRL	C/$\overline{D}$ Valid to Read Low		25		ns
TAVWH	C/$\overline{D}$ Valid to Write High		170		ns
TCHCH	X2 High to X2 High (X2 Period) (Note 13)		145		ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width) (Note 13)		70		ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width) (Note 13)		70		ns
TDVWH	Data In Valid to Write High		80		ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 7)		145		ns
TEHEL TELEH	Count Source Pulse Duration (Note 7)		70		ns
TEHFV	Count Source High to FOUT Valid (Note 7)			500	ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 7, 9, 10)		10		ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 2, 7)		190		ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 3, 7)		−100		ns
TEHYV	Count Source High to Out Valid (Note 7)	TC Output		300	ns
		Immediate or Delayed Toggle Output		300	
		Comparator Output		350	
TFN	FN High to FN + 1 Valid (Note 11)			75	ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 7, 9, 10)		100		ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 8, 10)		145		ns
TGVWH	Gate Valid to Write High (Notes 3, 10)		−100		ns
TRHAX	Read High to C/$\overline{D}$ Don't Care		0		ns
TRHEH	Read High to Count Source High (Notes 4, 7)		0		ns
TRHQX	Read High to Data Out Invalid		10		ns
TRHQZ	Read High to Data Out at High-Impedance (Data Bus Release Time)			85	ns
TRHRL	Read High to Read Low (Read Recovery Time)		1000		ns
TRHSH	Read High to $\overline{CS}$ High (Note 12)		0		ns
TRHWL	Read High to Write Low (Read Recovery Time)		1000		ns
TRLQV	Read Low to Data Out Valid			110	ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)		20		ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 12)		160		ns
TSLRL	$\overline{CS}$ Low to Read Low (Note 12)		20		ns
TSLWH	$\overline{CS}$ Low to Write High (Note 12)		170		ns
TWHAX	Write High to C/$\overline{D}$ Don't Care		20		ns
TWHDX	Write High to Data In Don't Care		20		ns
TWHEH	Write High to Count Source High (Notes 5, 7, 14, 15)		550		ns
TWHGV	Write High to Gate Valid (Notes 5, 10, 14)		475		ns
TWHRL	Write High to Read Low (Write Recovery Time) (Note 16)		1500		ns
TWHSH	Write High to $\overline{CS}$ High (Note 12)		20		ns
TWHWL	Write High to Write Low (Write Recovery Time) (Note 16)		1500		ns
TWHYV	Write High to Out Valid (Notes 6, 14)			650	ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 12)		150		ns
TGVEH2	Gate Valid to Count Source High (Special Gate) (Notes 10, 13, 17)		200		ns
TEHGV2	Count Source High to Gate Valid (Special Gate) (Notes 10, 13, 18)		80		ns

Notes:

1. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

   A (Address) = C/$\overline{D}$
   C (Clock) = X2
   D (Data In) = DB0 – DB15
   E (Enabled counter source input) = SRC1 – SRC5, GATE1 – GATE5, F1 – F5,TCN-1
   F = FOUT
   G (Counter gate input) = GATE1 – GATE5, TCN-1
   Q (Data Out) = DB0 – DB15
   R (Read) = $\overline{RD}$
   S (Chip Select) = $\overline{CS}$
   W (Write) = $\overline{WR}$
   Y (Output) = OUT1 – OUT5

The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = HIGH
L = LOW
V = VALID
X = Unknown or Don't care
Z = High-Impedance

2. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.

3. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write and the counter may be off by one count.

4. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.

5. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation and the counter may be off by one count.

6. This parameter applies to cases where the write operation causes a change in the output bit.

7. The enabled count source is one of F1 – F5, TCN-1 SRC1 – SRC5 or GATE1 – GATE 5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.

8. This parameter applies to edge gating (CM15 – CM13 = 110 or 111) and gating when both CM7 = 1 and CM15 – CM13 ≠ 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.

9. This parameter applies to both edge and level gating (CM15 – CM13 = 001 through 111 and CM7 = 0). This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge and the counter may be off by one count.

10. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.

11. Signals F1 – F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals. F1 = X2.

12. This timing specification assumes that $\overline{CS}$ is active whenever $\overline{RD}$ or $\overline{WR}$ are active. $\overline{CS}$ may be held active indefinitely.

13. This parameter assumes X2 is driven from an external gate with a square wave.

14. This parameter assumes that the write operation is to the command register.

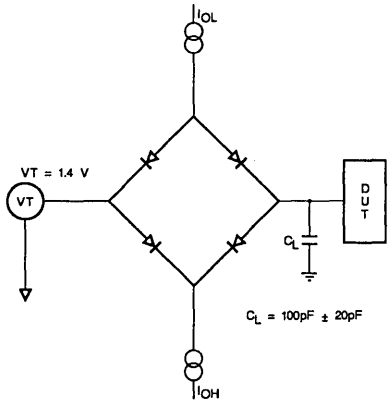
15. This timing specification applies to single-action commands only (e.g., LOAD, ARM, SAVE, etc.). For double-action commands such as LOAD AND ARM and DISARM AND SAVE, TWHEH minimum = 700 ns.

16. In short data write mode, TWHRL and TWHWL minimum = 1000 ns.

17. This parameter applies to the hardware retrigger/save modes N, O, Q, R, and X (CM7 = 1 and CM15 – CM13 < > 000). This parameter ensures that the gating pulse initiates a hardware retrigger/save operation.

18. This parameter applies to hardware load source select modes S and V (CM7 = 1 and CM15 – CM13 = 000). This parameter represents the minimum hold time to ensure that the GATE input selects the correct load source on the active source edge.

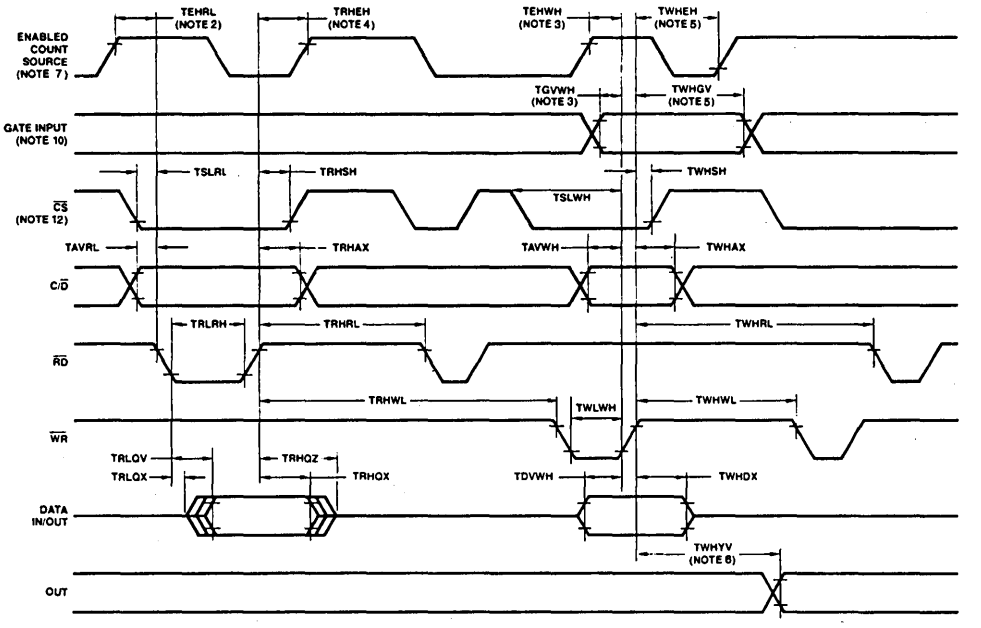
SWITCHING TEST CIRCUIT



TC003853

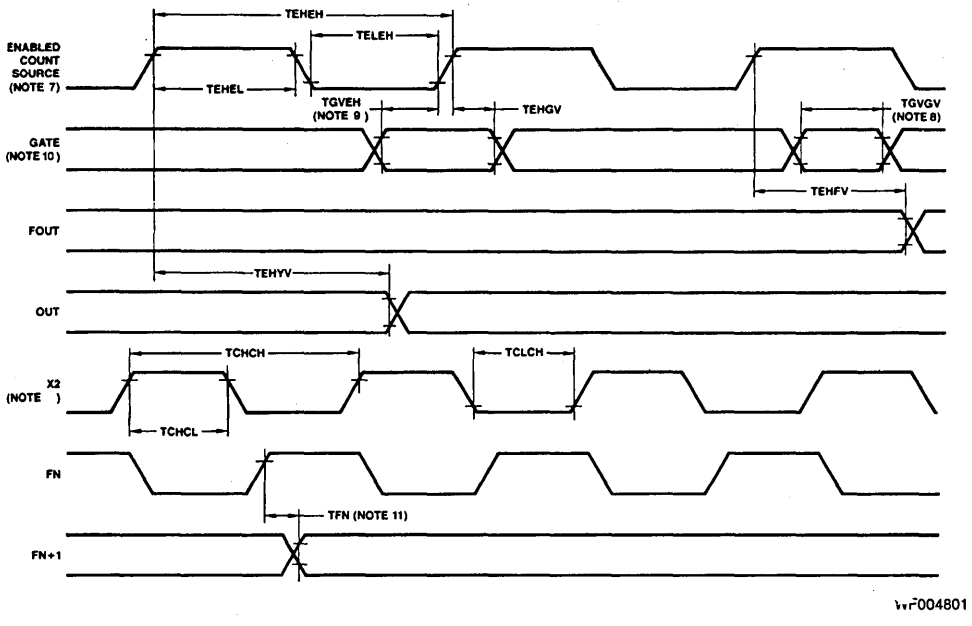
This test circuit is the dynamic load of a Teradyne J941.

SWITCHING WAVEFORMS



WF004792

Figure 21. Bus Transfer Switching Waveforms



WF004801

Figure 22. Counter Switching Waveforms

APPENDIX A

Design Hints

1) When a crystal is not being used, X1 and X2 should be connected as shown for TTL input (Figure A1) and no input (Figure A2).

2) Recommended oscillator capacitor values are 18 pF on X1 and X2.

3) Unused inputs should be tied to VCC.

4) The TC output can glitch when the counter is loaded. For this reason this output should not be connected to edge sensitive interrupts. The counter output should be set or cleared after the LOAD command.

5. The two most significant bits of the status register are not specified. They may be zero or one.

6. The mode register should not be modified when the counter is armed.

7. The LOAD and HOLD registers should not be changed during TC.

8. When using the different clocks for different counters be aware that there is a 75 ns skew between F1, F2, F3, F4 and F5.

9. The TC output will remain inactive if programmed to be in the TC TOGGLE mode and the step command is used to increment or decrement the counter. The output will go into TC if programmed to be in the active High or active Low terminal count modes. The only two ways out of TC in this case are:

   —Arming the counter and having an active source connected to it.

   —Issuing another step command.

10) Timing parameters TEHWH and TGVWH are specified as negative. The diagrams in Figure A3 show the relationship between these signals.

11) In mode X the counter will count all qualified source edges until the second (not the first) TC and then stop.

12) A TC can occur when the counters are loaded if the counter was stopped at $FFFF_H$ or 9999_{10} in the count up mode or at count 0001 when counting down. This is because an internal TC is generated which forces TC to be generated on the next count pulse.

13) In modes that alternate the reload source between the load and the hold registers (e.g., mode J), if the counter is disarmed at 0001_H for down counting or 9999_{10} for BCD up counting or $FFFF_H$ for binary up counting and rearmed, the reload source after the first TC will be the load register instead of the hold register. To avoid this, issue a software "dummy" load to the counter immediately after the disarm command.

14) In the down counting mode of the Am9513A, if a 0001 is loaded into the counter and another LOAD COUNTER command is issued, the TC of that counter will go active. If the load register contents are subsequently changed, and the counter armed, the first clock edge will cause the new load register contents to transfer into the counter and the next clock edge will decrement the counter and make it go out of TC.

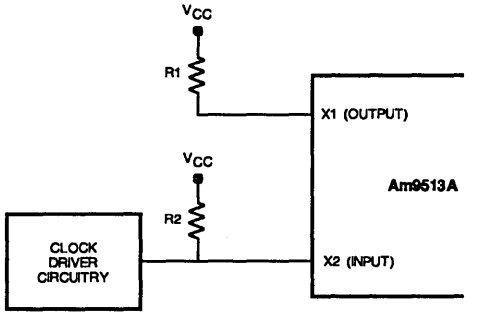
15) Glitches on $\overline{CS}$ just before the $\overline{RD}$ or $\overline{WR}$ pulse may cause the part to behave incorrectly.

16) Timing parameters TGVEH & TEHGV must not be violated; Figure A4 shows a method.

Troubleshooting (Symptom: Solution)

1) Registers not being programmed correctly: Check READ or WRITE recovery time.

2) Setup and hold problems observed in synchronous systems: Try switching from positive edge to negative edge triggering.



R1 = 6.8 kΩ ±10%
R2 is a function of Driver Circuitry to meet
X2 VIH = 3.8 V
X2 VIL = 0.8 V

Figure A1. Crystal Input Configuration

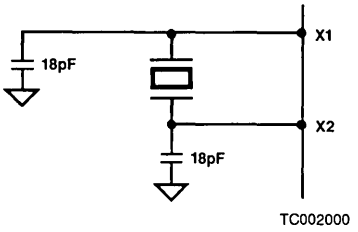


Figure A2. Crystal Input Configuration

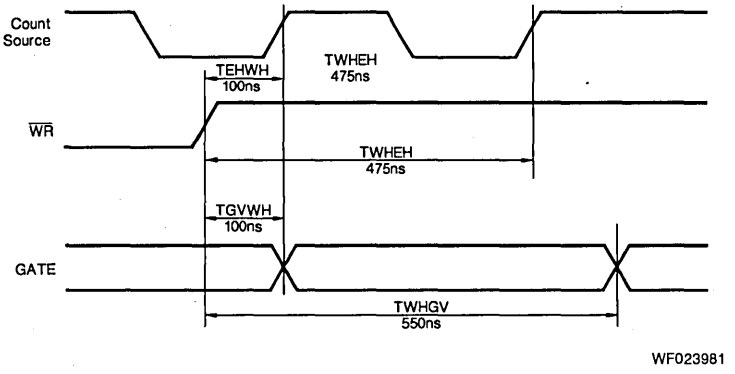


Figure A3. TEHWH/TGVWH Timing Diagram

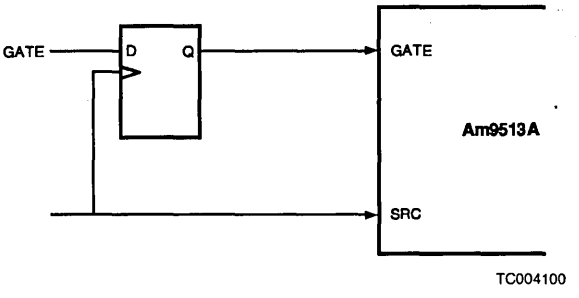


Figure A4. GATE/SRC Configuration Suggestion

Am9516A

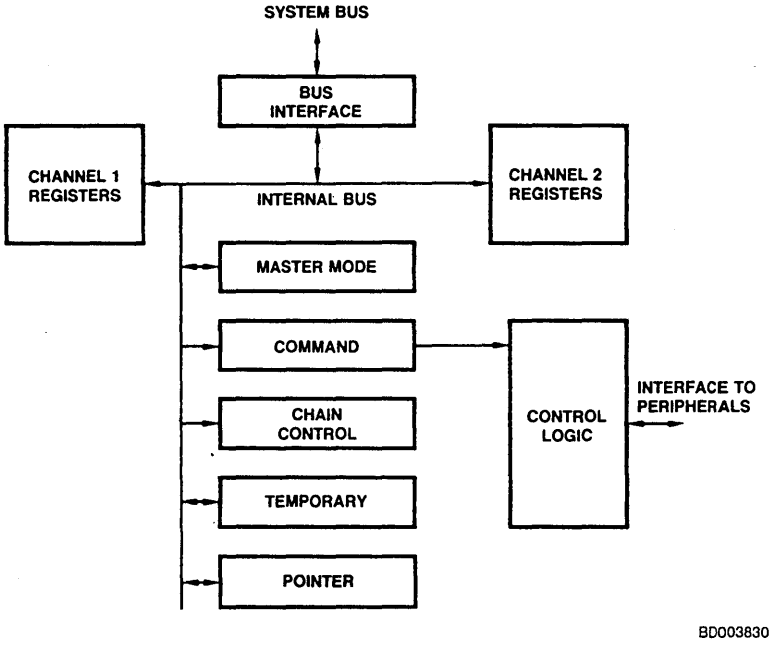
Universal DMA Controller (UDC)

FINAL

DISTINCTIVE CHARACTERISTICS

- Transfer Modes: Single, demand dedicated with bus hold, demand dedicated with bus release, demand interleave
- 16 MB physical addressing range
- Automatic loading/reloading of control parameters by each channel
- Optional automatic chaining of operations
- Channel interleave operations
- Interleave operations with system bus
- Masked data pattern matching for search operations
- Vectored interrupts on selected transfer conditions
- Software DMA request
- Software or hardware wait state insertion
- Transfer up to 6.66 MB/second at 10 MHz clock

BLOCK DIAGRAM



Publication # 03242 Rev. E Amendment /0
Issue Date: August 1989

GENERAL DESCRIPTION

The Am9516A Universal DMA Controller (UDC) is a high performance peripheral interface circuit for 8086 and 68000 CPUs. In addition to providing data block transfer capability between memory and peripherals, each of the UDC's two channels can perform peripheral-to-peripheral as well as memory-to-memory transfer. A special Search Mode of Operation compares data read from a memory or peripheral source to the content of a pattern register.

For all DMA operations (search, transfer, and transfer-and-search), the UDC can operate with either byte or word data sizes. In some system configurations it may be necessary to transfer between word-organized memory and a byte-oriented peripheral. The UDC provides a byte packing/unpacking capability through its byte-word funneling transfer or transfer-and-search option. Some DMA applications may continuously transfer data between the same two memory areas. These applications may not require the flexibility inherent in reloading registers from memory tables. To service these repetitive DMA operations, base registers are provided on each channel which re-initialize the current source and destination Address and Operation Count registers. To change the data transfer direction under CPU control, provision is made for reassigning the source address as a destination and the destination as a source, eliminating the need for actual reloading of these address registers.

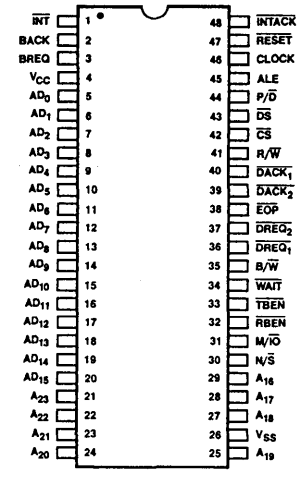
Frequently DMA devices must interface to slow peripherals or slow memory. In addition to providing a hardware $\overline{\text{WAIT}}$ input, the Am9516A UDC allows the user to select independently for both source and destination addresses and automatic insertion of 0, 1, 2 or 4 wait states. The user may even disable the $\overline{\text{WAIT}}$ input pin function altogether and use these software programmed wait states exclusively.

High throughput and powerful transfer options are of limited usefulness if a DMA requires frequent reloading by the host CPU. The Am9516A UDC minimizes CPU interactions by allowing each channel to load its control parameters from memory into the channel's control registers. The only action required of the CPU is to load the address of the control parameter table into the channel's Chain Address register and then issue a "Start Chain" Command to start the register loading operation.

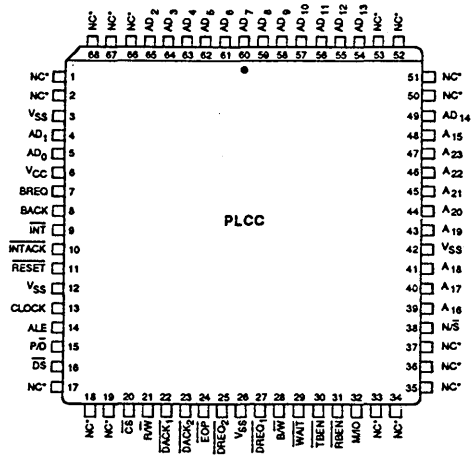
The Am9516A UDC is packaged in a 48-pin DIP and uses a single +5 V Power Supply.

CONNECTION DIAGRAM
Top View

DIPs



CD005592



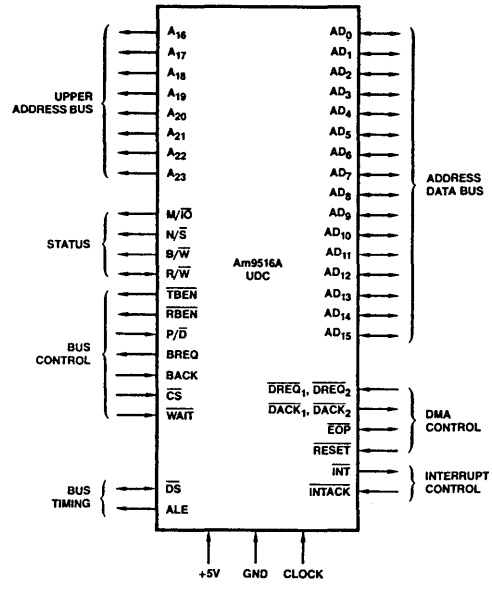
CD011970

*NC = No Connection

Note: Pin 1 is marked for orientation.

2

LOGIC SYMBOL

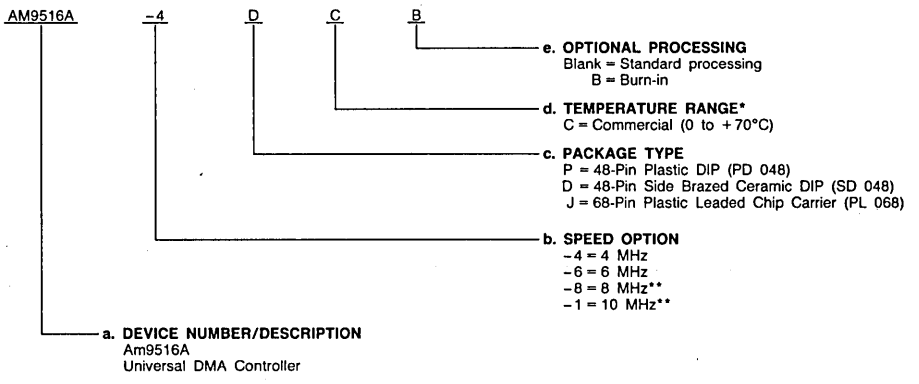


LS001331

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9516A-4	DC, DCB, PC, JC
AM9516A-6	
AM9516A-8	
AM9516A-1	DC

*This device is also available in Military temperature range.

**Preliminary; to be announced.

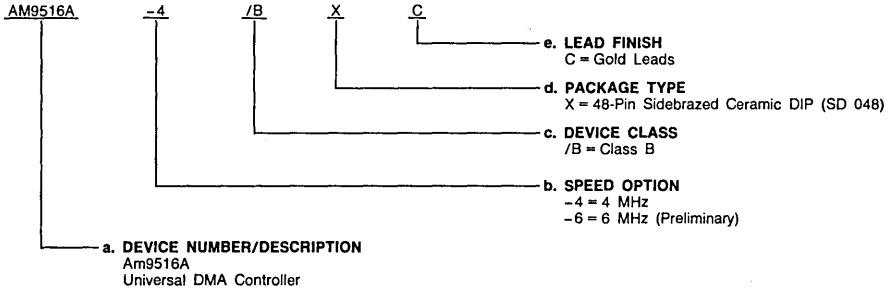
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM9516A-4	/BXC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
4	V_{CC}		+5 V Power Supply.
26	V_{SS}		Ground.
46	CLOCK	I	Clock.
46	CLOCK	I	(Clock). The Clock signal controls the internal operations and the rates of data transfers. It is usually derived from a master system clock or the associated CPU clock. The Clock input requires a high voltage input signal. Many UDC input signals can make transitions independent of the UDC clock; these signals can be asynchronous to the UDC clock. On other signals, such as $\overline{WAIT}$ inputs, transitions must meet setup and hold requirements relative to the UDC clock. See the Timing diagrams for details.
5-20	$AD_0 - AD_{15}$	I/O	(Address-Data Bus, Three-State). The Address Data Bus is a time-mulitiplexed, bidirectional, active-high, three-state bus used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD_0 is the least significant bit position and AD_{15} is the most significant. The presence of addresses is defined by the timing edge of ALE, and the asserted or requested presence of data is defined by the $\overline{DS}$ signal. The status output lines $M/\overline{IO}$ and $N/\overline{S}$ indicate the type of transaction, either memory or I/O. The $R/\overline{W}$ line indicates the direction of the transaction. When the UDC is in control of the system bus, it dominates the AD Bus; when the UDC is not in control of the system bus, the CPU or other external devices dominate the AD Bus. The presence of address of data on the $AD_0 - AD_{15}$ bus is defined only by ALE and $\overline{DS}$. When the UDC is not in control of the bus, there is no required relation between the presence of address or data and the UDC clock. This allows the UDC to be used with a system bus which does not have a bussed clock signal.
43	$\overline{DS}$	I/O	(Data Strobe, Three-State). Data Strobe is a bidirectional, active-low, three-state signal. A LOW on this signal indicates that the $AD_0 - AD_{15}$ bus is being used for data transfer. When the UDC is not in control of the system bus and the external system is transferring information to or from the UDC, $\overline{DS}$ is a timing input used by the UDC to move data to or from the $AD_0 - AD_{15}$ bus. Data is written into the UDC by the external ststem on the LOW-to-HIGH $\overline{DS}$ transition. Data is read from the UDC by the external system while $\overline{DS}$ is LOW. There are no timing requirements between $\overline{DS}$ as an input and the UDC clock; this allows use of the UDC with a system bus which does not have a bussed clock. During a DMA operation when the UDC is in control of the ststem, $\overline{DS}$ is an output generated by the UDC and used by the system to move data to or from the $AD_0 - AD_{15}$ bus. When the UDC has bus control, it writes to the external system by placing data on the $AD_0 - AD_{15}$ bus before the HIGH-to LOW $\overline{DS}$ transition and holding the data stable until after the LOW-to-HIGH $\overline{DS}$ transition; while reading from the external system, the LOW-to-HIGH transition of $\overline{DS}$ inputs data from the $AD_0 - AD_{15}$ bus into the UDC (see Timing diagram).
41	$R/\overline{W}$	I/O	(Read/Write, Three-State). Read/Write is a bidirectional, three-state signal. Read polarity is HIGH and write polarity is LOW. $R/\overline{W}$ indicates the data direction of the current bus transaction, and is stable starting when ALE is HIGH until the bus transaction ends (see Timing diagram). When the UDC is not in control of the system bus and the external system is transferring information to or from the UDC, $R/\overline{W}$ is a status input used by the UDC to determine if data is entering or leaving on the $AD_0 - AD_{15}$ bus during $\overline{DS}$ time. In such a case, Read (HIGH) indicates that the system is requesting data from the UDC, and Write (LOW) indicates that the system is presenting data to the UDC. There are no timing requirements between $R/\overline{W}$ as an input and the UDC clock; transitions on $R/\overline{W}$ as an input are only defined relative to $\overline{DS}$. When the UDC is in control of the system bus, $R/\overline{W}$ is an output generated by the UDC, with Read indicating that data is being requested from the addressed location or device. the addressed location or device and Write indicating that data is being presented to the addressed location or device. Flyby DMA operations are a special case where $R/\overline{W}$ is valid for the normally addressed memory or peripheral locations and must be interpreted in reverse by the "Flyby" peripheral that uses it.
33	$\overline{TBEN}$	O	(Transmit Buffer Enable, Open Drain). Transmit Buffer Enable is an active-low, open drain output. When UDC is a bus master, a LOW on this output indicates that the data is being transferred from the UDC to the data bus lines through the buffer. The purpose of this signal is to eliminate bus contention. When UDC is not in control of the system bus, these pins float to three-state OFF.
32	$\overline{RBEN}$	O	(Receive Buffer Enable, Open Drain). Receive Buffer Enable is an active-low, open drain output. When UDC is in control of system bus, a LOW on this output indicates that the data is being transferred from the data bus lines to the UDC through the buffer. The purpose of this signal is to eliminate bus contention. This pin floats to three-state OFF when the UDC is not in control of the system bus.
45	ALE	O	(Address Latch Enable). This active HIGH signal is provided by the UDC to latch the address signals $AD_0 - AD_{15}$ into the address latch. This pin is never floated.
44	$P/\overline{D}$	I	(Pointer/Data). Pointer/Data is an input signal to indicate the information is on the $AD_0 - AD_{15}$ bus only when the UDC is the bus slave. A HIGH on this signal indicates the information is on the AD bus is an address of the internal register to be accessed. The data on the AD bus is loaded into the Pointer register of UDC. A LOW on this signal indicates that a data transfer is taking place between the bus and the internal register designated by the Pointer register. Note that if a transaction is carried out with $R/\overline{W}$ HIGH and $P/\overline{D}$ HIGH, the contents of the Pointer register will be read.
31	$M/\overline{IO}$	O	(Memory/Input-Output, Three-State). This signal specifies the type of transaction. A HIGH on this pin indicates a memory transaction. A LOW on this pin indicates an I/O transaction. It floats to three-state OFF when UDC is not in control of the ststem bus.
30	$N/\overline{S}$	O	(Normal/System, Three-State). This output is a three-state signal activated only when the UDC is the bus master. Normal is indicated when $N/\overline{S}$ is HIGH, and System is indicated when $N/\overline{S}$ is LOW. This signal supplements the $M/\overline{IO}$ line and is used to indicate which memory or I/O space is being accessed.
35	$B/\overline{W}$	O	(Byte/Word, Three-State). This output indicates the size of data transferred on the $AD_0 - AD_{15}$ bus. HIGH indicates a byte (8-bit) and LOW indicates a word (16-bit) transfer. This output is activated when ALE is HIGH and remains valid for the duration of the whole transaction (see Timing diagram). All word-sized data are word-aligned and must be addressed by even addresses ($A_0 = 0$). When addressing byte read transactions, the least significant address bit determines which byte is needed; an even address specifies the most significant byte ($AD_8 - AD_{15}$), and an odd address specifies the least significant byte ($AD_0 - AD_7$). (Note that the higher address specifies the least significant byte!) This addressing mechanism applies to memory accesses as well as I/O accesses. When the UDC is a slave, it ignores the $B/\overline{W}$ signal and this pin floats to three-state OFF.

PIN DESCRIPTION (continued)

Pin No.	Name	I/O	Description
42	$\overline{CS}$	I	(Chip Select). This pin is an active-low input. A CPU or other external device uses $\overline{CS}$ to activate the UDC for reading and writing of its internal registers. There are no timing requirements between the $\overline{CS}$ input and the UDC clock; the $\overline{CS}$ input timing requirements are only defined relative to $\overline{DS}$. This pin is ignored when UDC is in control of system bus.
34	$\overline{WAIT}$	I	(WAIT). This pin is an active-low input. Slow memories and peripheral devices may use $\overline{WAIT}$ to extend $\overline{DS}$ and $\overline{RBEN}$ or $\overline{TBEN}$ during operation. Unlike the $\overline{CS}$ input, transitions on the $\overline{WAIT}$ input must meet certain timing requirements relative to the UDC clock. See Timing Diagram 4 for details. The Wait function may be disabled using a control bit in the Master Mode register (MM2).
3	BREQ	O	(Bus Request). Bus Request is an active-HIGH signalsignal used by the UDC to obtain control of the bus from the CPU. BREQ lines from multiple devices are connected to a priority encoder.
2	BACK	I	(Bus Acknowledge). BACK is an active-HIGH, asynchronchronous input, indicating that the CPU has relinquished the bus and that no higher priority device has assumed bus control. Since BACK is internally synchronized by the UDC before being used, transitions on BACK do not have to be synchronous with the UDC clock. The BACK input is usually connected to the HLDA line from the CPU or to the output of a priority decoder.
1	$\overline{INT}$	O	(Interrupt Request, Open Drain). Interrupt Request is an active-low output used to interrupt the CPU. It is driven LOW whenever the IP and CIE bits of the Status Register are set. It is cleared by UDC after receiving a clear IP command.
48	$\overline{INTACK}$	I	(Interrupt Acknowledge). Interrupt Acknowledge is an active-low input indicating that the request for interrupt has been granted. The UDC will place a vector onto the AD bus if the No Vector or Interrupt bit (MM3) is reset.
47	$\overline{RESET}$	I	(Reset). Reset is an active-low input to disable the UDC and clear its Master Mode register.
36, 37	$\overline{DREQ_1}$, $\overline{DREQ_2}$	I	(DMA Request). The DMA Request lines are two active-low inputs, one per channel. They may make transitions independent of the UDC clock and are used by external logic to initiate and control DMA operations performed by the UDC.
40, 39	$\overline{DACK_1}$, $\overline{DACK_2}$	O	(DMA Acknowledge). The DMA Acknowledge lines are active-low outputs, one per channel, which indicate that the channel is performing a DMA operation. $\overline{DACK}$ is pulsed, held active or held inactive during DMA operations as programmed in the Channel Mode register. For Flowthru operations, the peripheral is fully addressed using the conventional I/O addressing protocols and therefore may choose to ignore $\overline{DACK}$. $\overline{DACK}$ is always output as programmed in the Channel Mode register for a DMA operation, even when the operation is intiated by a CPU software request command or as a result of chaining. $\overline{DACK}$ is not output during the chaining operations.
38	$\overline{EOP}$	I/O	(End of Process). $\overline{EOP}$ is an active-low, open-drain, bidirectional signal. It must be pulled up with an external resistor of 1.8 kohm or more. The UDC emits an output pulse on $\overline{EOP}$ when a TC or MC termination occurs, as defined later. An external source may terminate a DMA operation in progress by driving $\overline{EOP}$ low. $\overline{EOP}$ always applies to the active channel; if no channel is active, $\overline{EOP}$ is ignored.
29 – 27 25 – 21	$A_{16} - A_{23}$	O	(Upper Address Bus, Three-state). The $A_{16} - A_{23}$ address lines are three-state outputs activated only when the UDC is controlling the system bus. Combined with the lower 16 address bits appearing on AD_0 through AD_{15} respectively, this 24-bit linear address allows the UDC to access anywhere within 16 megabytes of memory.

Note: All inputs to the UDC, except the clock, are directly TTL compatible.

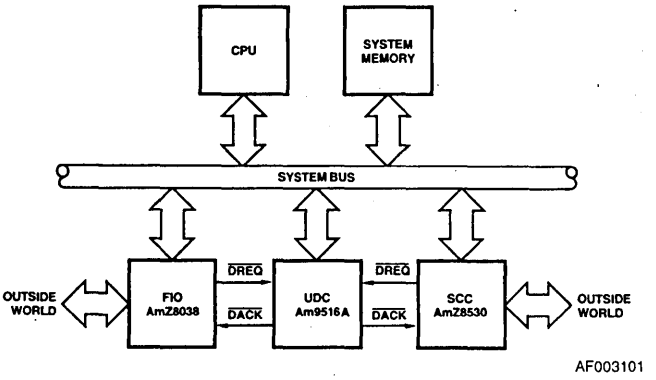


Figure 1. UDC Configurations

PRODUCT OVERVIEW

Register Description

The Am9516A UDC Block Diagram illustrates the internal registers. Figure 2 lists each register along with its size and read/write access restrictions. Registers which can be read by the CPU are either fast (F) or slow (S) readable. Fast registers can be read by a normal CPU I/O operation without additional wait states. Reading slow registers requires multiple wait states. Registers can be written to by the host CPU (W) and/or can be loaded by the DMA channel itself during chaining (C). All reads or writes must be word accesses since the UDC ignores the B/$\overline{W}$ line in slave mode. It is the responsibility of the user to supply the necessary external logic if slow readable registers are to be read.

The UDC registers can be categorized into chip-level registers, which control the overall operation and configuration of the UDC, and channel-level registers which are duplicated for each channel. The five chip-level registers are the Master Mode register the Command register, the Chain Control register, the Pointer register and the Temporary register. The Master Mode register selects the way the UDC chip interfaces to the system. The Command register is written to by the host CPU to initiate certain operations within the UDC chip, such as resetting the unit. The Chain Control register is used by a channel while it is reloading its channel-level registers from memory. The Pointer register is written to by the host CPU when the P/$\overline{D}$ input is HIGH. The data in Pointer register is the address of the internal register to be accessed. The Temporary register is used to hold data for Flowthru Transfer/Transfer-and-Searches.

The channel-level registers can be divided into two subcategories: general purpose registers, which would be found on most DMA chips, and special purpose registers, which provide additional features and functionality. The general purpose registers are the Base and Current Operation Count registers, the Base and Current Address registers A and B and the Channel Mode register. The special purpose registers are the Pattern and Mask registers, the Status register, the Interrupt Vector register, the Interrupt Save register and the Chain Address register.

The internal registers are read or written in two steps. First, the address of the register to be accessed is written to the Pointer register, when the P/$\overline{D}$ input is HIGH. Then, the data is read from or written into the desired register, which is indicated by the Pointer register, when P/$\overline{D}$ input is LOW. Note that a read with P/$\overline{D}$ HIGH causes the contents of the Pointer register to be read on AD_1 through AD_6.

Master Mode Register

The 4-bit Master Mode register, shown in Figure 3, controls the chip-level interfaces. It can be read from and written to by the host CPU without wait states through pins $AD_0 - AD_3$, but it is not loadable by chaining. On a reset, the Master Mode register is cleared to all zeroes. The function of each of the Master Mode bits is described in the following paragraphs.

The Chip Enable bit CE = 1 enables the UDC to request the bus. When enabled, the UDC can perform DMA Operations and reload registers. It can always issue interrupts and respond to interrupt acknowledges. When the Chip Enable bit is cleared, the UDC is inhibited from requesting control of the system bus and, therefore, inhibited from performing chaining or DMA operations.

The CPU Interleave bit enables interleaving between the CPU and the UDC.

The Wait Line Enable bit is used to enable sampling of the $\overline{WAIT}$ line during Memory and I/O transactions. Because the UDC provides the ability to insert software programmable wait states, many users may disable sampling of the $\overline{WAIT}$ pin to eliminate the logic driving this pin. The Wait Line Enable bit provides this flexibility. See the "Wait States" section of this document for details on wait state insertion.

The "No Vector on Interrupt" bit selects whether the UDC channel or a peripheral returns a vector during interrupt acknowledge cycles. When this bit is cleared, a channel receiving an interrupt acknowledge will drive the contents of its Interrupt Save register onto the $AD_0 - AD_{15}$ data bus while $\overline{INTACK}$ is LOW. If this bit is set, interrupts are serviced in an identical manner, but the $AD_0 - AD_{15}$ data bus remains in a high-impedance state throughout the acknowledge cycle.

Pointer Register

The Pointer register contains the address of the internal register to be accessed. It can be read from or written to by the CPU when the P/$\overline{D}$ line is HIGH.

Name	Size	Number	Access Type	Port Address CH-1/CH-2
Master Mode Register	4 bits	1	FW	38
Pointer Register	6 bits	1	FW	
Chain-Control Register	10 bits	1	C	
Temporary Register	16 bits	1	D	
Command Register	8 bits	1	W	2E/2C*
Current Address Register – A:				
Up-Addr/Tag field	14 bits	2	CFW	1A/18
Lower Address field	16 bits	2	CFW	0A/08
Current Address Register – B:				
Up-Addr/Tag field	14 bits	2	CFW	12/10
Lower Address field	16 bits	2	CFW	02/00
Base Address Register – A:				
Up-Addr/Tag field	14 bits	2	CFW	1E/1C
Lower Address field	16 bits	2	CFW	0E/0C
Base Address Register – B:				
Up-Addr/Tag field	14 bits	2	CFW	16/14
Lower Address field	16 bits	2	CFW	06/04
Current Operation Count	16 bits	2	CFW	32/30
Base Operation Count	16 bits	2	CFW	36/34
Pattern Register	16 bits	2	CSW	4A/48
Mask Register	16 bits	2	CSW	4E/4C
Status Register	16 bits	2	F	2E/2C
Interrupt Save Register	16 bits	2	F	2A/28
Interrupt Vector Register	8 bits	2	CSW	5A/58
Channel Mode Register – HIGH	5 bits	2	CS	56/54
Channel Mode Register – LOW	16 bits	2	CSW	52/50
Chain Address Register:				
Up-Addr/Tag field	10 bits	2	CFW	26/24
Lower Address field	16 bits	2	CFW	22/20

Access Codes: C = Chain Loadable
D = Accessible by UDC channel
F = Fast Readable
S = Slow Readable
W = Writable by CPU

Note: The address of the register to be accessed is stored in the Pointer register.

*Port addresses of the Command register can be used alternately for both channels except when issuing a "set or clear IP" command.

Figure 2. UDC Internal Register

Chain Control Register

When a channel starts a chaining operation, it fetches a Reload word from the memory location pointed to by the Chain Address register (Figure 10). This word is then stored in the Chain Control register. The Chain Control register cannot be written to or read from by the CPU. Once a channel starts a chain operation, the channel will not relinquish bus control until all registers specified in the Reload word are reloaded unless an $\overline{EOP}$ signal is issued to the chip. Issuing an $\overline{EOP}$ to a channel during chaining will prevent the chain operation from resuming and the contents of the Reload Word register can be discarded.

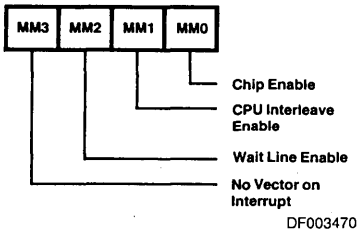


Figure 3. Master Mode Register

Temporary Register

The Temporary register is used to stage data during Flowthru transfers and to hold data being compared during a Search or a Transfer-and-Search. The temporary register cannot be written to or read from by the CPU. In byte-word funneling, data may be loaded into or from the Temporary register on a byte-by-byte basis, with bytes sometimes moving between the low byte of the data bus and the high byte of the Temporary register or vice-versa. See the "Transfer" section for details.

Command Register

The UDC Command register (Figure 19) is an 8-bit write-only register written to by the host CPU. The Command register is loaded from the data on $AD_7 - AD_0$; the data on $AD_{15} - AD_8$ is disregarded. A complete discussion of the commands is given in the "Command Descriptions" section.

Current and Base Address Registers A and B

The Current Address registers A and B (Current ARA and ARB) are used to point to the source and destination addresses for DMA operations. The contents of the Base ARA and ARB registers are loaded into the Current ARA and ARB registers at the end of a DMA operation if the user enables Base-to-Current reloading in the Completion Field of the Channel Mode register. This facilitates DMA operations without reloading of the Current registers. The ARA and ARB registers can be loaded during chaining, can be written to by the host CPU without wait states and can be read by the CPU.

Each of the Base and Current ARA and ARB registers consists of two words organized as a 6-bit Tag Field and an 8-bit Upper Address in one word and a 16-bit Lower Address in the other. See Figure 5. The Tag Field selects whether the address is to be incremented, decremented or left unchanged, and the status codes associated with the address. The Tag field also allows the user to insert 0, 1, 2 or 4 wait states into memory or I/O accesses addressed by the offset and segment fields.

The Address Reference Select Field in the Tag field selects whether the address pertains to memory space or I/O space. Note that the N/$\overline{S}$ output pin may be either HIGH (indicating Normal) or LOW (indicating System) for space. At the end of each iteration of a DMA Operation, the user may select to leave the address unchanged or to increment it or to decrement it. I/O addresses, if changed, are always incremented/decremented by 2. Memory addresses are changed by 1 if the address points to a byte operand (as programmed in the Channel Mode register's Operation field) and by 2 if the address points to a word operand. Note that, if an I/O or memory address is used to point to a word operand, the address must be even to avoid unpredictable results. An address used to point to a byte operand may be even or odd. Since memory byte operand addresses will increment/decrement by 1, they will toggle between even and odd values. Since I/O byte operand addresses will increment/decrement by 2, once programmed to an even or odd value, they will remain even or odd, allowing consecutive I/O operations to access the same half of the data bus. High bus is for even address and low bus for odd.

Current and Base Operation Count Registers

Both the Current and Base Operation Count registers may be loaded during chaining and may be written to and read from by the host CPU.

The 16-bit Current Operation Count register is used to specify the number of words or bytes to be transferred, searched or transferred-and-searched. For word-to-word operations and byte-word funneling, the Current Operation Count register must be programmed with the number of words to be transferred or searched.

Each time a datum is transferred or searched, the Operation Count register is decremented by 1. Once all of the data is transferred or searched, the transfer or search operation will stop, the Current Operation Count register will contain all zeroes, and the TC bit in Status Register will be "1." If the transfer or search stops before the Current Operation Count register reaches 0, the contents of the register will indicate the number of bytes or words remaining to be transferred or searched. This allows a channel which had been stopped prematurely to be restarted where it left off without requiring reloading of the Current Operation Count register.

For the byte-to-byte operations, the Current Operation Count register should specify the number of bytes to be transferred or searched. The maximum number of bytes which can be specified is 64K bytes by setting the Current Operation Count register to 0000.

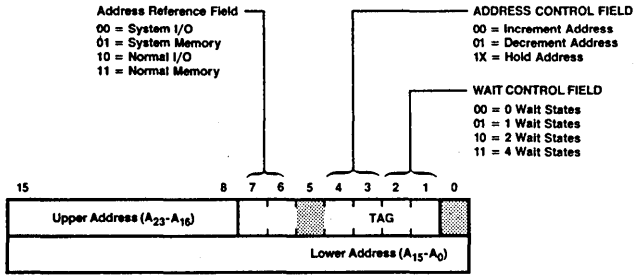


Figure 4. Address Registers A and B

Pattern and Mask Registers

The 16-bit Pattern and Mask registers are used in Search and Transfer-and-Search operations. Both the Pattern and Mask registers may be loaded by chaining, may be written to by the host CPU and may be read from by the host CPU, provided wait states are inserted, since these registers are slow readable. The Pattern register contains the pattern that the read data is compared to. Setting a Mask register bit to "1" specifies that the bit always matches. See the "Search" and "Transfer-and-Search" sections for further details.

Status Register

The two 16-bit Status registers, depicted in Figure 5, are read-only registers which can be read by the CPU without wait states. Each of these registers reports on the status of its associated channel.

The Interrupt Status Field in the Status register contains the Channel Interrupt Enable (CIE) and Interrupt Pending (IP) bits. These bits are described in detail in the "Interrupt" section of this document.

The UDC status field contains the current channel status. The "channel initialized and waiting for request" status is not explicitly stated — it is reflected by Status register bits ST_{12} through ST_9 being all zero. The "Waiting for Bus" (WFB) status will cause bit ST_{10} to be set and indicates that the channel wants bus control to perform a DMA operation. The channel may or may not actually be asserting BREQ HIGH, depending on the programming of the Master Mode Chip Enable bit (MMO) when the channel decided it wanted the bus. See the "Bus Request/Grant" section for details. If a channel completes a DMA operation and neither Base-to-Current reloading nor auto-chaining were enabled, the No Auto-Reload or Chaining (NAC) bit will be set. The NAC bit will be reset when the channel receives a "Start Chain Command." If two interrupts are queued, the Second Interrupt Pending bit (SIP) will be set and the channel will be inhibited from further activity until an interrupt acknowledge occurs. See the "Interrupt" section for details. Finally, if the channel is issued an $\overline{EOP}$ during chaining, the Chaining Abort (CA) and the NAC will be set. These bits are also set when a "reset" is issued to the UDC. The CA bit holds the NAC bit in the set state. The CA bit is cleared when a new Chain Upper Address and Tag word or Lower Address word is loaded into the channel.

The Hardware Interface Field provides a Hardware Request (HRQ) bit which provides a means of monitoring the channels $\overline{DREQ}$ input pin. When the $\overline{DREQ}$ pin is LOW, the HRQ bit will be "1" and vice-versa. The Hardware Mask (HM) bit, when set, prevents the UDC from responding to a LOW on $\overline{DREQ}$. Note, however, that the Hardware Request bit always reports the true (unmasked) status of $\overline{DREQ}$ regardless of the setting of the HM bit.

The Completion Field stores data at the end of each DMA operation. This data indicates why the DMA operation ended. When the next DMA operation ends, new data is loaded into these bits overwriting, thereby erasing the old setting. Three bits indicate whether the DMA operation ended as a result of a TC, MC or $\overline{EOP}$ termination. The TC bit will be "1" if the Operation Count reaching zero ended the DMA operation. The MC bit will be "1" if an MC termination occurred regardless of whether Stop-on-Match or Stop-on-no-Match was selected. The EOP bit is set only when an external $\overline{EOP}$ ends a DMA transfer; it is not set for $\overline{EOP}$ issued during chaining. Note that two or even all three of MC, TC and EOP may be set if multiple reasons exist for ending the DMA operation. The MCH and MCL bits report on the match state of the upper and lower comparator bytes, respectively. These bits are set when the associated comparator byte has a match and are reset otherwise, regardless of whether Stop-on-Match or Stop-on-no-Match is programmed. Regardless of the DMA operation performed, these bits will reflect the comparator status at the end of the DMA operation. These two bits are provided to help determine which byte matched or did not match when using 8-bit matches with word searches and transfer-and-searches. The three reserved bits return zeroes during reads.

Interrupt Vector and Interrupt Save Registers

Each channel has an Interrupt Vector register and an Interrupt Save register. The Interrupt Vector is 8-bit wide and is written to and read from on $AD_0 - AD_7$. The Interrupt Save register may be read by the CPU without wait states. The Interrupt Vector register contains the vector or identifier to be output during an Interrupt Acknowledge cycle. When an interrupt occurs (IP = 1), either because a DMA operation terminated or because $\overline{EOP}$ was driven LOW during chaining, the contents of the Interrupt Vector register and part of the Channel Status register are stored in the 16-bit Interrupt Save register (see Figure 6).

Because the vector and status are stored, a new vector can be loaded into the Interrupt Vector register during chaining, and a new DMA operation can be performed before an interrupt acknowledge cycle occurs. If another interrupt occurs on the channel before the first is acknowledged, further channel activity is suspended.

As soon as the first clear IP command is issued, the status and vector for the second interrupt are loaded into the Interrupt Save register and Channel Operation resumes. The UDC can retain only two interrupts for each channel; a third operation cannot be initiated until the first interrupt has been cleared. See the "Interrupt" section for further details.

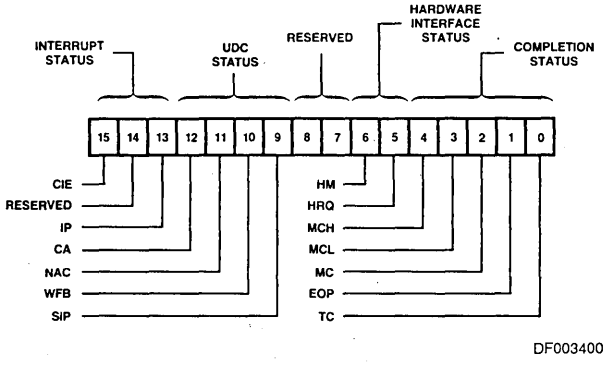


Figure 5. Status Register

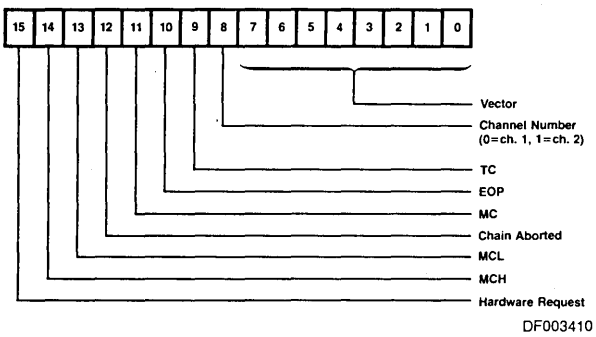


Figure 6. Interrupt Save Register

Channel Mode Register

The Channel Mode registers are two words wide. There are 21 bits defined in each Channel Mode register; the other 11 bits are unused. See Figure 7. The Channel Mode registers may be loaded during chaining and may be read by the host CPU. CPU reads of the Channel Mode register are slow reads and require insertion of multiple wait states. The Channel Mode Low word (bits 0–15) may be written to directly by the host CPU. The Channel Mode register selects what type of DMA operation the channel is to perform, how the operation is to be executed, and what action, if any, is to be taken when the channel finishes.

The Data Operation Field and the Transfer Type field select the type of operation the channel is to perform. It also selects the operand size of bytes or words (see Figure 8 for code-definition). The different types of operations are described in detail in the "DMA Operations" section. The Flip bit is used to select whether the Current ARA register points to the source and the Current ARB register points to the destination or vice-versa.

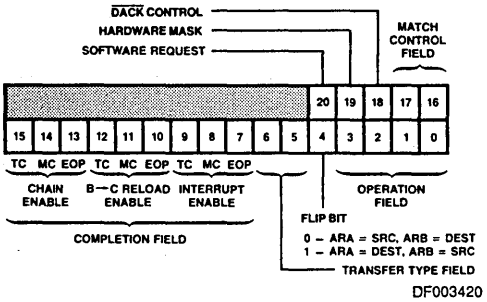


Figure 7. Channel Mode Register

The Completion Field is used to program the action taken by the channel at the end of a DMA operation. This field is discussed in the "Completion Options" section. The 2-bit Match Control field selects whether matches use an 8-bit or 16-bit pattern and whether the channel is to stop-on-match or stop-on-no-match. See Figure 8 and the "Search" section for details. The Software Request bit and Hardware Mask bit can be set and cleared by software command in addition to being loaded in parallel with other Channel Mode bits. These bits are described in detail in the "Initiating DMA Operations" section.

The $\overline{\text{DACK}}$ Control bit is used to specify when the $\overline{\text{DACK}}$ pin is driven active. When this bit is cleared, the channel's $\overline{\text{DACK}}$ pin will be active whenever the channel is performing a DMA Operation, regardless of the type of transaction. Note that the pin will not be active while the channel is chaining. If this bit is set, the $\overline{\text{DACK}}$ pin will be inactive during chaining, during both Flowthru Transfers and Flowthru Transfer-and-Searches, and during Searches, but $\overline{\text{DACK}}$ will be pulsed active during Flyby Transfers and Flyby Transfers-and-Searches at the time necessary to strobe data into or out of the Flyby peripheral. Flyby operations are discussed in detail in the "Flyby Transactions" section.

DATA OPERATION FIELD			
	Operand Size		
Code/Operation	ARA	ARB	Transaction Type
Transfer			
0001	Byte	Byte	Flowthru
100X	Byte	Word	Flowthru
0000	Word	Word	Flowthru
0011	Byte	Byte	Flyby
0010	Word	Word	Flyby
Transfer-and-Search			
0101	Byte	Byte	Flowthru
110X	Byte	Word	Flowthru
0100	Word	Word	Flowthru
0111	Byte	Byte	Flyby
0110	Word	Word	Flyby
Search			
1111	Byte	Byte	N/A
1110	Word	Word	N/A
101X	Illegal		

TRANSFER TYPE FIELD AND MATCH CONTROL FIELD		
Transfer Type	Code	Match Control
Single Transfer	00	Stop on No Match
Demand Dedicated/Bus Hold	01	Stop on No Match
Demand Dedicated/Bus Release	10	Stop on Word Match
Demand Interleave	11	Stop on Byte Match

Figure 8. Channel Mode Coding

Chain Address Register

Each channel has a Chain Address register which points to the chain control table in memory containing data to be loaded into the channel's registers. The Chain Address register, as shown in Figure 9, is two words long. The first word consists of an Upper Address and Tag field. The second word contains the 16-bit Lower Address portion of the memory address. The Tag field contains 2 bits used to designate the number of wait states to be inserted during accesses to the Chain Control Table.

The Chain Address register may be loaded during chaining and may be read from and written to by the host CPU without wait states. If an EOP is issued to the UDC during chaining, the Chain Address register holds the old address. This is true even if the access failure occurred while new Chain Address data was being loaded, since the old data is restored unless both words of the new data are successfully read. Note, however, that EOPs that occur when chaining and while loading a new Chain Address cause the new data to be lost.

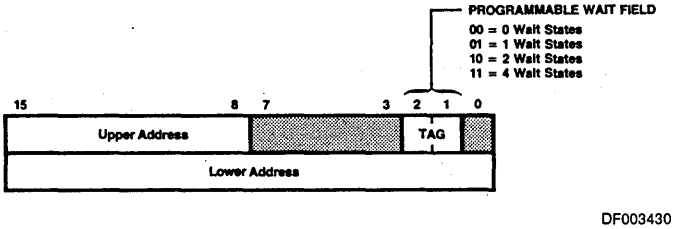


Figure 9. Chain Address Register

DETAILED DESCRIPTION

Any given DMA operation, be it a Transfer, a Search, or a Transfer-and-Search operation, consists of three phases. In the first phase, the channel's registers are initialized to specify and control the desired DMA operation. In the second phase, the DMA operation itself is started and performed. The final phase involves terminating the DMA operation and performing any actions selected to occur on termination. Each of these different phases is described in detail in the following sections.

Reset

The UDC can be reset either by hardware or software. The software reset command is described in the "Commands" section. Hardware resets are applied by pulling $\overline{\text{RESET}}$ LOW. The UDC may be in control of the bus when a reset is applied. BACK is removed internally causing the outputs to go tri-state. If BACK remains HIGH after reset, the UDC will not drive the bus unless BREQ is active. As soon as BACK goes inactive, the UDC places the $AD_0 - AD_{15}$, $A_{16} - AD_{23}$, $R/\overline{W}$, $\overline{DS}$, $N/\overline{S}$, $M/\overline{IO}$ $B/\overline{W}$, $\overline{TBEN}$ and $\overline{RBEN}$ signals in the high-impedance state.

Both software and hardware resets clear the Master Mode register, clear the CIE, IP and SIP bits, and set the CA and NAC bits in each Channel's Status register. The contents of all other UDC registers will be unchanged for a software reset. Since a hardware reset may have been applied partway through a DMA operation being performed by a UDC channel, the channel's registers should be assumed to contain indeterminate data following a hardware reset.

The Master Mode register contains all zeroes after a reset. The UDC is disabled, and the CPU interleave and hardware wait are inhibited.

Because the CA and NAC bits in the Status register are set by a reset, the channel will be prevented from starting a DMA operation until its Chain Address register's Segment, Tag and Offset fields are programmed and the channel is issued a "Start Chain" Command.

Channel Initialization

The philosophy behind the Am9516A UDC design is that the UDC should be able to operate with a minimum of interaction with the host CPU. This goal is achieved by having the UDC load its own control parameters from memory into each channel. The CPU has to program only the Master Mode register and each Channel's Chain Address register. All other registers are loaded by the channels themselves from a table located in the System memory space and pointed to by the Chain Address register. This reloading operation is called chaining, and the table is called the Chain Control Table.

The Upper and Lower Address fields of the Chain Address register form a 24-bit address which points to a location in system memory space. Chaining is performed by repetitively reading words from memory. Note that the Chain Address register should always be loaded with an even Address; loading an odd Address will cause unpredictable results. The 2-bit Tag field facilitates interfacing to slow memory by allowing the user to select 0, 1, 2 or 4 programmable wait states. The UDC will automatically insert the programmed number of wait states in each memory access during chaining.

The Chain Address register points to the first word in the Chain Control Table. This word is called the Reload Word. See Figure 10. The purpose of the Reload Word is to specify which registers in the channel are to be reloaded. Reload Word bits 10 – 15 are undefined and may be 0 or 1. Each of bits 0 through 9 in the Reload Word correspond to either one or two registers in the channel (see Figure 11). When a Reload Word bit is "1," it means that the register or registers corresponding to that bit are to be reloaded. If a Reload Word bit is "0," the register or registers corresponding to that bit are not to be reloaded. The data to be loaded into the selected register(s) follow the Reload Word in memory (i.e., the data are stored at successively larger memory addresses). The Chain Control Table is a variable length table. Only the data to be loaded are in the table, and the data are packed together.

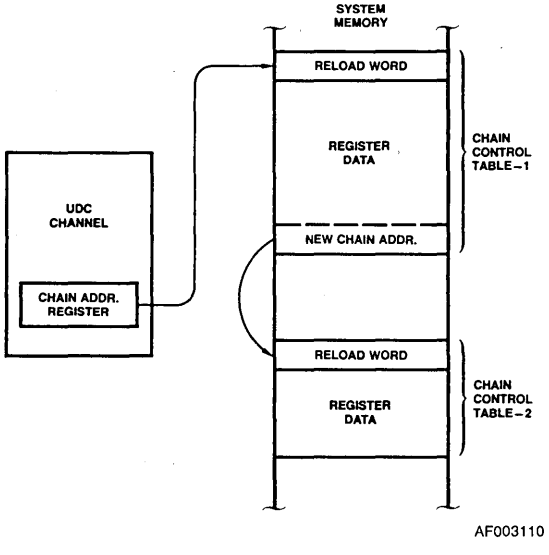


Figure 10. Chaining and Chain Control Tables

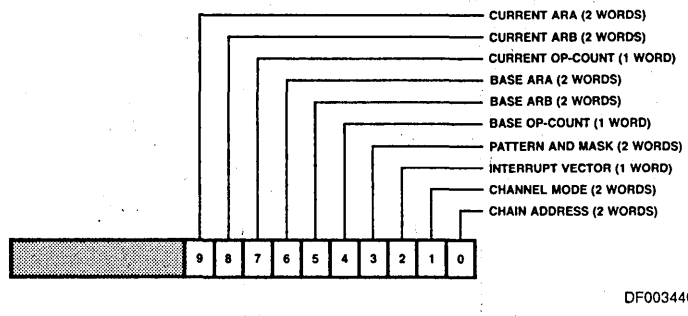


Figure 11. Reload Word/Chain Control Register

When the channel is to reload itself, it first uses the Chain Address register contents to load the Reload Word into the UDC's Chain Control register. Next, the Chain Address register contents are incremented by two to point to the next word in memory. The channel than scans the Reload Word register from bit 9 down to bit 0 to see which registers are to be reloaded. If no registers are specified (bits 9 – 0 are all 0), no registers will be reloaded. If at least one of bits 9 – 0 is set to "1," the register(s) corresponding to the set bit are reloaded, the bit is cleared and the Chain Address register is incremented by 2. The channel continues this operation of scanning the bits from the most significant to least significant bit position, clearing each set bit after reloading its associated registers and incrementing the Chain Address register by 2. If all of bits 9 to 0 are set, all the registers will be reloaded in the order: Current ARA, Current ARB, Current Operation Count,. . . Channel Mode and Chain Address. Figure 12 shows two examples of Chain Control Tables. Example 1 shows the ordering of data when all registers are to be reloaded. In example 2 only some registers are reloaded. Once the channel is reloaded, it is ready to perform a DMA operation. Note when loading address registers, the Upper Address and Tag word are loaded first, then the Lower Address word. Also, the Pattern register is loaded before the Mask register.

Initiating DMA Operations

DMA operations can be initiated in one of three ways — by software request, by hardware request and by loading a set software request bit into the Channel Mode register during Chaining.

Starting After Chaining

If the software request bit of the Channel Mode register is loaded with a "1" during chaining, the channel will perform the programmed DMA operation at the end of chaining. If the channel is programmed for Single Operation or Demand, it will perform the operation immediately. The channel will give up the bus after chaining and before the operation if the CPU interleave bit in the Master Mode register is set. See the "Channel Response" section for details. Note that once a channel starts a chaining operation by fetching a Reload Word, it retains bus control at least until chaining of the last register's data is performed.

Software Requests

The CPU can issue Software Request commands to start DMA operations on a channel. This will cause the channel to request the bus and perform transfers. See the description of the software request command for details.

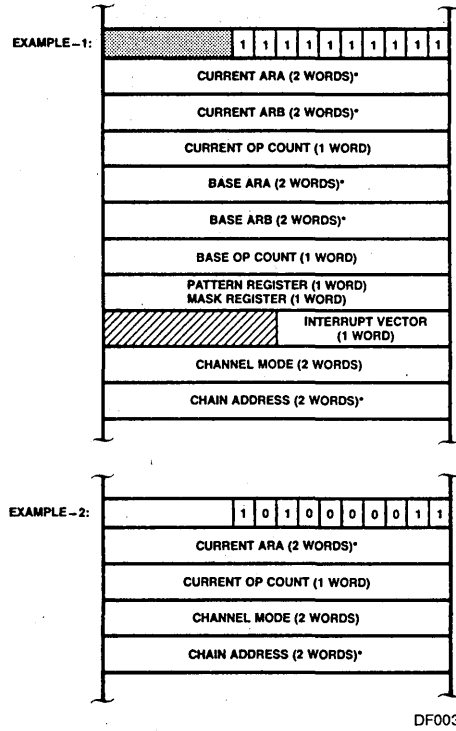


Figure 12. Examples of Chain Control Table

*Load the Upper Address and Tag Word first, then the Lower Address Word.

Hardware Requests

DMA operations will often be started by applying a LOW on the channel's $\overline{DREQ}$ input. The "Channel Response" section describes when the LOW $\overline{DREQ}$ signals are sampled and when the $\overline{DREQ}$ requests can be applied to start the next DMA operation after chaining (see Timing Diagrams 1 and 2).

Bus Request/Grant

Before the UDC can perform a DMA Operation, it must gain control of the system bus. The BREQ and BACK interface pins provide connections between the UDC and the host CPU and other devices, if present, to arbitrate which device has control of the system bus. When the UDC wants to gain bus control, it drives BREQ HIGH.

Some period of time after the UDC drives BREQ HIGH, the CPU will relinquish bus control and drive its HLDA signal HIGH. When the UDC's BACK input goes HIGH, it may begin performing operations on the system bus. When the UDC finishes its operation, it stops driving BREQ HIGH.

When more than one device is used, a priority encoder and a priority decoder are used to decide the bus grant priority.

DMA Operations

There are three types of DMA operations: Transfer, Search, and Transfer-and-Search. Transfers move data from a source location to a destination location. Two types of transfers are provided: Flowthru and Flyby. Searches read data from a source and compare the read data to the contents of the Pattern register. A Mask register allows the user to declare "don't care" bits.

The user can program that the search is to stop either when the read data matches the masked pattern or when the read data fails to match the masked pattern. This capability is called Stop-on-Match and Stop-on-no-Match. Transfer-and-Search combines the two functions to facilitate the transferring of variable length data blocks. Like Transfer, Transfer-and-Search can be performed in either Flowthru or Flyby mode.

Transfers

Transfers use four of the Channel registers to control the transfer operation: the Current ARA and ARB registers; the Current Operation Count register; and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select whether ARA is to point to the source and ARB is to point to the destination or vice-versa. The Current Operation Count register specifies the number of words or bytes to be transferred.

Bits $CM_3 - M_0$ in the Channel Mode register program whether a Flowthru or Flyby transfer is to be performed. Flowthru transfers are performed in either two or three steps. First, the channel outputs the address of the source and reads the source data into the UDC's Temporary register. In two-step Flowthru Transfer, the channel will then address the destination and write the Temporary register data to the destination location. The three-step Flowthru operation (i.e., the byte-word funneling) is described later in this section. The source and destination for Flowthru Transfers can both be memory locations or both peripheral devices, or one may be a memory location and the other a peripheral device. The $\overline{DACK}$ output for the transferring channel may be programmed to be inactive throughout the transfer or active during the transfer. This is controlled by bit CM_{18} in the Channel Mode register.

Flyby transfers provide improved transfer throughput over Flowthru but are restricted to transfers between memory and peripherals or between two peripherals. Flyby operations are described in detail in the "Flyby Transactions" section.

Transfers can use both byte- and word-sized data. Flowthru byte-to-byte transfers are performed by reading a byte from the source and writing a byte to the destination. The Current Operation Count register must be loaded with the number of bytes to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±1 if the register points to a memory space ($TG_6 = 2$) and by ±2 if the register points to an I/O space ($TG_6 = 0$).

Flowthru word-to-word transfers require that the Current Operation Count specify the number of words to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±2 regardless of whether the register points to memory or I/O space.

Byte-word funneling provides packing and unpacking of byte data to facilitate high speed transfers between byte and word peripherals and/or memory. This funneling option can only be used in Flowthru mode. Funneled Flowthru transfers are performed in three steps. For transfers from a byte source to a word destination, two consecutive byte reads are performed from the source address. The data read is assembled into the UDC's Temporary register. In the third step, the Temporary register data is written to the destination address in a word transfer. Funneled transfers from a word source to a byte destination are performed by first loading a word from the source into the UDC's Temporary register. The word is then written out to the destination in two byte writes. For funnel operations, the byte-oriented address must be in the Current ARA register, and the word-oriented address must be in the Current ARB register. The Flip bit (CM_4) in the Channel Mode register is used to specify which address is the source and which is the destination. When the byte address is to be incremented or decremented, the increment/decrement operation occurs after each of the two reads or writes. The Current Operation Count Register must be loaded with the number of words to be transferred.

In byte-to-word funneling operations, it is necessary to specify which half of the Temporary register (upper or lower byte) is loaded with the first byte of data. Similarly, for word-to-byte funneling operations, it is necessary to define which half of the Temporary register is written out first. Figure 13 summarizes these characteristics for both byte-to-word and word-to-byte funneling operations. The criterion used to determine the packing/unpacking order is based on whether the Current ARB register is programmed for incrementing or decrementing of the address. Note that if the address is to remain unchanged (i.e., if bit TG_4 in the Tag Field of the Current ARB register is 1), the increment/decrement bit (bit TG_3) still specifies the packing order.

Search

Searches use five of the Channel registers to control the operation: either the Current ARA or ARB, the Operation Count, the Pattern and Mask registers, and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select either Current ARA or ARB as the register specifying the source for the search. Only one of the Current Address registers is used for search operations since there is no destination address required. The Current Operation Count register specifies the maximum number of words or bytes to be searched.

Search operations involve repetitive reads from the peripheral or memory until the specified match condition is met. The search then stops. This is called a Match Condition or MC termination. Each time a read is performed, the Source address, if so programmed, is incremented or decremented and the Operation Count is decremented by 1. If the match condition has not been met by the time the Operation Count reaches zero, the zero value will force a TC termination, ending the search. Searches can also stop due to a LOW being applied to the $\overline{EOP}$ interface pin. During a Search operation, the channel's $\overline{DACK}$ output will be either inactive or active throughout the search. This is controlled by bit CM_{18} in the Channel Mode register. The reads from the peripheral or

memory performed during Search follow the timing sequences described in the "Flowthru Transactions" sections.

On each read during a Search operation, the UDC's Temporary register is loaded with data and compared to the Pattern register. The user can select that the Search is to stop when the Pattern and Temporary register contents match or when they don't match. This Stop-On-Match/Stop-On-No-Match feature is programmed in bit CM_{17} of the Channel Mode register. CM_2 is an enable for the output of the comparator and allows the MC signal to be generated. A Mask register allows the user to exclude or mask selected Temporary register bits from the comparison by setting the corresponding Mask register bit to "1." The masked bits are defined to always match. Thus, in Stop-On-Match, successful matching of the unmasked bits, in conjunction with the always-matched masked bits, will cause the search to stop. For Stop-On-No-Match, the always-matched masked bits are by definition excluded from not matching and therefore excluded from stopping the search.

For word reads the user may select either 8-bit or 16-bit compares through the Channel Mode register bit CM_{16}. In an 8-bit, Stop-On-Match, word-read operation, successful matching of either the upper or lower byte of unmasked Pattern and Temporary registers bits will stop the search. Both bytes do not have to match. In 16-bit Stop-On-Match with word reads, all unmasked Pattern and Temporary register bits must match to stop the search. In an 8-bit or 16-bit, Stop-On-No-Match, word-read Search operation, failure of any bit to match will terminate the Search operation.

In an 8-bit Stop-On-Match with byte-reads, the Search will stop if either the upper or lower byte of unmasked Pattern and Temporary register bits match. For an 8-bit Stop-On-No-Match with byte reads, failure of matching in any unmasked Pattern and Temporary register bit will cause the Search to stop. For 8-bit searches, the upper and lower bytes of the Pattern and Mask register should usually be programmed with the same data. Failure to set the upper and lower bytes of the Pattern and Mask registers to identical values will result in different comparison criteria being used for the upper and lower bytes of the Temporary register. Users failing to program identical values for the upper and lower bytes can predict the results by recognizing that in 8-bit Stop-On-Match, the search will end if all the unmasked bits in either the upper or lower bytes match, and for 8-bit Stop-On-No-Match, the failure of any unmasked bit to match will end the Search. For accurate predictions, it is also necessary to know that for word reads the Temporary register high and low bytes are loaded from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. In byte reads, the read byte is duplicated in both halves of the Temporary register except in funneling.

Funneling Direction	Current ARB Tag Field		Increment/Decrement and Packing/Unpacking Rules
	TG_4	TG_3	
Word-to-Byte (CM_4 = 1)	0	0	Increment ARB, Write High Byte First
	0	1	Decrement ARB, Write Low Byte First
	1	0	Hold ARB, Write High Byte First
	1	1	Hold ARB, Write Low Byte First
Byte-to-Word (CM_4 = 0)	0	0	Increment ARB, Read High Half of Word First
	0	1	Decrement ARB, Read Low Half of Word First
	1	0	Hold ARB, Read High Half of Word First
	1	1	Hold ARB, Read Low Half of Word First

Figure 13. Byte/Word Funneling

Transfer-and-Search

Transfer-and-Search combines the operations of the Transfer and the Search functions. The registers used to control Transfer-and-Searches are the Current ARA and ARB registers, the Operation Count register, the Pattern and Mask registers, and the Channel Mode register.

A Transfer-and-Search operation will end when the data transferred meets the match condition specified in Channel Mode register bits $CM_{17} - CM_{16}$. The Mask and Pattern registers indicate those bits being compared with the Temporary register contents. Like Transfers and Searches, Transfers-and-Searches will also be terminated if the operation count goes to zero or if a LOW is applied to the $\overline{EOP}$ pin. Regardless of whether Transfer-and-Search stops because of a TC, MC or EOP, it will always complete the iteration by writing to the destination address before ending (writing twice for word-to-byte funneling).

In Flowthru mode, the Transfer-and-Search timing is identical to Flowthru Transfer. While the data is in the Temporary register, it is masked by the Mask register and compared to the Pattern register. For word Transfer and Transfer-and-Search, the high and low bytes of the Temporary register are always written to and read from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. For byte Transfer and Transfer-and-Search, the byte read is always loaded into both halves of the Temporary register, and the entire register is driven directly out onto the $AD_{15} - AD_0$ bus. Transfer-and-Search can also be used with byte word funneling. In funneling, the match is an 8-bit match or 16-bit match as determined by the setting of bit CM_{16}.

Flyby Transfer-and-Search can be used to increase throughput for transfer between two peripherals or between memory and a peripheral. Memory-to-Memory Flyby is not supported. Also, in Flyby, the operand sizes of the source and destination must be the same, funneling is not supported. A complete discussion of Flyby timing is given the "Flyby Transactions" section. During a Flyby Transfer-and-Search, data is loaded into the Temporary register to facilitate the comparison operation, and at the same time, data is transferred from the source to the destination. When byte operands are used, data is loaded into both bytes of the Temporary register, from the $AD_{15} - AD_8$ bus if the Current ARA register is even and from $AD_7 - AD_0$ line if the Current ARA register is odd. This will alternate for memory bytes so the user must drive both halves of the bus to use the search. When word operands are used, data is loaded directly from $AD_{15} - AD_8$ and $AD_7 - AD_0$ into the Temporary register's high and low bytes respectively.

Channel Response

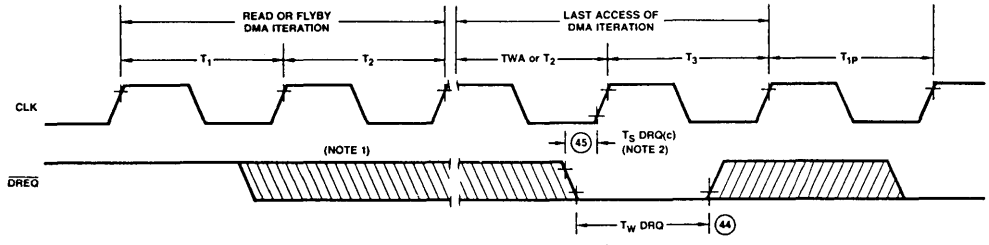
Channel Mode register bits $CM_6 - CM_5$ select the channel's response to the request to start a DMA operation. The response falls into either of two types: Single Operation or Demand. There are three subtypes for Demand operations: Demand Dedicated with Bus Hold, Demand Dedicated with Bus Release, and Demand Interleave. To make discussions clear, it is necessary to define the term "single iteration of a

DMA operation." For Search operations, one iteration consists of a single read operation and a comparison of the read data to the unmasked Pattern register bits. The Operation count will be decremented by 1, and the Current Address register used incremented or decremented if so programmed. For Transfer and Transfer-and-Search operations, a single iteration comprises reading a datum from the source, writing it to the destination, comparing the read datum to the unmasked Pattern register bits (Transfer-and-Search only), decrementing the Operation count by 1 and incrementing/decrementing the Current ARA and ARB registers if so programmed. In byte-word funneling, a single iteration consists of two reads followed by a write (Byte-to-Word funneling) or one read followed by two writes (Word-to-Byte funneling). In all Transfer and Transfer-and-Search cases, the iteration will not stop until the data in the Temporary register is written to the destination. See Appendix B for flowchart.

Single Operation

The Single Operation response is intended for use with peripherals which transfer single bytes or words at irregular intervals. Each application of a Software request command will cause the channel to perform a single iteration of the DMA operation. Similarly, if the Software request bit is set by chaining, at the end of chaining the channel will perform a single iteration of the DMA operation. Each application of a HIGH-to-LOW transition on the $\overline{DREQ}$ input will also cause a single iteration of the DMA operation. If the Hardware mask bit is set when the transition is made, the iteration will be performed when the mask is cleared, providing the DMA operation has not terminated. See the Set/Clear Hardware mask bit command for details. Each time a Single Operation ends, the channel will give up control of the bus unless a new transition has occurred on $\overline{DREQ}$. The new transition can occur anytime after the HIGH-to-LOW ALE transition of a read or Flyby memory or I/O access of the DMA iteration. Timing Diagram 1 shows the times after which a new transition can be applied and recognized to avoid giving up the us at the end of the current iteration.

*TIMING DIAGRAM 1. Sampling $\overline{DREQ}$ During Single Transfer DMA Operations



WF007460

Notes: 1. HIGH-to-LOW $\overline{DREQ}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of a read or flyby DMA iteration.

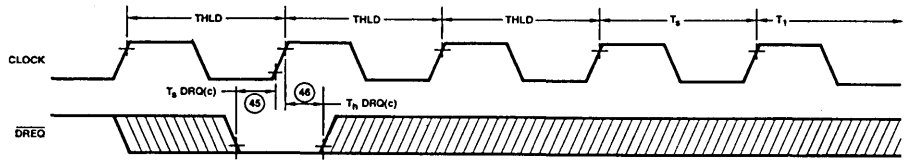
2. A HIGH-to-LOW $\overline{DREQ}$ transition must meet the conditions in Note 1 and must occur TsDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{DREQ}$ may go HIGH before TsDRQ(c) if it has met the TwDRQ parameter.

3. Flyby and Search transactions have only a single access; parameter TsDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

*See Appendix D for timing parameters.

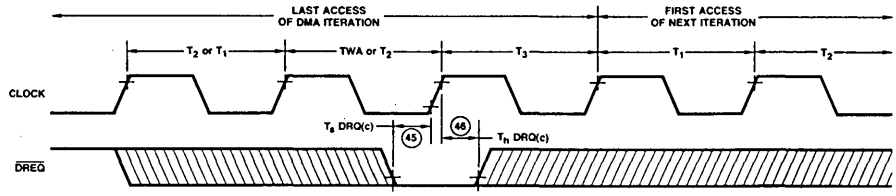
TIMING DIAGRAM 2. $\overline{DREQ}$ Sampling in Demand Mode

(a) Sampling of $\overline{DREQ}$ while in Bus Hold Mode



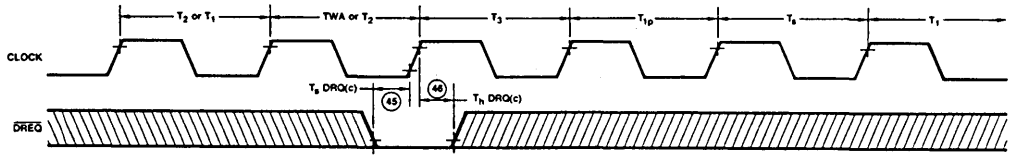
WF007480

(b) $\overline{DREQ}$ Sampling in Demand Mode During DMA Operations



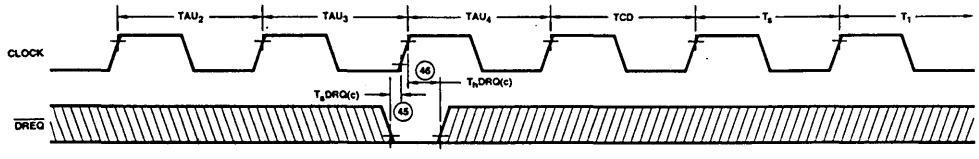
WF007490

(c) Sampling $\overline{DREQ}$ at the End of Chaining



WF007500

(d) Sampling $\overline{DREQ}$ at the End of Base-to-Current Reloading



WF007510

Notes: 1. $\overline{DREQ}$ must be LOW from the start of TsDRQ(c) to the end of ThDREQ(c) to ensure that the request is recognized.
2. Failure to meet this setup time will result in the channel releasing the bus.
3. Ts is a setup state, generated before entering DMA operation cycle.
4. TAU_2, TAU_3 and TAU_4 are auto-reload states, followed by TCD (chain decision) state.

Demand Dedicated With Bus Hold

In Demand Dedicated with Bus Hold (abbreviated Bus Hold), the application of a Software request command or the setting of the software request bit during chaining or applying a LOW level on the $\overline{DREQ}$ input will cause the channel to acquire bus control.

If $\overline{DACK}$ is programmed as a level output ($CM_{18} = 0$), $\overline{DACK}$ will be active from when the channel acquires bus control to when it relinquishes control. A Software Request will cause the channel to request the bus and perform the DMA operations until TC, MC or EOP.

Once the channel gains bus control due to a LOW $\overline{DREQ}$ level, it samples $\overline{DREQ}$ as shown in Timing Diagram 2. If $\overline{DREQ}$ is LOW, an iteration of the DMA operation is performed. If $\overline{DREQ}$ is HIGH, the channel retains bus control and continues to drive all bus control signals active or inactive, but performs no DMA operation. Thus the user can start or stop execution of DMA operations by modulating $\overline{DREQ}$. Once TC, MC or EOP occurs, the channel will either release the bus or, if chaining or Base-to-Current reloading is to occur, perform the desired operation. After chaining or Base-to-Current reloading, if the channel is still in Bus Hold mode and does not have a set software request bit (set either by chaining or command), the channel will relinquish bus control unless a LOW $\overline{DREQ}$ level occurs within the time limits.

Demand Dedicated With Bus Release

In Demand Dedicated with Bus Release (abbreviated Bus Release), the application of a Software Request will cause the channel to request the bus and perform the programmed DMA operation until TC, MC or EOP. If the channel was programmed for Bus Release and the software request bit was set during chaining, the channel will start the DMA operation as soon as chaining ends, without releasing the bus and will continue performing the operation until TC, MC or EOP.

When an active LOW $\overline{DREQ}$ is applied to a channel programmed for Bus Release, the channel will acquire the bus and perform DMA operations: (a) until TC, MC or EOP or (b) until $\overline{DREQ}$ goes inactive. Timing Diagram 2 shows when $\overline{DREQ}$ is sampled to determine if the channel should perform another cycle or release the bus. Note that this sampling also occurs on the last cycle of a chaining operation. If a channel has an active $\overline{DREQ}$ at the end of chaining, it will begin performing DMA operations immediately, without releasing the bus. When a TC, MC or EOP occurs, terminating a Bus Release mode operation, the channel, if enabled for chaining and/or Base-to-Current reloading, will perform reloading and/or chaining (assuming the Status register's SIP bit is clear) without releasing the bus.

If the SIP bit in the Channel Status register is set when a DMA termination occurs, the channel will relinquish the bus control until an Interrupt Acknowledge has been received and the SIP bit is cleared. After an interrupt has been serviced, the channel will perform the Base-to-Current reloading and/or chaining if enabled for the termination.

If an active request is not applied and the channel is in Demand Dedicated with Bus Hold, the channel will go into state THLD (see Timing Diagram 2(a)). If an active request is not applied and the channel is in Demand Dedicated with Bus Release or Demand Interleave mode, it will release the bus. Note that even if an active request is applied in Demand Interleave, the channel may still release the bus. The request for Demand Interleave should continue to be applied to ensure that the channel eventually responds to the request by acquiring the bus (i.e., the request is not latched by the channel).

Demand Interleave

Demand Interleave behaves in different ways depending on the setting of Master Mode register bit MM. If MM is set, the UDC will always relinquish bus control and then re-request it after each DMA iteration. This permits the CPU and other devices to gain bus control. If MM is clear, control can pass from one UDC channel to the other without requiring the UDC to release bus control. If both channels have active requests, control will pass to the channel which did not just have control. For instance if MM is clear and both channels have active requests and are in Demand Interleave mode, control will toggle between the channels after each DMA operation iteration and the UDC will retain bus control until both channels are finished with the bus. If MM is set and both channels have active requests and are in Demand Interleave mode, each channel will relinquish control to the CPU after each iteration resulting in the following control sequence: channel 1, CPU, channel 2, CPU, etc. Note that if there are other devices on the bus, they may gain control during the part of the sequence labelled CPU. See Appendix B for flowchart.

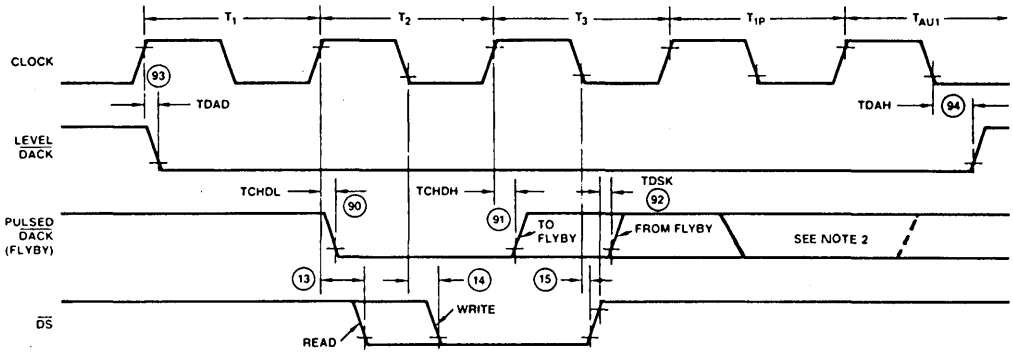
A software or hardware request will cause a channel programmed for Demand Interleave to perform interleaved DMA operations until TC, MC or EOP. If the Software request bit is set during chaining, the channel will retain the bus after chaining and will immediately start performing a DMA iteration and will interleave all DMA iterations after the first. If $\overline{DREQ}$ is LOW on the last cycle during chaining, the channel will perform a single iteration immediately after chaining and interleave thereafter until: (a) TC, MC or EOP or (b) $\overline{DREQ}$ goes HIGH. If (b) occurs, the channel will relinquish the bus until $\overline{DREQ}$ goes LOW again and the channel again starts performing interleaved operations. If (a) occurs, the channel will not interleave before first performing chaining and/or Base-to-Current reloading (assuming SIP is cleared).

The waveform of $\overline{DACK}$ is programmed in Channel Mode Register (CM_{18}). The Pulsed $\overline{DACK}$ is for flyby transaction only. See Timing Diagram 3. Note: This figure shows a single Search or Flyby iteration. State TWA is optionally inserted if programmed. For more than one iteration, the level $\overline{DACK}$ output would stay active during the time the channel had bus control. When CM_{18} is set, the $\overline{DACK}$ output will be inactive for all nonflyby modes.

Wait States

The number of wait states to be added to the memory or I/O transfer can be programmed by the user as 0, 1, 2 or 4 and can be separately programmed for the Current Address registers A and B and for the Chain Address register. This allows different speed memories and peripherals to be associated with each of these addresses. The Base Address registers A and B also have a Tag Field which is loaded into the Current ARA and ARB registers during Base-to-Current reloading. Because many users utilizing the software programmable wait states will not need the ability to generate hardware wait states through the $\overline{WAIT}$ pin, the wait function can be disabled by clearing the Wait Line Enable bit (MM_2) in the Master Mode register.

TIMING DIAGRAM 3. $\overline{DACK}$ Timing



WF007521

Notes: 1. Level $\overline{DACK}$ RE occurs as shown if auto-reloading is not programmed; otherwise, it stays LOW for three additional clocks.

2. This extra $\overline{DACK}$ pulse occurs only at EOP. It should be used to distinguish which channel got the EOP.

During DMA transactions, the $\overline{WAIT}$ input is sampled in the middle of the T_2 state. If $\overline{WAIT}$ is HIGH, and if no programmable wait states are selected, the UDC will proceed to state T_3. Otherwise, at least one wait state will be inserted. The $\overline{WAIT}$ line is then sampled in the middle of state TWA. If $\overline{WAIT}$ is HIGH, the UDC will proceed to state T_3. Otherwise additional wait states will be inserted. (See Timing Diagram 4.)

Consider what happens in a transaction when both hardware and software wait states are inserted. Each time the $\overline{WAIT}$ line is sampled, if it is LOW, a hardware wait state will be inserted in the next cycle. The software wait state insertion will be suspended until $\overline{WAIT}$ is sampled and is HIGH. The hardware wait states may be inserted anytime during the software wait state sequence. It is important to know that hardware wait states are served consecutively rather than concurrently with software wait states. For example, assume for a Flowthru I/O Transaction that a user has programmed 4 software wait states. Driving a LOW on the $\overline{WAIT}$ input during T_2 for 2 cycles would insert 2 hardware wait states. Driving $\overline{WAIT}$ HIGH for 3 cycles would allow insertion of three of the four software wait states. Driving $\overline{WAIT}$ LOW for 2 more cycles would insert 2 more hardware wait states. Finally, driving $\overline{WAIT}$ HIGH would allow the final software wait state to be inserted. During this last software wait state, the $\overline{WAIT}$ pin would be sampled for the last time. If it is HIGH, the channel will proceed to state T_3. If the pin is LOW, the channel will insert hardware wait states until the pin goes HIGH and the channel would then enter state T_3 to complete the I/O transaction.

DMA Transactions

There are three types of transactions performed by the Am9516A UDC: Flowthru, Flyby and Search. Figures 14 and 15 show the configurations of Flowthru and Flyby Transactions.

Flowthru Transactions

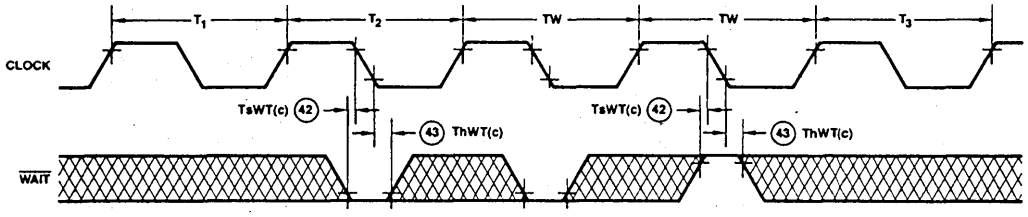
A Flowthru Transaction consists of Read and Write cycles. Each cycle consists of three states: T_1, T_2, and T_3 as shown in Timing Diagram 5. The user may select to insert software wait states through the Tag fields of the Current ARA and ARB registers. In addition, if Master Mode register bit $MM_2 = 1$, hardware wait states may be inserted by driving a LOW signal on the $\overline{WAIT}$ pin.

The M/$\overline{IO}$ and N/$\overline{S}$ lines will reflect the appropriate level for the current cycle early in T_1. The TG_6 and TG_7 bits of the current ARA and ARB registers should be programmed properly. The ALE output will be pulsed HIGH to mark the beginning of the cycle. The offset portion of the address for the peripheral being accessed will appear on $AD_0 - AD_{15}$ during T_1. The R/$\overline{W}$ and B/$\overline{W}$ lines will select a read or write operation for bytes or words. The R/$\overline{W}$, N/$\overline{S}$, M/$\overline{IO}$ and B/$\overline{W}$ lines will become stable during T_1 and will remain stable until after T_3.

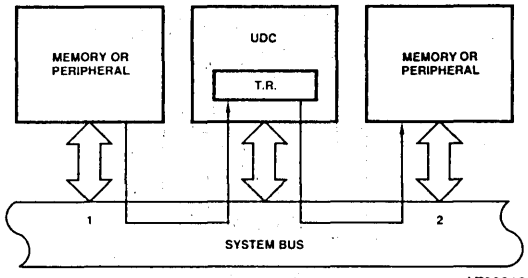
I/O address space is byte-addressed, but both 8- and 16-bit data sizes are supported. During I/O transactions the B/$\overline{W}$ output signal will be HIGH for byte transactions and LOW for word transactions. For I/O transactions, both even and odd addresses can be output; hence, the address bit output on AD_0 may be 0 or 1.

The channel can perform both I/O read and I/O write operations; the M/$\overline{IO}$ line will be LOW. During an I/O read, the $AD_0 - AD_{15}$ bus will be placed in the high-impedance state by the UDC during T_2. The UDC will drive the $\overline{DS}$ output LOW to signal the peripheral that data can be gated onto the bus. The UDC will strobe the data into its Temporary register during T_3. $\overline{DS}$ will be driven HIGH to signal the end of the I/O transaction. During I/O write, the UDC will drive the contents of the Temporary register onto the $AD_0 - AD_{15}$ bus and shortly after will drive the $\overline{DS}$ output LOW until T_3. Peripherals may strobe the data on AD bus into their internal registers on either the falling or rising edge. If the peripheral is to be accessed in a Flyby transaction also, data should be written on the rising edge of $\overline{DS}$ only.

TIMING DIAGRAM 4. $\overline{WAIT}$ Timing

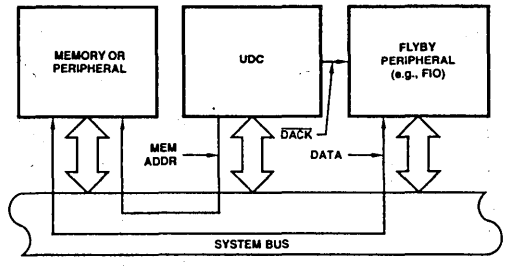


WF007680



AF003120

Figure 14. Configuration of Flowthru Transaction



AF003130

Figure 15. Configuration of Flyby Transaction

For byte I/O writes, the channel will drive the same data on data bus lines $AD_0 - AD_7$ and $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD_0 is 0, the UDC will strobe data in from data lines $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD_0 is 1, the UDC will strobe data in from data lines $AD_0 - AD_7$. Thus, when an 8-bit peripheral is connected to the bus, its internal registers will typically be mapped at all even or all odd addresses. To simplify accesses to 8-bit peripherals, byte oriented I/O addresses are incremented/decremented by 2.

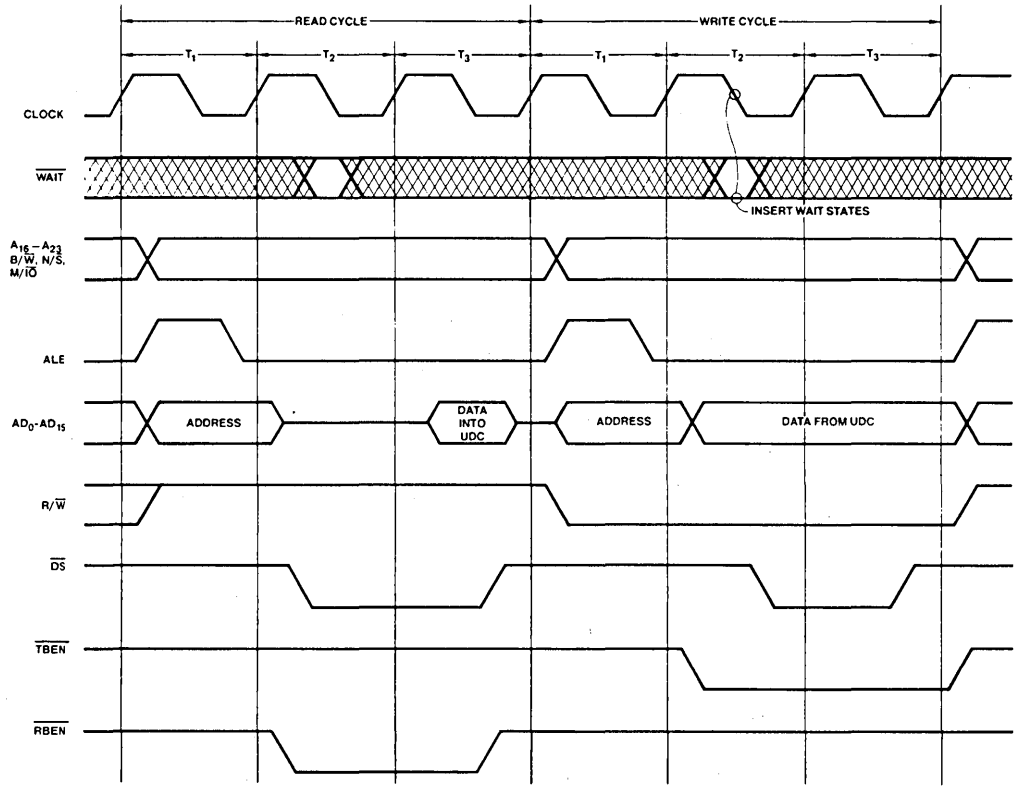
The channel can perform the I/O read and memory write operation, the memory read and I/O write operation, and the memory read and memory write operation, also. The timing for all Flowthru transactions is the same.

During chaining operations the UDC reads words from an address in System memory pointed to by the active channel's Chain Address register. Those chaining operations are performed identically to the Flowthru memory read transactions, except that the data is loaded into an internal UDC channel register rather than the Temporary register. Note that chaining never causes a write or a byte read; thus, all memory writes or all byte accesses are due to DMA operations. A typical memory operation consists of three states: T_1, T_2, and T_3, as shown in Timing Diagram 5. The user may select to insert 1, 2 or 4 software wait states after state T_2 and before state T_3 by programming the Tag field of the Current Address register or the Chain Address register. If the Wait Line Enable bit in the Master Mode register is set, the user may also insert hardware wait states after state T_2 and before state T_3 by driving a LOW on the $\overline{WAIT}$ line. The operation of Flowthru memory transactions is performed identically to the Flowthru I/O transactions. (See Timing Diagram 5.)

Flyby Transactions

Flyby Transfer and Flyby Transfer-and-Search operations are performed in a single cycle, providing a transfer rate significantly faster than that available from Flowthrus. In Flyby, operations can only be performed between memory and peripheral or between peripheral and peripheral. Memory-to-Memory operations cannot be performed in Flyby mode; these must be done using Flowthru.

TIMING DIAGRAM 5. Flowthru Transactions



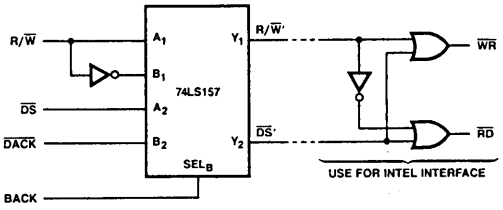
WF007540

The Flyby Transaction can only be used with peripherals having a special Flyby signal input or with external logic. This Flyby input is connected to the channel's $\overline{DACK}$ output. For memory-peripheral Flyby, the address of the source memory location must be programmed in the Current ARA register. The Current ARB register must be programmed with the destination memory location for peripheral-memory Flyby. For Flyby peripheral-to-peripheral transaction, if both peripherals have a Flyby input, only one (called "flyby peripheral") should be connected to $\overline{DACK}$; the other peripheral's Flyby input should be held high during the Flyby operation. The address of the peripheral (called "non-flyby peripheral") not connected to the channel's $\overline{DACK}$ output should be programmed in the Current ARB register when it is a destination. When the non-flyby peripheral is a source, its address should be programmed in the current ARA register. Note that a set Flip bit ($CM_4 = 1$) is for Flyby peripheral to Non-Flyby peripheral or Memory Write transaction (defined as "From Flyby Transaction"), and a clear Flip bit ($CM_4 = 0$) is for the memory or non-flyby peripheral read to Flyby peripheral transaction (defined as "To Flyby Transaction").

Transaction	CM_4	R/$\overline{W}$	Address of Memory or Non-Flyby Peripheral
To Flyby	0	HIGH	ARA
From Flyby	1	LOW	ARB

A Flyby operation is performed using three states: T_1, T_2, and T_3. During T_1 the channel pulses ALE and outputs the address information. See Timing Diagram 6. The R/$\overline{W}$ line is HIGH for "To Flyby" Transaction, and the R/$\overline{W}$ line is LOW for "From Flyby" Transaction.

The channel's M/$\overline{IO}$ and N/$\overline{S}$ lines are coded as specified by the Current ARA or ARB Tag field. The B/$\overline{W}$ line indicates the operand size programmed in the Channel Mode register Operation field. During state T_1 the channel drives R/$\overline{W}$ line to indicate the transaction direction. During state T_2 the channel drives both $\overline{DS}$ and $\overline{DACK}$ active. The Flyby Peripheral connected to $\overline{DACK}$ inverts the R/$\overline{W}$ signal to determine whether it is being read from or written to (see Figure 17).



$R/\overline{W}' = \overline{BACK \cdot R/\overline{W} + \overline{BACK} \cdot \overline{R/W}}$

$\overline{DS}' = \overline{BACK \cdot DACK + \overline{BACK} \cdot DS}$

$\overline{RD} = \overline{DACK \cdot \overline{R/W} \cdot BACK + DS \cdot R/\overline{W} \cdot \overline{BACK}}$

$\overline{WR} = \overline{DACK \cdot R/W \cdot BACK + \overline{DS} \cdot R/W \cdot \overline{BACK}}$

AF003140

Figure 16. Flyby Peripheral Interface

The pulsed $\overline{DACK}$ input serves two purposes: to select the peripheral for the Read/Write, and to provide timing information on when to drive data onto or input data from the $AD_0 - AD_{15}$ bus. Note that because the "Flyby Peripheral" never gets explicitly addressed by $AD_0 - AD_{15}$, it must know which internal register is to be loaded from or driven onto the $AD_0 - AD_{15}$ bus. On state T_3, the $\overline{DS}$ and $\overline{DACK}$ lines are driven inactive to conclude the transfer. In Transfer-and-Search mode, data is loaded into the UDC's Temporary register on the LOW-to-HIGH $\overline{DS}$ transition to perform the Search function.

To provide adequate data setup time, the rising edge of $\overline{DS}$ or $\overline{DACK}$ should be the edge used to perform the write to the transfer destination. To extend the active time of $\overline{DS}$ and $\overline{DACK}$, wait states can be inserted between T_2 and T_3. Software wait states can be inserted by programming the appropriate code in the Tag field of the Current ARA or ARB registers. Hardware wait states can be inserted by pulling $\overline{WAIT}$ LOW if the Wait Line Enable bit in the Master Mode register is set. The $\overline{WAIT}$ line is sampled in the middle of the T_2 or TWA state.

Termination

There are three ways a Transfer-and-Search or Search operation can end and two ways a Transfer operation can end. When a channel's Current Operation count goes to 0, the DMA operation being performed will end. This is called a TC or Terminal Count termination. A DMA operation can also be stopped by driving the $\overline{EOP}$ pin LOW with external logic. This is called an EOP termination. Search and Transfer-and-Search operations have a third method of terminating called Match Condition or MC termination. An MC termination occurs when the data being Transferred-and-Searched or Searched meets the match condition programmed in Channel Mode register bits $CM_{17} - CM_{16}$. These bits allow the user to stop when a match occurs between the unmasked Pattern register bits and the data read from the source, or when a no-match occurs. Both byte and word matches are supported. MC terminations do not apply to Transfer operations since the pattern matching logic is disabled in Transfer mode.

End-of-Process

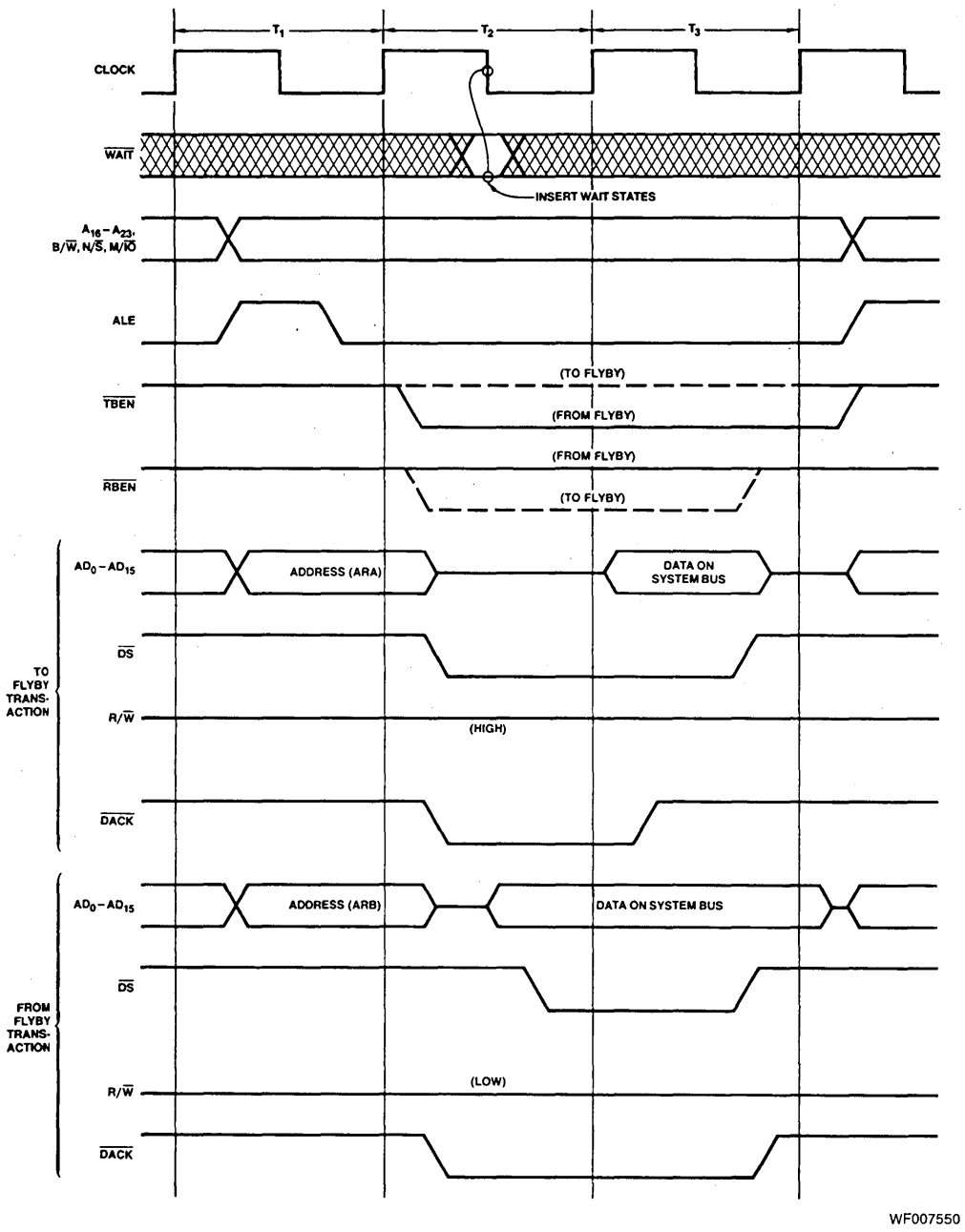
The End-of-Process (EOP) interface pin is a bi-directional signal. Whenever a TC, MC or EOP termination occurs, the UDC will drive the $\overline{EOP}$ pin LOW. During DMA operations, the $\overline{EOP}$ pin is sampled by the UDC to determine if the $\overline{EOP}$ is being driven LOW by external logic. Timing Diagram 7 shows when internal EOPs are generated marking termination of all Transfers. These figures also show the point during the DMA iteration when the EOP pin is sampled. The generation of internal EOPs and sampling of external EOPs for Transfer-and-Searches follow the same timing used for Transfers. Since there is a single $\overline{EOP}$ pin for both channels, $\overline{EOP}$ should only be driven LOW by a channel while that channel is being serviced. This can be accomplished by selecting a level $\overline{DACK}$ output ($CMR_{18} = 0$) and gating each channel's $\overline{EOP}$ request with $\overline{DACK}$, as shown in Figure 17.

If an EOP is detected while the channel is trying to reload the Chain Address register, the new Chain Address Offset and Segment are discarded and the old address +2 is preserved to allow inspection of the erroneous address.

Programming Completion Options

When a channel ends a DMA operation, the reason for ending is stored in the Completion Status Field of the channel's Status register. See Figure 5. This information is retained until the next DMA operation ends at which time the Status register is updated to reflect the reason(s) for the latest termination. Note that it is conceivable that more than one bit in the Completion Field could be set. An as extreme example, if a channel decremented its Current Operation count to zero, causing a TC termination; input data from the source generated a match causing an MC termination; and there was a LOW on the $\overline{EOP}$ pin resulting in an EOP termination, all three of the channel's Status register completion bits would be set.

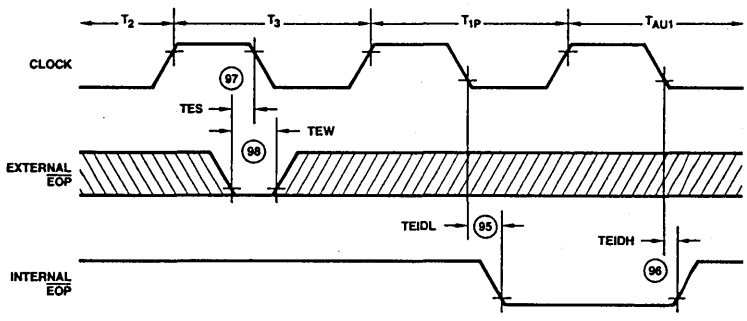
TIMING DIAGRAM 6. Flyby Transactions



WF007550

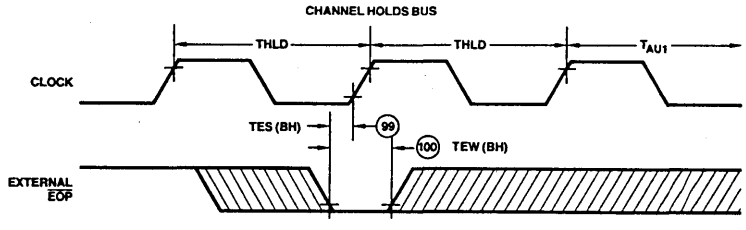
TIMING DIAGRAM 7. $\overline{EOP}$ Timing

(a) $\overline{EOP}$ Sampling and Generation During DMA Operations



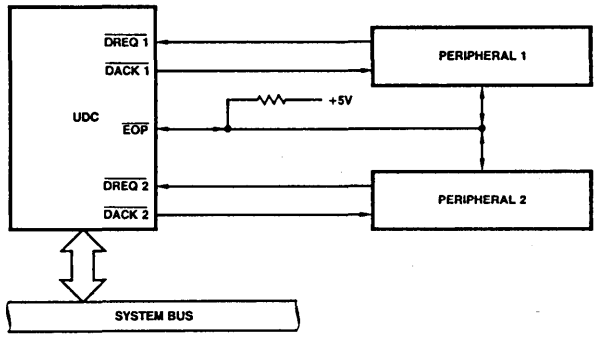
WF007560

(b) Sampling of $\overline{EOP}$ During Bus Hold



WF007570

Notes: 1. The diagram lists state names for both I/O and memory accesses. Sampling of $\overline{EOP}$ will occur on the falling edge of state T_3.
2. State T_{1P} is a pseudo-T_1 state, generated following termination of any DMA operation.
3. T_{AU1} is an auto-intialization state, generated following the TC, MC or $\overline{EOP}$ termination.



AF003150

Figure 17. $\overline{EOP}$ Connection

When a DMA operation ends, the channel can:

(a) Issue an Interrupt request (i.e., setting the IP or SIP bit of the channel's Status register);

(b) Perform Base-to-Current reloading;

(c) Chain reload the next DMA operation;

(d) Perform any combination of the above; or

(e) None of the above.

The user selects the action to be performed by the channel in the Completion option field of the Channel Mode register. For each type of termination (TC, MC or $\overline{EOP}$), the user can choose which action or actions are to be taken. If no reloading is selected for the type of termination that occurred, the NAC bit in the Status register will be set.

More than one action can occur when a DMA operation ends. This may arise because more than one action was programmed for the applicable termination. The priorities of those actions are Interrupt request first, Base-to-Current reloading second, and then chaining. The Interrupt cannot be serviced unless the UDC has relinquished the bus.

Interrupts

To allow the UDC to start executing a new DMA operation after issuing an Interrupt, but before an Interrupt acknowledge is received, a two-deep Interrupt queue is implemented on each channel. The following discussion will describe the standard Interrupt structure and then elaborate on the additional Interrupt queuing capability of the UDC.

A complete Interrupt cycle consists of an Interrupt request followed by an Interrupt-acknowledge transaction. The request, which consists of $\overline{INT}$ being pulled LOW, notifies the CPU that an Interrupt is pending. The Interrupt-acknowledge transaction, which is initiated by the CPU as a result of the request, performs two functions: it selects the peripheral whose Interrupt is to be acknowledged, and it obtains a vector that identifies the selected device and operation — the cause of the Interrupt.

A peripheral can have one or more sources of Interrupt. Each Interrupt source has two bits that control how it generates Interrupts. These bits are a Channel Interrupt Enable bit (CIE) and an Interrupt Pending bit (IP). On the UDC, each channel is an Interrupt source. The two Interrupt control bits are located in bits CM_{15} and CM_{13} of each channel's Status register.

Each channel has its own vector register for identifying the source of the Interrupt during an Interrupt acknowledge transaction. There is one bit (MM_3) in the Master Mode register used for controlling Interrupt behavior for the whole device.

Once a channel issues an Interrupt, it is desirable to allow the channel to proceed with the next DMA operation before the Interrupt is acknowledged. This could lead to problems if the UDC channel attempted to chain reload the Vector register contents. In such a situation, it may not be clear whether the old or new vector would be returned during the acknowledge. This dilemma is resolved in the UDC by providing each channel with an Interrupt Save register. When the channel sets IP as part of the procedure followed to issue an Interrupt, the contents of the Vector register and some of the Status register bits are saved in an Interrupt Save register. See Figure 7. When an Interrupt Acknowledge cycle is performed, the contents of the Interrupt Save register are driven onto the bus. Although the use of an Interrupt Save register allows the channel to proceed with a new task, problems can still potentially arise if a second Interrupt is to be issued by the channel before the first Interrupt is acknowledged. To avoid conflicts between the first and second Interrupt, each channel has a Second Interrupt Pending (SIP) bit in its Status register. When a second Interrupt is to be issued before the first Interrupt is acknowledged, the SIP bit is set and the channel relinquishes the bus until an acknowledge occurs. For compatibility with polled Interrupt schemes, the Interrupt save register can be read by the host CPU without wait states. As an aid to debugging a system's Interrupt logic, whenever IP is set, the Interrupt Save register is loaded from the Vector and Status registers.

Note that the SIP bit is transferred to the IP bit when IP is cleared by the host CPU. Whenever CIE is set, $\overline{INT}$ will go LOW as soon as IP is set.

Base-to-Current Reloading

When a channel finishes a DMA operation, the user may select to perform a Base-to-Current reload. (Base-to-Current reloading is also referred to as Auto-reloading in this document.) In this type of reload, the Current Address registers A and B are loaded with the data in the Base Address registers A and B respectively, and the Current Operation Count register is loaded with the data in the Base Operation Count. The Base-to-Current reload operation facilitates repetitive DMA operations without the multiple memory accesses required by chaining. Although the channel must have bus control to perform Base-to-Current reloading, the complete reloading operation occurs in four clock cycles (i.e., TAU_1 through TAU_4). Note that if the channel had to relinquish the bus because two unacknowledged interrupts were queued, it will have to regain bus control to perform any Base-to-Current reloading (or chaining, for that matter). In this case it acquires the system bus once an Interrupt acknowledge is received, even if it immediately afterward will relinquish the bus because no hardware/software request is present.

Chaining

If the channel is programmed to chain at the end of a DMA operation, it will use the Chain Address register to point to a Chain Control Table in memory. The first word in the table is a Reload word, specifying the register(s) to be loaded. Following the Reload word are the data values to be transferred into the register(s). Chaining is described in detail in the "Channel Initialization" section.

Because chaining occurs after Base-to-Current reloading, it is possible to reset the Current Address registers A and B and the Current Operation Count register to the values used for previous DMA operations, then chain reload one or two of these registers to some special value to be used, perhaps, for this DMA operation only. If the Base values are not reloaded during chaining, the channel can revert back to the Base values at a later cycle.

If an all zero Reload word is fetched during chaining, the chain operation will not reload any registers, but in all other respects, it will perform like any other chaining operation. Thus, the Chain Address will be incremented by 2 to point to the next word in memory, and at the end of the all Zero-Reload word chain operation, the channel will be ready to perform a DMA operation. All Zero-Reload words are useful as "Stubs" to start or terminate linked lists of DMA operations traversed by chaining. On the other hand, care must be taken in their use since the channel may perform an erroneous operation if it is unintentionally started after the chaining operation.

COMMAND DESCRIPTIONS

Figure 18 shows a list of UDC commands. The commands are executed immediately after being written by the host CPU into the UDC's Command register (Figure 19). A description of each command follows.

Reset (00)

This command causes the UDC to be set to the same state generated by a Hardware Reset. The Master Mode register is set to all zeros; the CIE, IP and SIP bits are cleared; the NAC and CA bits in each channel's Status register are set; and the channel activity is forbidden. The Chain Address must be programmed since its state may be indeterminate after a Reset. The lockout preventing channel activity is cleared by issuing a Start Chain command.

Start Chain Channel 1/Channel 2 (A_0/A_1)

This command causes the selected channel to clear the No Auto-Reload or Chain (NAC) bit in the channel's Status register and to start a chain reload operation of the channel's registers, as described in the "Channel Initialization" section. These effects will take place even if the fetched Reload word is all zeros. This command will only be honored if the Chain Abort (CA) bit and the Second Interrupt Pending (SIP) bit in the channel's Status register are clear. If either the CA or SIP bit is set, this command is disregarded.

When the Waiting For Bus (WFB) bit of Status register is set, if the "Start Chain" command is issued, the channel will honor the command after one DMA iteration. It is nearly impossible for the CPU to issue a command when WFB = 1 and the UDC is enabled.

Command	Opcode Bits 7654	Opcode Bits 3210	Example Code HEX
Reset	000X	XXXX	00
Start Chain Channel 1	101X	XXX0	A0
Start Chain Channel 2	101X	XXX1	A1
Set Software Request Channel 1	010X	XX10	42
Set Software Request Channel 2	010X	XX11	43
Clear Software Request Channel 1	010X	XX00	40
Clear Software Request Channel 2	010X	XX01	41
Set Hardware Mask Channel 1	100X	XX10	82
Set Hardware Mask Channel 2	100X	XX11	83
Clear Hardware Mask Channel 1	100X	XX00	80
Clear Hardware Mask Channel 2	100X	XX01	81
Set CIE, or, IP Channel 1	001E	XP10	32*
Set CIE, or, IP Channel 2	001E	XP11	33*
Clear CIE, or, IP Channel 1	001E	XP00	30*
Clear CIE, or, IP Channel 2	001E	XP01	31*
Set Flip Bit Channel 1	011X	XX10	62
Set Flip Bit Channel 2	011X	XX11	63
Clear Flip Bit Channel 1	011X	XX00	60
Clear Flip Bit Channel 2	011X	XX01	61

*Notes: 1. E = Set to 1 to perform set/clear on CIE; Clear to 0 for no effect on CIE.
2. P = Set to 1 to perform set/clear on IP; Clear to 0 for no effect on IP.
3. X = "don't care" bit. This bit is not decoded and may be 0 or 1.

Figure 18. UDC Command Summary

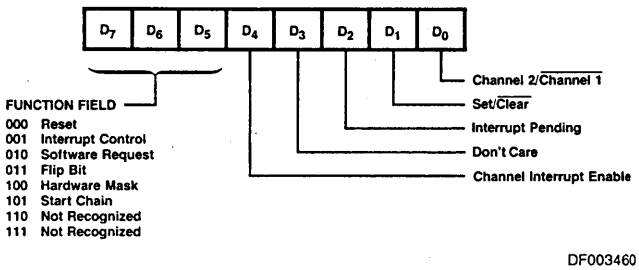


Figure 19. Command Register

Software Request Channel 1/Channel 2 (Set: 42/43, Clear: 40/41)

This command sets or clears the software request bit in the selected channel's Mode register. If the Second Interrupt Pending (SIP) bit and No Auto-Reload or Chain (NAC) bit in the channel's Status register are both cleared, the channel will start executing the programmed DMA operation. If either the SIP or NAC bit is set, the channel will not start executing a DMA operation until both bits are cleared. The SIP bit will clear when the channel receives an Interrupt acknowledge. One way to clear the NAC bit is to issue a Start Chain command to the channel. If the fetched Reload Word is all zeros, the channel's registers will remain unchanged and the software request bit, if set earlier by command, will cause the programmed DMA operation to start immediately. If during chaining new information is loaded into the Channel Mode register, this new information will, of course, overwrite the software request bit.

Set/Clear Hardware Mask 1/Mask 2 (Set: 82/83; Clear: 80/81)

This command sets or clears the Hardware Mask bit in the selected channel's Mode register. This command always takes effect. The Hardware Mask bit inhibits recognition of an active signal on the channel's $\overline{DREQ}$ input; this bit does not affect recognition of a software request. If the channel is in single transfer mode, it performs DMA operations upon receipt of a transition on $\overline{DREQ}$ rather than in response to a $\overline{DREQ}$ level. Transitions occurring while the Hardware Mask bit is set will be stored and serviced when the Hardware Mask is cleared, assuming the Channel has not chained. The UDC will request the system bus 1 1/2 to 2 clocks after the receipt of any $\overline{DREQ}$, after which a minimum of one DMA iteration is unavoidable. $\overline{DREQ}$ transitions are only stored for the current DMA operation. If the channel performs a chain operation of single transfer mode, any $\overline{DREQ}$ transition stored for later service is cleared.

Timing Diagrams 1 and 2 show the minimum times when a new $\overline{DREQ}$ can be applied if it is to be serviced by the new DMA operation. Note in Diagram 1 the notation of First iteration and Last iteration. This means, for example, $\overline{DREQ}$ may be asserted during the write cycle T_1 of a Flowthru transaction, but may never be asserted during T_1 of a Flyby transaction because Flyby is done in one iteration.

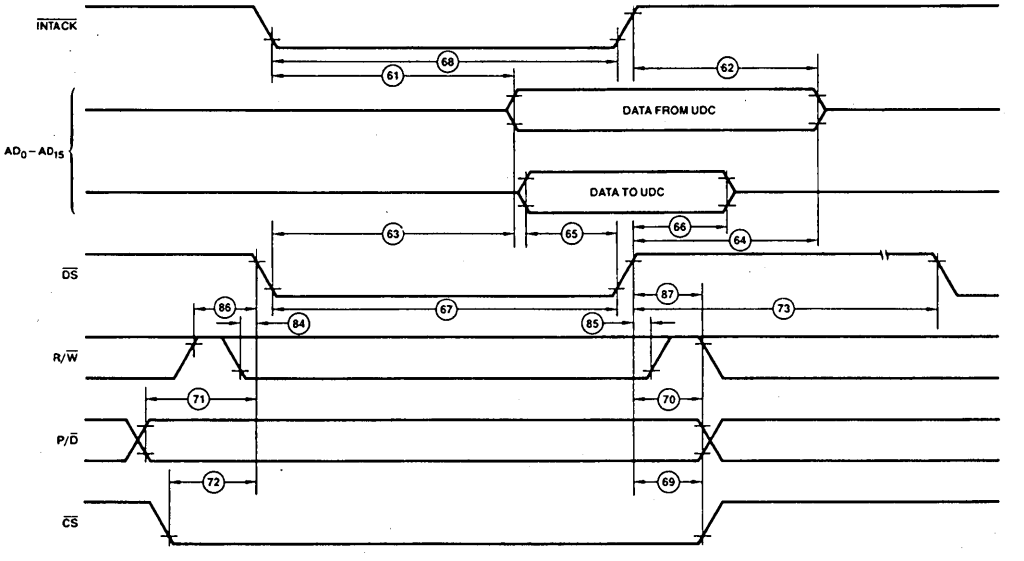
Set/Clear CIE, and IP Channel 1/Channel 2 (see Figure 18)

This command allows the user to either set or clear any combination of the CIE and IP bits in the selected channel's Status register. These bits control the operation of the channel's Interrupt structure and are described in detail in the "Interrupts" section. Setting the IP bit causes the Interrupt Save register to be loaded with the current Vector and Status. The IP bit is cleared to facilitate an efficient conclusion to the processing of an interrupt.

Set/Clear Flip Bit Channel 1/Channel 2 (Set: 62/63; Clear: 60/61)

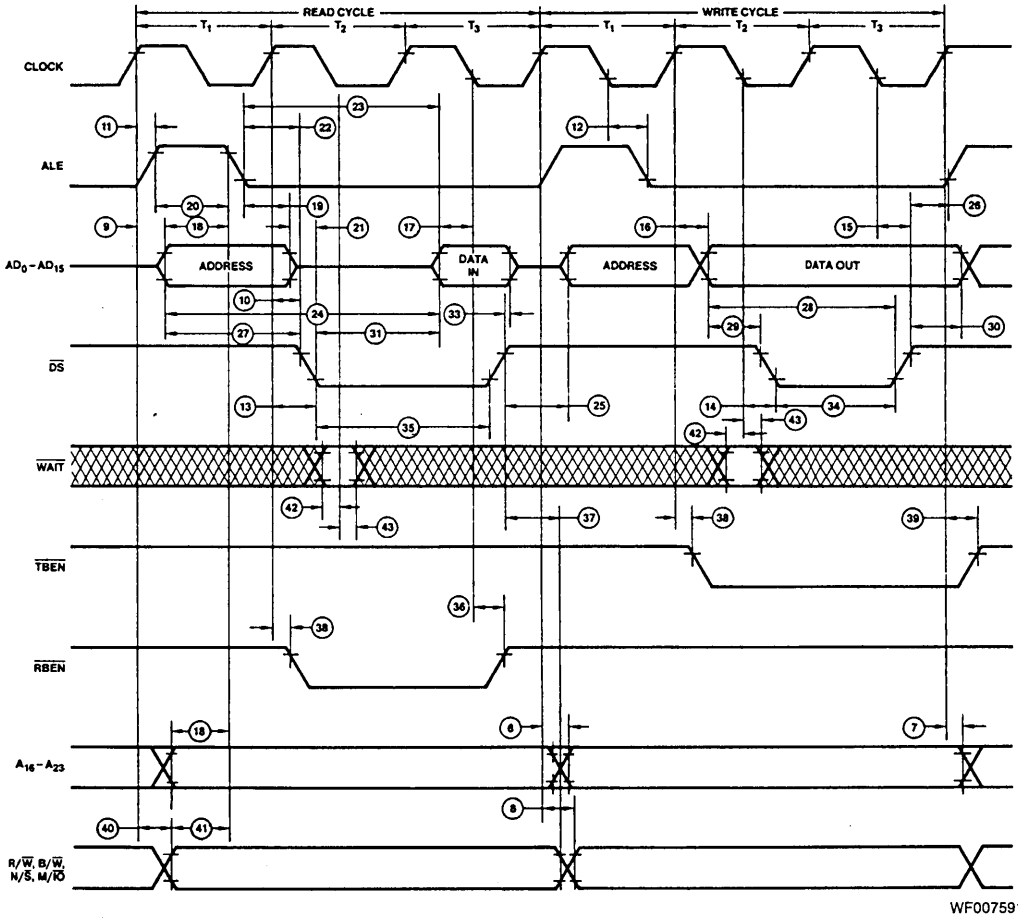
The Flip Bit in the selected channel's Mode register can be cleared and set by this command. This allows the user to reverse the source and destination and thereby reverse the data transfer direction without reprogramming the channel. This command will be most useful when repetitive DMA operations are being performed by the channel, using Base-to-Current reloading for channel reinitialization and using this command to control the direction of transfer. Chaining new information into the Channel Mode register will, of course, overwrite the Flip bit.

TIMING DIAGRAM 8. AC Timing when UDC is a Bus Slave



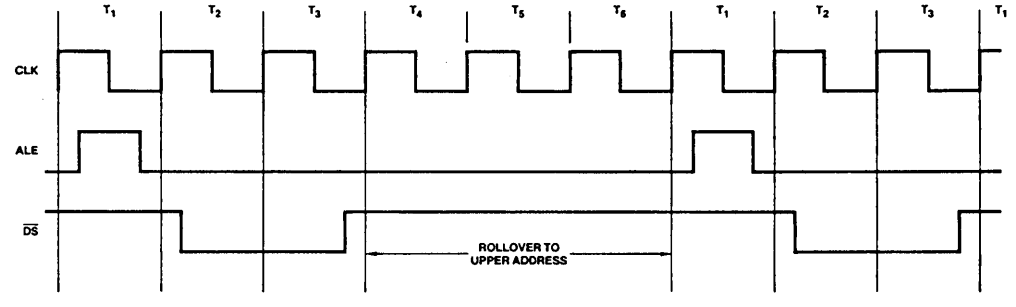
WF007580

TIMING DIAGRAM 9. AC Timing when UDC is a Bus Master



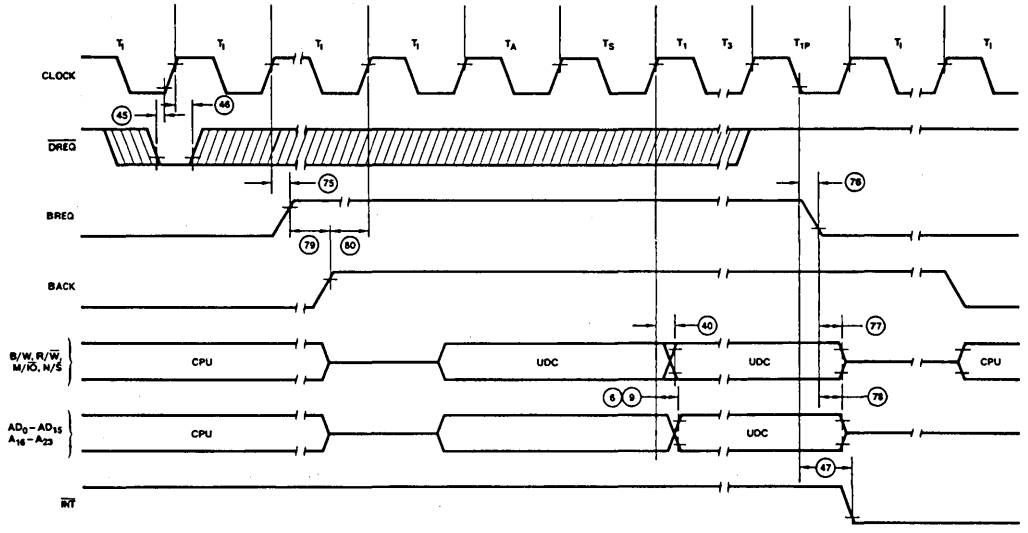
WF007591

TIMING DIAGRAM 10. Upper Address Rollover Timing



WF007600

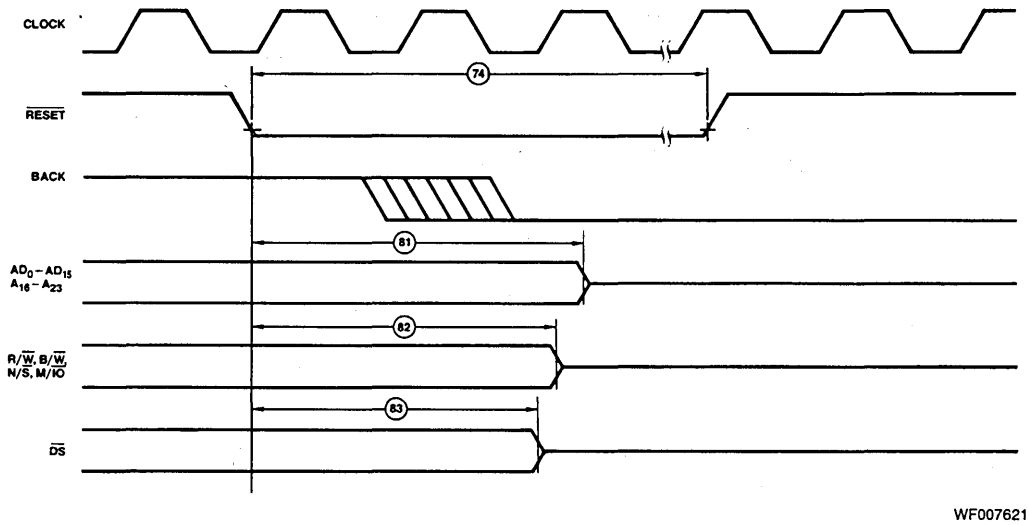
TIMING DIAGRAM 11. Bus Exchange Timing



WF007610

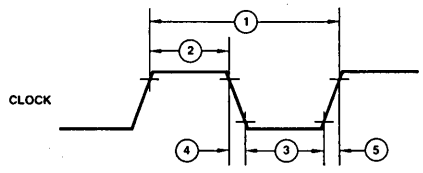
Notes: 1. Under no circumstance can BACK be removed prior to BREQ.
2. One extra ALE occurs each time the 9516 releases the bus. No $\overline{DS}$ accompanies it, so this should not present a problem.

TIMING DIAGRAM 12. Reset Timing



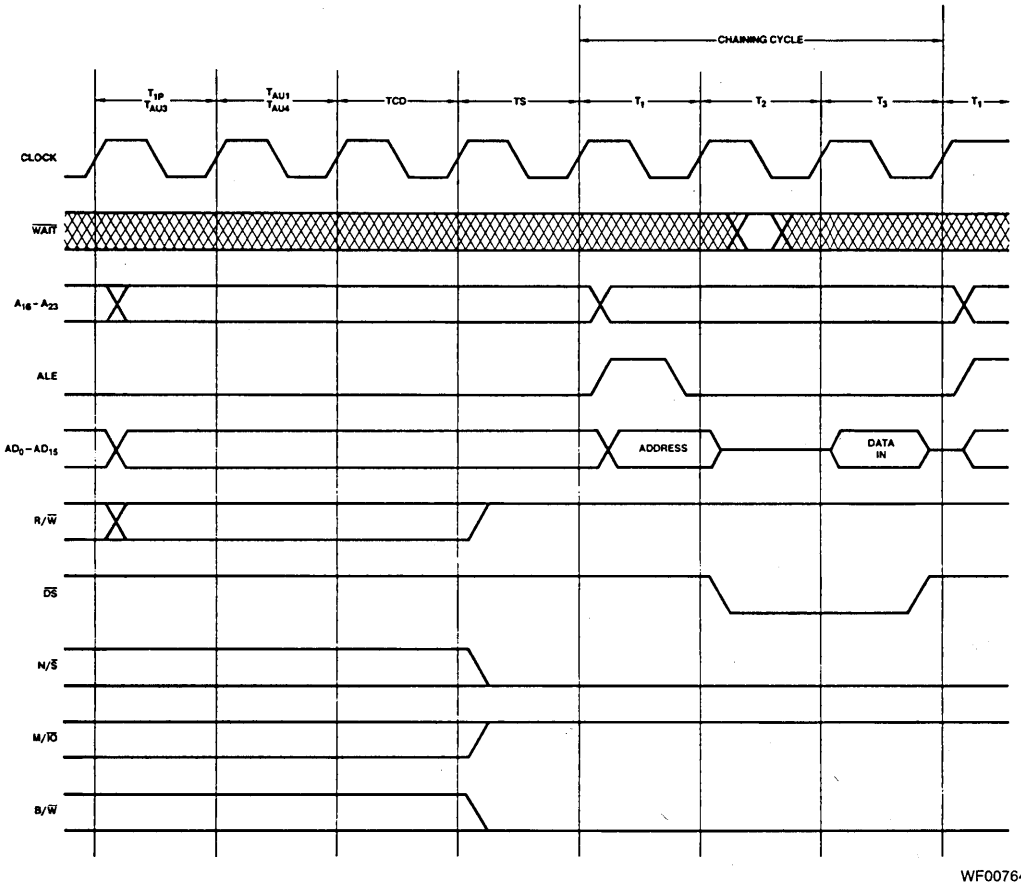
WF007621

TIMING DIAGRAM 13. Clock Waveform



WF007630

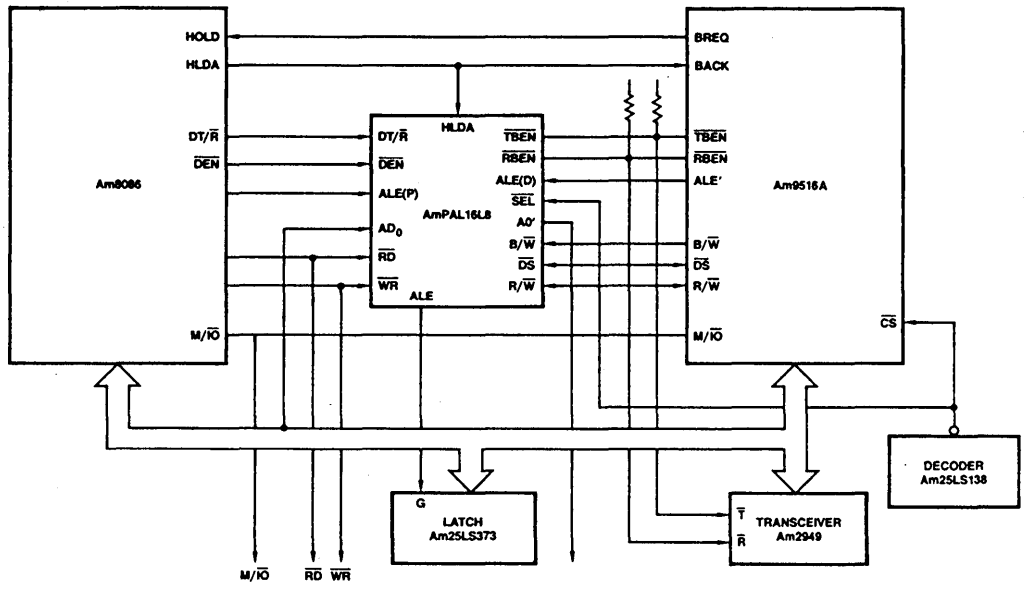
TIMING DIAGRAM 14. Timing During Chaining



WF007640

APPLICATIONS INFORMATION

Figures 20a and 20b show the configuration of an Am9516A UDC and an Am8086 microprocessor on the same board. Figure 21 shows a configuration for them when the Am9516A UDC is on a different board. The configuration of an Am9516A UDC to 68000 CPU interface is shown in Figure 22. An example of an Am8086 initialization program is shown in Figure 23. Figure 24 shows the reload table for chaining. The details of the Programmable Array Logic (PAL*) for those interfaces are described in Appendix B.



AF003161

Figure 20a. Am9516A UDC to Am8086 CPU Interface (Minimum Mode)

AmPAL16L8 PALASM FILE

PAL16L8

Pat 001

Am9516A to Am8086 min mode interface chip

Advanced Micro Devices

NC ALED ALEP HLDA BW AD_0 DT/DEN/SEL GND

NC/RBEN/RD ALE A_0/RW/DS/WR/TBEN V_{CC}

If (/HLDA) DS = RD + WR

If (/HLDA) RW = DT

If (/HLDA) TBEN = /DT*/SEL*DEN

If (/HLDA) RBEN = DT*/SEL*DEN

If (HLDA) RD = /RW * DS

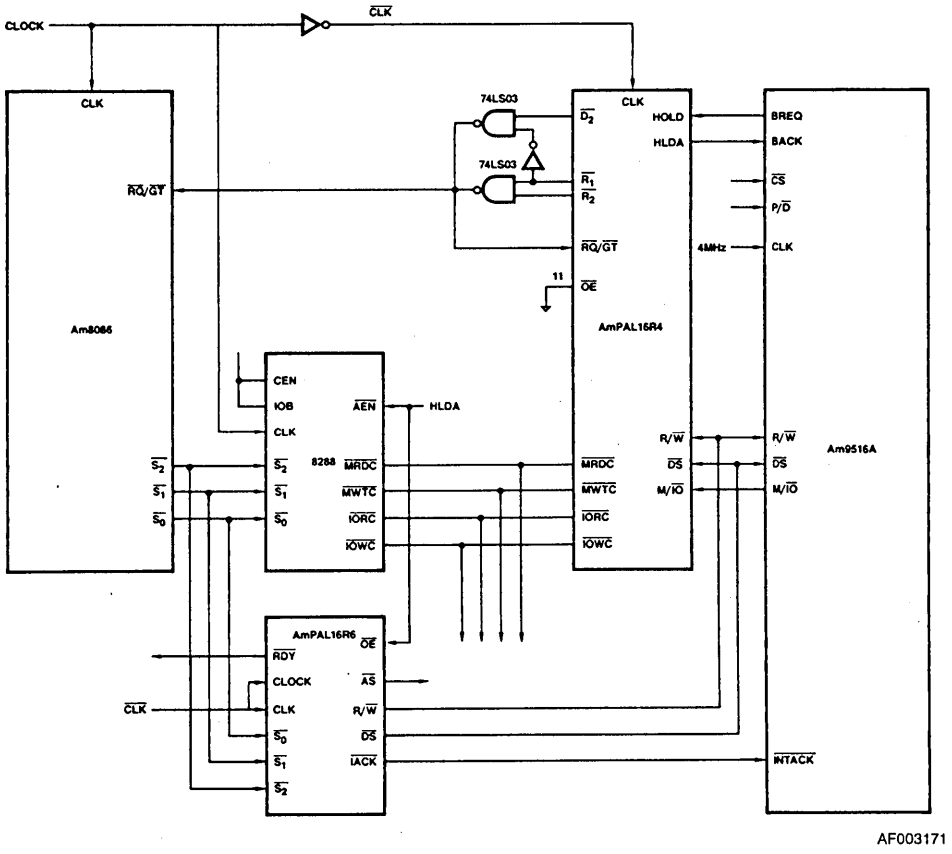
If (HLDA) WR = RW * DS

ALE = /ALEP * /ALED

A_0 = /AD_0*/BW*HLDA*ALED +
/AD_0*BW*HLDA*ALED +
/AD_0*/HLAD*ALEP + A_0*/ALEP + A_0*/ALED

DESCRIPTION

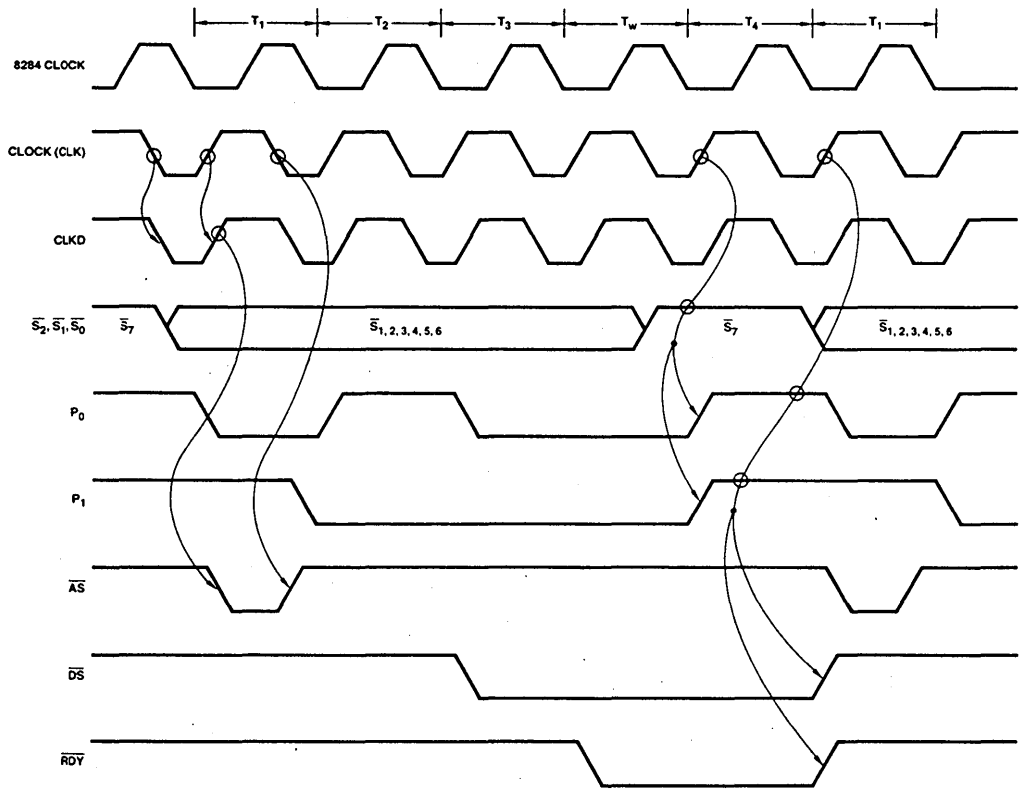
This PAL converts the control signals to interface the Am8086 in min mode to the Am9516A DMA controller. Another example shows how this is done in max mode.



AF003171

Figure 20b. Am9516A UDC to 8086 CPU Interface (Maximum Mode)

Timing Diagram of AmPAL16R6



WF007650

AmPAL16R6 PALASM FILE

AmPAL16R6
PAT003
Am8086 to AmZ85XX Peripheral Interface
Advanced Micro Devices

CLOCK RESET CLK/S_0/S_1/S_2 NC NC NC GND
/OE/AS/P_1/RW/DS/PO/LACK/RDY CLKD V_{CC}

P_0: = /RESET*S_0*/P_0*/P_1 +
/RESET*S_1*/P_0*/P_1 +
/RESET*S_2*/P_0*/P_1 +
/RESET*S_0*P_1 +
/RESET*S_1*P_1 +
/RESET*S_2*P_1

P_1: = /RESET*P_0*/P_1 +
/RESET*P_1*S_0 +
/RESET*P_1*S_1 +
/RESET*P_1*S_2

DS: = /IACK*/P_0*P_1*S_0*/S_1*S_2 +
/IACK*/P_0*P_1*/S_0*S_1*S_2 +
IACK*S_0*S_1*S_2 +
DS*P_0*P_1

RW: = S_0*/S_1

IACK: = /RESET*S_0*S_1*S_2 + IACK*P_0*P_1*/DS +
IACK*/P_0*/P_1

RDY: = /RESET*S_0*/S_1*S_2*P_0*P_1 +
/RESET*/S_0*S_1*S_2*P_0*P_1 +
/RESET*DS*RDY*P_0*P_1

/CLKD = CLK

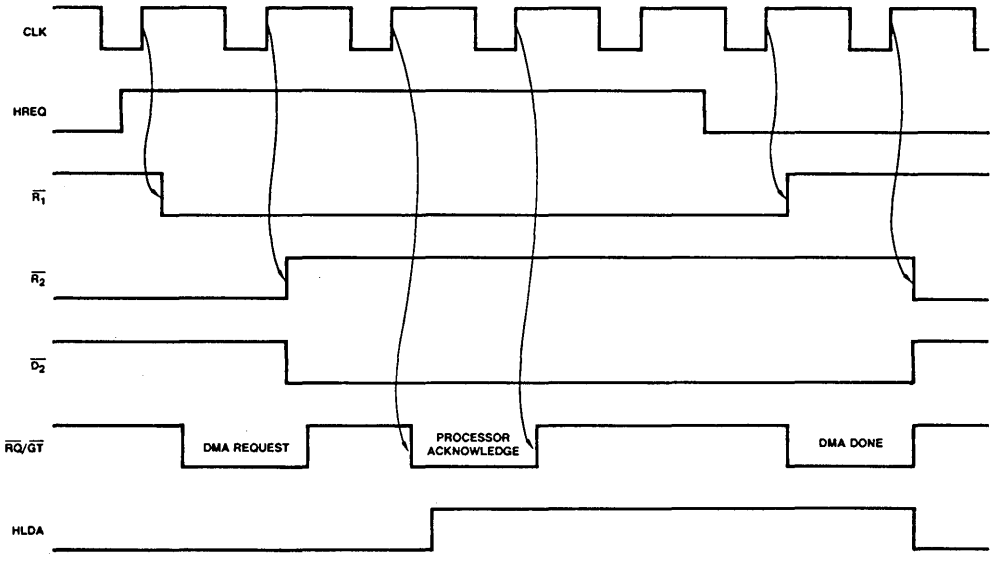
AS = /CLKD*P_0*/P_1*/IACK*CLK

DESCRIPTION

This PAL translates Am8086 bus signals into compatible signals for the Am9516A. It is also applicable to AmZ85XX peripherals by altering/RW and /DS to/RD and /WR. One flip-flop is available to give the necessary delay to the falling edge of/WR.

Note: The CLK signal must be externally inverted for this design.
A >

AmPAL16R4 Timing



WF007660

AmPAL16R4 PALASM FILE

B > Type Am9516A PAL
PAL16R4

Am8086 to Am9516A interface

Advanced Micro Devices

CLK/RQGT HOLD NC NC NC/RW/DS MIO GND
/OE/MWTC/MRDC HLDA/D_2/R_2/R_1/IOWC/IORC V_{CC}

If (HLDA) IORC = /MIO*DS*/RW
If (HLDA) IOWC = /MIO*DS*RW
If (HLDA) MRDC = MIO*DS*/RW
If (HLDA) MWTC = MIO*DS*RW

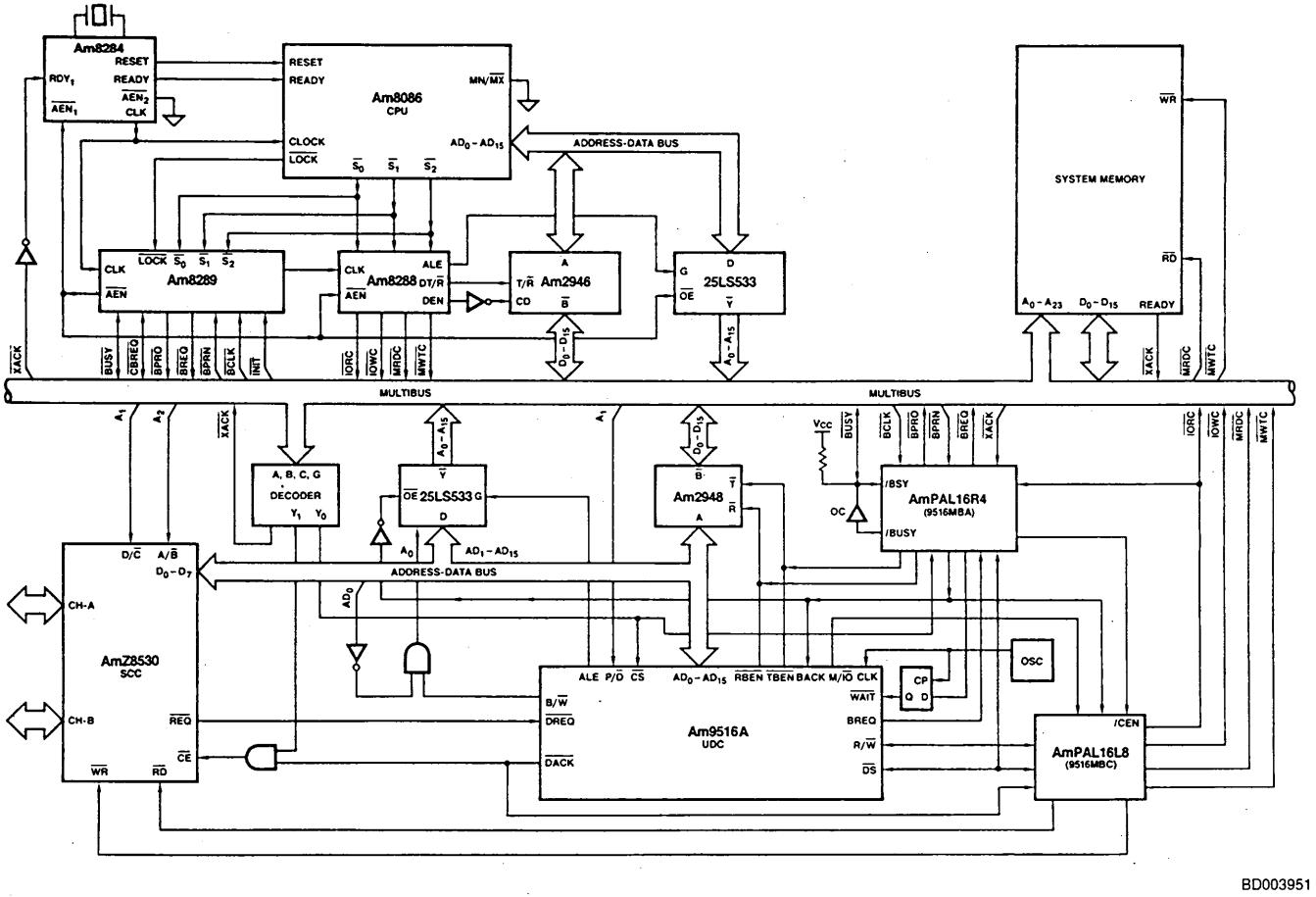
R_1: = HOLD
R_2: = /R_1
D_2: = R_1
/HLDA: = /R_1 + /D_2*/HLDA + /RQGT*/HLDA

DESCRIPTION

This device converts the min mode signals HOLD and HLDA to the max mode /RQGT protocol. Additionally, it generates the 8288 equivalent control outputs /MRDC, /MWTC, /IORC, and /IOWC. This PAL was used to connect the Am9516A to the Am8086 in max mode.

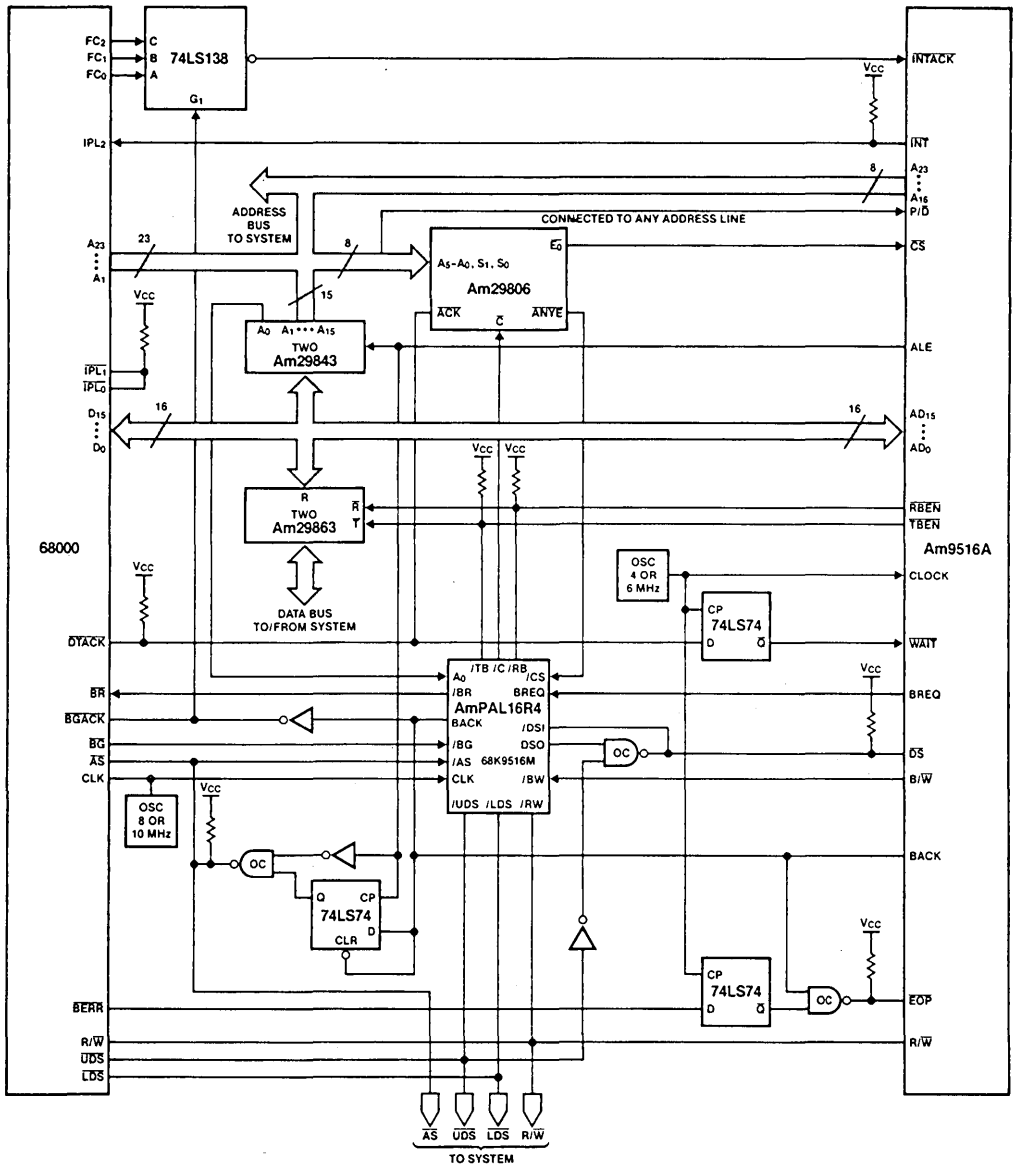
B >

Note: If HOLD is taken away prior to grant pulse, design will not work correctly because the release pulse will overlap the grant pulse.



BD003951

Figure 21. Am9516A to 8086 in a MULTIBUS Environment



BD003941

Figure 22. The Am9516A UDC to 68000 CPU Interface

```
PAL16R4                          PAL DESIGN SPECIFICATION
PAT006                             JOE BRCICH 9/01/83
68000 TO Am9516A INTERFACE WITH DATA HOLD CORRECTION
ADVANCED MICRO DEVICES

CLK   RW    AO    BREQ   /BG    /DSI    /AS    /BW    /CS
/OE   /LDS  /UDS  DSO    /C     BACK    /BR    /TB    /RB

IF (/BACK) RB  = /CS * RW * UDS      +
                 /CS * RW * LDS
IF (/BACK) TB  = /CS * /RW
IF (BACK) UDS  = DSI * /A0 * /BW     +
                 BW * DSI
IF (BACK) LDS  = DSI * A0 * /BW      +
                 BW * DSI

BR := BREQ * BG * BR * AS            +
      BREQ * /BG * /BACK

/BACK :=  /BREQ                        +
          /BREQ * /BG                  +
          /BREQ * AS                   +
          /BREQ * /BACK                +
          /BG * /BACK                  +
          AS * /BACK

C := UDS * /BACK                     +
     LDS * /BACK

/DSO := BACK                         +
        /BACK * /RW * C
```

DESCRIPTION

IF BREQ*BACK IS TRUE THE Am9516A HAS THE BUS, OTHERWISE THE 68000 HAS THE BUS. THIS PAL CONNECTS THE Am9516A TO THE 68000 WITH ONE WAIT STATE DURING WRITES WHILE SHORTENING /DS TO ACHIEVE PROPER DATA HOLD TIME. IT ALSO CONVERTS THE BUS EXCHANGE PROTOCOL INTO 68000 FORMAT. THIS DESIGN ASSUMES NO OTHER BUS MASTERS IN THE SYSTEM. /RB AND /TB CONTROL THE TRANCEIVERS WHEN CPU IS BUS MASTER. /CS MUST BE FUNCTION OF ALL DEVICES CONNECTED TO THE CPU BUS NOT JUST THE Am9516A /CS AS SHOWN HERE.

The /CS to /DS set-up time of 30ns is met in the following ways:

(1) During a read cycle the only effect from not meeting this set-up time is that the data valid access time from the Am9516A will be delayed by a proportional amount. Since the minimum /DS Low width from the 10-MHz 68000 (during a read) is 193ns and the minimum /DS Low width to the Am9516A is 150ns, we have 43ns margin not counting gate delays which will further increase this margin.

(2) During a write cycle this is not an issue since the /DS comes later and is stretched longer due to the Wait state.

AmPAL16R4 68K9516M PALASM File

```
PAL16L8                    PAL DESIGN SPECIFICATION 9516MBC
PAT 003                          JOE BRCICH 26 JULY 84
MULTIBUS CONTROL FOR Am9516A
ADVANCED MICRO DEVICES

BACK  MIO   NC    NC    /DACK  NC    NC    NC    /CEN  GND
NC    /RD   /IORC /DS   /MWTC  /MRDC /IOWC /RW   /WR   VCC

IF (BACK) IORC  = /MIO * DS * /RW * CEN

IF (BACK) IOWC  = /MIO * DS * RW * CEN

IF (BACK) MRDC  = MIO * DS * /RW * CEN

IF (BACK) MWTC  = MIO * DS * RW * CEN

RD  = DACK * RW * BACK +
      IORC * /BACK

WR  = DACK * /RW * BACK +
      IOWC * /BACK

IF (/BACK) DS  = IORC + IOWC

IF (/BACK) RW  = IOWC
```

DESCRIPTION

THIS PAL CONVERTS MULTIBUS SIGNALS INTO Am9516A COMPATIBLE SIGNALS AND VICE VERSA. IT ALSO SUPPORTS THE 8530 IN FLYBY MODE.

MULTIBUS Control for Am9516A (AmPAL16L8)

2

```
PAL16R4                    PAL DESIGN SPECIFICATION 9516MBA
PAT 004                          JOE BRCICH 30 July 84
MULTIBUS ARBITER FOR Am9516A
ADVANCED MICRO DEVICES

/BCLK  /XACK  BRQ   /BSY  /BPRN /DS   NC    /IORC /CS   GND
/OE    /RBEN  /TBEN BACK  /CEN  /BREQ /BUSY /BPRO /WAIT VCC

IF (/BACK) TBEN  = IORC * CS

IF (/BACK) RBEN  = /IORC * CS

WAIT  = /XACK * BACK

BREQ := BRQ

BPRO  = /BRQ * BPRN

/BACK := /BUSY

BUSY := BREQ * BPRN * /BSY * /BUSY  +
        BREQ * BUSY * BPRN          +
        BREQ * BUSY

CEN := BACK
```

DESCRIPTION

/CEN DELAYS THE COMMANDS TO MEET THE MULTIBUS REQUIREMENT THAT ADDRESS AND DATA BE VALID AT LEAST 50NS PRIOR TO CONTROL ACTIVE. /IOWC WAS NOT USED SINCE USING /IORC IMPROVES HOLD TIME. THIS DESIGN DOES NOT SUPPORT THE /CBRQ FUNCTION.

MULTIBUS Arbiter for Am9516A (AmPAL16R4)

```
•                 •
•                 •
•                 •
B0 38             MOV       AL,38H      ;LOADING POINTER OF MASTER
E6 12             OUT       12H         ;MODE REGISTER
B8 07 00          MOV       AX,007H     ;LOADING MMR CODE
E7 10             OUTW      10H         ;
B0 26             MOV       AL,26H      ;LOADING POINTER OF CHAIN
E6 12             OUT       12H         ;ADDRESS REGISTER'S SEGMENT
B8 00 00          MOV       AX,0000H    ;LOADING SEGMENT OF CAR-1
E7 10             OUTW      10H         ;
B0 22             MOV       AL,22H      ;LOADING POINTER OF CHAIN
E6 12             OUT       12H         ;ADDRESS REGISTER'S OFFSET
B8 20 10          MOV       AX,1020H    ;LOADING OFFSET OF CAR-1
E7 10             OUTW      10H         ;
B0 2C             MOV       AL,2CH      ;LOADING POINTER OF COMMAND
E6 12             OUT       12H         ;REGISTER
B0 A0             MOV       AL,A0H      ;LOADING "START CHAIN" COMMAND
E6 10             OUT       10H         ;ISSUING "START CHAIN" COMMAND
•                 •
•                 •
•                 •
```

Figure 23. Initialization Program for 8086 CPU

Notes: The P/$\overline{D}$ input is connected to A1 line; $\overline{CS}$ is decoded from A7 through A4 (all O).

ADDRESS	0	2	4	6	8	A	C	E
1000	0000	1020	0000	1020	0007	0005	0006	0005
1010	0002	AAAA	0009	00A0	0004	0042	0042	0001
1020	03FF	0000	1F00	0000	1060	0010	0000	1F00
1030	0000	1080	0012	0000	FFFF	0001	0000	8020
1040	0000	1020	1111	1111	0000	FFFF	2004	0000
1050	0010	0000	0000	1020	0018	1020	2222	1007
1060	CACA	CACA	CACA	CACA	CACA	CACA	CACA	CACA
1070	CACA	CACA	CACA	CACA	CACA	CACA	CACA	CACA

Reload Word

TB000084

Figure 24. Reload Table for Chaining

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 V to +7.0 V
All Signal Voltages with Respect to V_{SS} −0.5 V to +7.0 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

Military (M) Devices
Temperature (T_C) ..... −55 to 125°C
Supply Voltage (V_{CC}) ..... 5 V ± 10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise specified)

Parameters	Description	Test Conditions		Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator		3.8	V_{CC} +0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator		−0.5	0.45	Volts
V_{IH1}	Input High Voltage	All Pins Except 2, 36, 37, 38, 47, 48		2.0	V_{CC} +0.3	Volts
V_{IH2}	Input High Voltage	Pins 2, 36, 37, 38, 47, 48		2.2	V_{CC} + 0.3	Volts
V_{IL}	Input Low Voltage			−0.5	0.8	Volts
V_{OH1}	Output High Voltage	I_{OH} = −250μA (Except Pins 1, 32, 33, 38)		2.4		Volts
V_{OH2}	Output High Voltage	I_{OH} = −200 μA, Pins 1, 32, 33, 38		2.0		Volts
V_{OL}	Output Low Voltage	I_{OL} = 3.2 mA			0.45	Volts
I_{IL}	Input Leakage	$V_{SS} \leq V_{IN} \leq V_{CC}$			±10	μA
I_{OL}	Output Leakage	$V_{SS} \leq V_{OUT} \leq V_{CC}$			±10	μA
I_{CC}	V_{CC} Supply Current	T_A = 0°C	MIL: T_C = −55°C		350	mA
		T_A = 70°C	MIL: T_C = +125°C		200	mA
CCLK	Input Capacitance (Clock)	Unmeasured pins returned to ground. f = 1 MHz over specified temperature range.			25*	pF
C_{IN}	Input Capacitance (Except Pin 46)	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.			10*	pF
C_{OUT}	Output Capacitance				15*	pF
$C_{I/O}$	Bidirectional Capacitance				20*	pF

*Guaranteed by design; not tested.

2

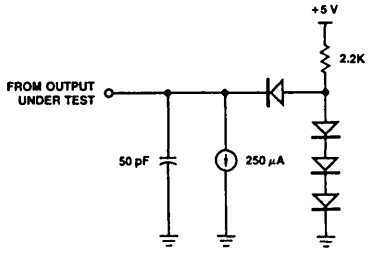
Standard Test Conditions

Commercial

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

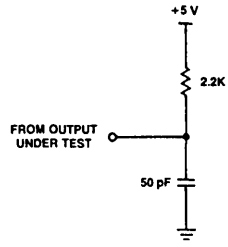
$+4.75 \leqslant V_{CC} \leqslant +5.25$ V
GND = 0 V
$0°C \leqslant T_A \leqslant +70°C$

Standard Test Load



TC002011

Open-Drain Test Load



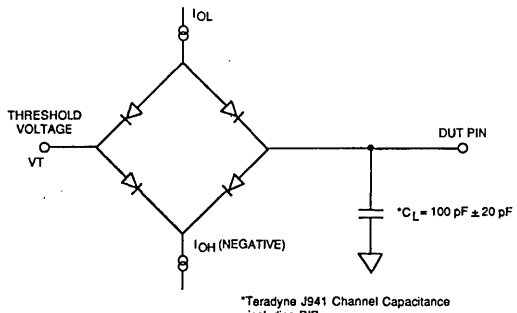
TC002021

Standard Test Conditions

Military

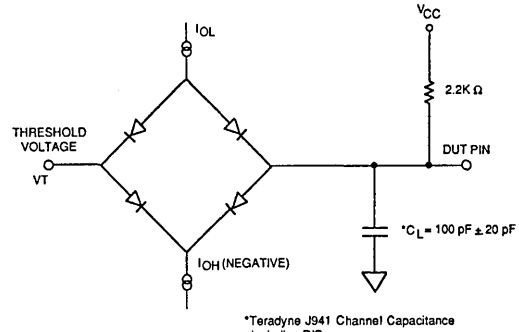
The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

$+4.75 \leqslant V_{CC} \leqslant +5.25$ V
GND = 0 V
$-55 \leqslant T_C \leqslant +125°C$



TC004391

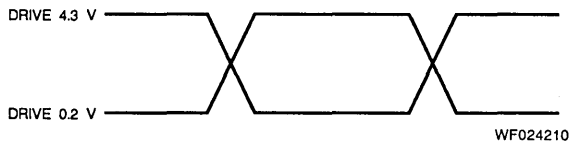
A. Standard Dynamic Load



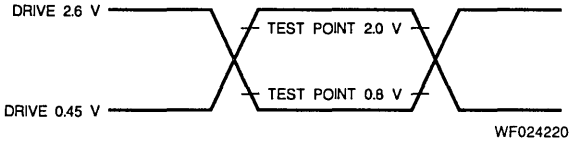
TC004381

B. Open-Drain Dynamic Load

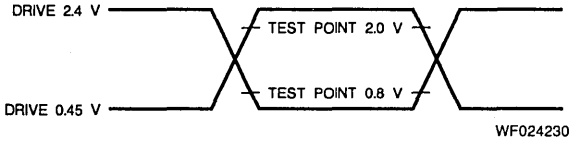
SWITCHING TEST WAVEFORMS



A. External CLOCK Generator



B. BACK, DREQ1, DREQ2, RESET, INTACK, and EOP only



C. ALL pins <u>except</u> BACK, DREQ1, DREQ2, RESET, INTACK, and EOP.

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating range unless otherwise specified
TIMING FOR UDC AS BUS MASTER

							Preliminary				
			4 MHz		6 MHz		8 MHz		10 MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
1	TcC	Clock Cycle Time	250	2000	165	2000	125	2000	100	2000	ns
2	TwCh	Clock Width (HIGH)	105	1000	70	1000	55		45		ns
3	TwCl	Clock Width (LOW)	105		70		55		45		ns
4	TfC	Clock Fall Time		20		10		5		5	ns
5	TrC	Clock Rise Time		20		15		10		5	ns
6	TdC(AUv)	Clock RE to Upper Address (A_{16}-A_{23}) Valid Delay		90		80		60		50	ns
7	ThC(AUv)	Clock RE to Upper Address Valid Hold Time	5		5		5		5		ns
8	TdC(ST)	Clock RE to R/$\overline{W}$ and B/$\overline{W}$ Valid Delay		110		90		60		50	ns
9	TdC(A)	Clock RE to Lower Address (A_0-A_{15}) Valid Delay		90		90		60		50	ns
10	TdC(Az)	Clock RE to Lower Address (A_0-A_{15}) Float Delay		60		60		50		40	ns
11	TdC(ALr)	Clock RE to ALE RE Delay		70		60		50		30	ns
12	TdC(AL)	Clock FE to ALE FE Delay		70		60		50		40	ns
13	TdC(DS)	Clock RE to $\overline{DS}$ (Read) FE Delay		60		60		50		40	ns
14	TdC(DSf)	Clock FE to $\overline{DS}$ (Write) FE Delay		60		60		50		45	ns
15	TdC(DSr)	Clock FE to $\overline{DS}$ RE Delay		60		60		50		40	ns
16	TdC(DO)	Clock RE to Data Out Valid Delay		90		90		65		60	ns
17	TsDI(C)	Data in to Clock FE Set-up Time	20		15		10		10		ns
18	TdA(AL)	Address Valid to ALE FE Delay	50		35		20		20		ns
19	ThAL(A)	ALE FE to Lower Address Valid Hold Time	60		40		30		30		ns
20	TwAL	ALE Width (HIGH)	80		60		45		40		ns
21	TdAz(DS)	Lower Address Float to $\overline{DS}$ LOW Delay	0		0		0		0		ns
22	TdAL(DS)	ALE FE to $\overline{DS}$ (Read) FE Delay	75		35		35		35		ns
23	TdAL(DI)	ALE FE to Data in Required Valid Delay		300		215		190		150	ns
24	TdA(DI)	Address Valid to Data in Required Valid Delay		410		305		240		190	ns
25	TdDS(A)	$\overline{DS}$ RE to Address Active Delay	80		45		30		20		ns
26	TdDS(AI)	$\overline{DS}$ RE 10 ALE RL Delay	75		40		40		35		ns
27	TdA(DS)	Address Valid to $\overline{DS}$ (Read) FE Delay	160		110		90		70		ns
28	TdDO(DSr)	Data Out Valid to $\overline{DS}$ RE Delay	230		150		125		80		ns
29	TdDO(DSf)	Data Out Valid to $\overline{DS}$ FE Delay	55		35		20		15		ns
30	ThDS(DO)	$\overline{DS}$ RE to Data Out Valid Hold Time	85		45		40		25		ns
31	TdDS(DI)	$\overline{DS}$ (Read) FE to Data in Required Valid Delay		205		155		125		100	ns
33	ThDI(DS)	$\overline{DS}$ RE to Data in Hold Time	0		0		0		0		ns
34	TwDSmw	$\overline{DS}$ (Write) Width (LOW)	185		110		105		80		ns
35	TwDSmr	$\overline{DS}$ (Read) Width (LOW)	275		220		160		130		ns
36	TdC(RBr)	Clock FE to $\overline{RBEN}$ RE Delay*		70		65		50		30	ns
37	ThDS(ST)	$\overline{DS}$ RE to B/$\overline{W}$, N/$\overline{S}$, R/$\overline{W}$ and M/$\overline{IO}$ Valid Hold Time	70		45		40		25		ns
38	TdC(TRf)	Clock RE to $\overline{TBEN}$ or $\overline{RBEN}$ FE Delay		60		60		50		35	ns
39	TdC(TRr)	Clock RE to $\overline{TBEN}$ RE Delay		60		60		45		45	ns
40	TdC(ST)	Clock RE to M/$\overline{IO}$ and N/$\overline{S}$ Valid Delay		90		75		65		50	ns
41	TdS(AL)	R/$\overline{W}$, M/$\overline{IO}$, B/$\overline{W}$ and N/$\overline{S}$ Valid to ALE FE Delay	60		35		20		20		ns
42	TsWT(C)	$\overline{WAIT}$ to Clock FF Set-up Time	20		20		10		10		ns
43	ThWT(C)	$\overline{WAIT}$ to Clock FE Hold Time	20		20		35		35		ns
44	TwDRQ	$\overline{DREQ}$ Pulse Width (Single Transfer Mode)	20		20		20		20		ns
45	TsDRQ(C)	$\overline{DREQ}$ Valid to Clock RE Set-up Time	60		50		30		20		ns
46	ThDRQ(C)	Clock RE to $\overline{DREQ}$ Valid Hold Time	20		20		20		20		ns
47	TdC(INTf)	Clock FE to $\overline{INT}$ FE Delay		150		150		105		105	ns

*These must not occur simultaneously.

Note: RE = rising edge
FE = falling edge

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for APL Products, Subgroups 9, 10, 11 are tested unless otherwise specified)

TIMING FOR UDC AS BUS MASTER

					Preliminary		
			4 MHz		6 MHz		
Number	Parameters	Description	Min.	Max.	Min.	Max.	Unit
1	TcC	Clock Cycle Time	250	2000	165	2000	ns
2	TwCh	Clock Width (HIGH)	105	1000	70	1000	ns
3	TwCl	Clock Width (LOW)	105		70		ns
4	TfC	Clock Fall Time		20		10	ns
5	TrC	Clock Rise Time		20		15	ns
6	TdC(AUv)	Clock RE to Upper Address (A_{16}-A_{23}) Valid Delay		90		80	ns
7	ThC(AUv)	Clock RE to Upper Address Valid Hold Time	5		5		ns
8	TdC(ST)	Clock RE to R/$\overline{W}$ and B/$\overline{W}$ Valid Delay		110		90	ns
9	TdC(A)	Clock RE to Lower Address (A_0-A_{15}) Valid Delay		90		90	ns
10	TdC(Az)	Clock RE to Lower Address (A_0-A_{15}) Float Delay		60		60	ns
11	TdC(ALr)	Clock RE to ALE RE Delay		70		60	ns
12	TdC(AL)	Clock FE to ALE FE Delay		70		60	ns
13	TdC(DS)	Clock RE to $\overline{DS}$ (Read) FE Delay		60		60	ns
14	TdC(DSf)	Clock FE to $\overline{DS}$ (Write) FE Delay		60		60	ns
15	TdC(DSr)	Clock FE to $\overline{DS}$ RE Delay		60		60	ns
16	TdC(DO)	Clock RE to Data Out Valid Delay		90		90	ns
17	TsDI(C)	Data in to Clock FE Setup Time	20		15		ns
18	TdA(AL)	Address Valid to ALE FE Delay	50		35		ns
19	ThAL(A)	ALE FE to Lower Address Valid Hold Time	60		40		ns
20	TwAL	ALE Width (HIGH)	80		60		ns
21	TdAz(DS)	Lower Address Float to $\overline{DS}$ LOW Delay	0		0		ns
22	TdAL(DS)	ALE FE to $\overline{DS}$ (Read) FE Delay	75		35		ns
23	TdAL(DI)	ALE FE to Data in Required Valid Delay		300		215	ns
24	TdA(DI)	Address Valid to Data in Required Valid Delay		410		305	ns
25	TdDS(A)	$\overline{DS}$ RE to Address Active Delay	80		45		ns
26	TdDS(AI)	$\overline{DS}$ RE to ALE RE Delay	75		40		ns
27	TdA(DS)	Address Valid to $\overline{DS}$ (Read) FE Delay	160		110		ns
28	TdDO(DSr)	Data Out Valid to $\overline{DS}$ RE Delay	230		150		ns
29	TdDO(DSf)	Data Out Valid to $\overline{DS}$ FE Delay	55		35		ns
30	ThDS(DO)	$\overline{DS}$ RE to Data Out Valid Hold Time	85		45		ns
31	TdDS(DI)	$\overline{DS}$ (Read) FE to Data in Required Valid Delay		205		155	ns
33	ThDI(DS)	$\overline{DS}$ RE to Data in Hold Time	0		0		ns
34	TwDSmw	$\overline{DS}$ (Write) Width (LOW)	185		110		ns
35	TwDSmr	$\overline{DS}$ (Read) Width (LOW)	275		220		ns
36	TdC(RBr)	Clock FE to $\overline{RBEN}$ RE Delay*		70		65	ns
37	ThDS(ST)	$\overline{DS}$ RE to B/$\overline{W}$, N/$\overline{S}$, R/$\overline{W}$ and M/$\overline{IO}$ Valid Hold Time	70		45		ns
38	TdC(TRf)	Clock RE to $\overline{TBEN}$ or $\overline{RBEN}$ FE Delay		60		60	ns
39	TdC(TRr)	Clock RE to $\overline{TBEN}$ RE Delay		60		60	ns
40	TdC(ST)	Clock RE to M/$\overline{IO}$ and N/$\overline{S}$ Valid Delay		90		75	ns
41	TdS(AL)	R/$\overline{W}$, M/$\overline{IO}$, B/$\overline{W}$ and N/$\overline{S}$ Valid to ALE FE Delay	60		35		ns
42	TsWT(C)	$\overline{WAIT}$ to Clock FF Setup Time	20		20		ns
43	ThWT(C)	$\overline{WAIT}$ to Clock FE Hold Time	20		20		ns
44	TwDRQ	$\overline{DREQ}$ Pulse Width (Single Transfer Mode)	20		20		ns
45	TsDRQ(C)	$\overline{DREQ}$ Valid to Clock RE Setup Time	60		50		ns
46	ThDRQ(C)	Clock RE to $\overline{DREQ}$ Valid Hold Time	20		20		ns
47	TdC(INTf)	Clock FE to $\overline{INT}$ FE Delay		150		150	ns

*These must not occur simultaneously.
Note: RE = rising edge
FE = falling edge

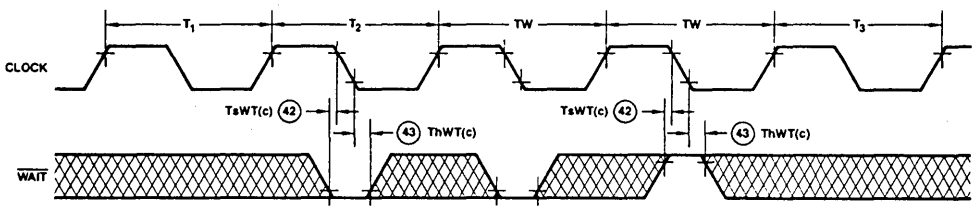
Am9516A CLOCK-CYCLE-TIME-DEPENDENT CHARACTERISTICS

The parameters listed below are also shown in the Switching specification. However, they are dependent on the actual values of the clock periods. The equations below define that dependence so that the exact limit for these parameters may be determined for any given system in relation to its specific clock characteristics.

Number	Parameters	Derivation
18	TdA (AL)	0.5TcC – #9 + (#12 – tr)
19	ThAL (A)	0.5TcC – #12 (ALE FE @ 0.8 V) + #10
21	TdAz (DS)	#13 – #10
22	TdAL (DS)	0.5TcC – #12 + #13
23	TdAL (DI)	2TcC – #12 – #17
24	TdA (DI)	2.5TcC – #9 – #17
25	TdDS (A)	0.5TcC – #15 + #9
26	TsDS (AL)	0.5TcC – #15 + #11 (ALE RE)
27	TdA (DS)	TcC – #9 + #13
28	TdDO (DSr)	1.5TcC – #16 + #15
29	TdDO (DSf)	0.5TcC – #16 + #14
30	ThDS (DO)	0.5TcC – #15 + #32
31	TdDS (DI)	1.5TcC – #13 – #17
34	TwDSmw	TcC – #14 + #15
35	TwDSmr	1.5TcC – #13 + #15
37	ThDS (ST)	0.5TcC – #15 + (#40 – tr)
41	TdS (AL)	0.5TcC – #40 + (#12 – tr)

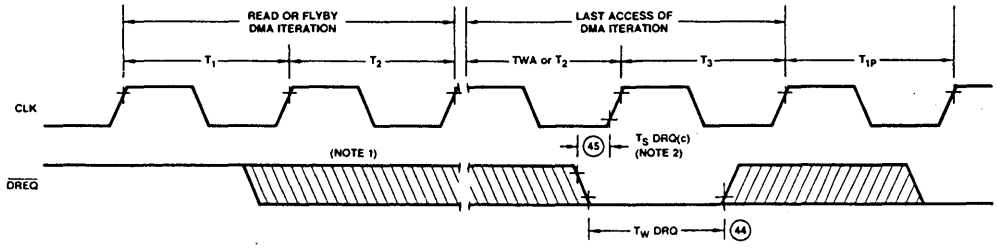
NOTE: tr (nominal) = 10ns
#32 CLCK RE to Data Out Not Valid Delay = 20ns (4 and 6 MHz)

WAIT Timing



WF007680

Sampling DREQ During Single Transfer DMA Operations



WF007670

Notes: 1. HIGH-to-LOW DREQ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of a read or flyby DMA iteration.

2. A HIGH-to-LOW DREQ transition must meet the conditions in Note 1 and must occur TsDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. DREQ may go HIGH before TsDRQ(c) if it has met the TwDRQ parameter.

3. Flyby and Search transactions have only a single access; parameter TsDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range unless otherwise specified
UDC AS BUS SLAVE BUS EXCHANGE

							Preliminary				
			4 MHz		6 MHz		8 MHz		10 MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
61	TdIN(DO)	$\overline{INTACK}$ FE to Data Output Valid Delay		135		135		120		110	ns
62	TdIN(DOz)	$\overline{INTACK}$ RE to Data Output Float Delay		80		80		45		35	ns
63	TdDS(DO)	$\overline{DS}$ FE (IOR) to Data Output Driven Delay		135*		135		120		110	ns
64	TdDS(DOz)	$\overline{DS}$ RE (IOR) to Data Output Float Delay		80		80		45		40	ns
65	TsDI(DS)	Data Valid to $\overline{DS}$ RE (IOW) Set-up Time	40		40		40		35		ns
66	ThDS(DI)	$\overline{DS}$ RE (IOW) to Data Valid Hold Time	40		30		0		0		ns
67	TwDS	$\overline{DS}$ Low Width	150*		150*		125		100		ns
68	TwIN	$\overline{INTACK}$ Low Width	150		150		125		100		ns
69	ThDS(CS)	$\overline{DS}$ RE to $\overline{CS}$ Valid Hold Time	20		20		15		10		ns
70	ThDS(PD)	$\overline{DS}$ RE to P/$\overline{D}$ Valid Hold Time	20		20		15		10		ns
71	TsPD(DS)	P/$\overline{D}$ Valid to $\overline{DS}$ FE Set-up Time (IOR)	10		10		10		10		ns
		P/$\overline{D}$ Valid to $\overline{DS}$ FE Set-up Time (IOW)	50		50		40		30		
72	TsCS(DS)	$\overline{CS}$ Valid to $\overline{DS}$ FE Set-up Time	30		30		20		10		ns
73	TrDS	$\overline{DS}$ RE to $\overline{DS}$ FE Recovery Time (for Commands Only)	4TcC		4TcC		4Tcc		4Tcc		ns
74	TwRST	$\overline{RESET}$ Low Width	3TcC		3TcC		3Tcc		3Tcc		ns
75	TdC(BRQf)	Clock RE to BREQ RE Delay		165		150		125		100	ns
76	TdC(BRQr)	Clock FE to BREQ FE Delay		150		150		125		100	ns
77	TdBRQ(CTRz)	BREQ FE to Control Bus Float Delay		140		140		100		60	ns
78	TdBRQ(ADz)	BREQ FE to AD Bus Float Delay		140		140		100		60	ns
79	TdBRQ(BAK)	BREQ RE to BACK RE Required Delay	0		0		0		0		ns
80	TsBAK(C)	BACK Valid to Clock RE Set-up Time	50		45		30		20		ns
81	TdRES(ADz)	(Reset) FE to A and AD Buses Float Delay		135		135		125		100	ns
82	TdRES(CTRz)	(Reset) FE to Control Bus Float Delay		100		100		100		75	ns
83	TdRES(DSz)	(Reset) FE to $\overline{DS}$ Float Delay		90		90		80		60	ns
84	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Set-up Time (IOW)	2		2		2		2		ns
85	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOW)	-10		-10		-10		-10		ns
86	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Set-up Time (IOR)	20		20		15		15		ns
87	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOR)	20		20		15		15		ns

*2000ns for slow readable registers (worst case)
Note: RE = rising edge
FE = falling edge

SWITCHING CHARACTERISTICS (continued)
UDC-PERIPHERAL INTERFACE

							Preliminary				
			4 MHz		6 MHz		8 MHz		10 MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
90	TCHDL	Clock RE to Pulsed $\overline{DACK}$ FE Delay (Flyby Transactions Only)		100		85		50		40	ns
91	TCHDH	Clock RE to Pulsed $\overline{DACK}$ RE Delay (To Flyby Transactions Only)		100		85		50		40	ns
92	TDSK	$\overline{DS}$ RE to Pulsed $\overline{DACK}$ RE Delay (FROM Flyby Transactions Only)	10		10		10		10		ns
93	TDAD	Clock RE to Level $\overline{DACK}$ Valid Delay		100		85		60		50	ns
94	TDAH	Clock FE to Level $\overline{DACK}$ Valid Hold Time		100		85		60		50	ns
95	TEIDL	Clock FE to Internal $\overline{EOP}$ LOW Delay		110		90		80		70	ns
96	TEIDH	Clock FE to Internal $\overline{EOP}$ RE Delay		110		90		80		70	ns
97	TES	External $\overline{EOP}$ Valid to Clock RE Set-up Time During Operation	10		10		10		10		ns
98	TEW	External $\overline{EOP}$ Pulse Width Required During Operation	20		20		20		20		ns
99	TES(BH)	External $\overline{EOP}$ Valid to Clock FE Set-up Time During Bus Hold	10		10		10		10		ns
100	TEW(BH)	External $\overline{EOP}$ Pulse Width Required During Bus Hold	20		20		20		20		ns

Note: RE = rising edge
FE = falling edge

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for APL Products, Subgroups 9, 10, 11 are tested unless otherwise specified)

UDC AS BUS SLAVE BUS EXCHANGE

					Preliminary		
			4 MHz		6 MHz		
No.	Parameter Symbol	Parameter Description	Min.	Max.	Min.	Max.	Unit
61	TdIN(DO)	$\overline{INTACK}$ FE to Data Output Valid Delay		135		135	ns
62	TdIN(DOz)	$\overline{INTACK}$ RE to Data Output Float Delay		80		80	ns
63	TdDS(DO)	$\overline{DS}$ FE (IOR) to Data Output Driven Delay		135*		135	ns
64	TdDS(DOz)	$\overline{DS}$ RE (IOR) to Data Output Float Delay		80		80	ns
65	TsDI(DS)	Data Valid to $\overline{DS}$ RE (IOW) Setup Time	40		40		ns
66	ThDS(DI)	$\overline{DS}$ RE (IOW) to Data Valid Hold Time	40		30		ns
67	TwDS	$\overline{DS}$ LOW Width	150*		150*		ns
68	TwIN	$\overline{INTACK}$ LOW Width	150		150		ns
69	ThDS(CS)	$\overline{DS}$ RE to $\overline{CS}$ Valid Hold Time	20		20		ns
70	ThDS(PD)	$\overline{DS}$ RE to P/$\overline{D}$ Valid Hold Time	20		20		ns
71	TsPD(DS)	P/$\overline{D}$ Valid to $\overline{DS}$ FE Setup Time (IOR)	10		10		ns
		P/$\overline{D}$ Valid to $\overline{DS}$ FE Setup Time (IOW)	50		50		
72	TsCS(DS)	$\overline{CS}$ Valid to $\overline{DS}$ FE Setup Time	30		30		ns
73	TrDS	$\overline{DS}$ RE to $\overline{DS}$ FE Recovery Time (for Commands Only)	4TcC		4TcC		ns
74	TwRST	$\overline{RESET}$ LOW Width	3TcC		3TcC		ns
75	TdC(BRQf)	Clock RE to BREQ RE Delay		165		150	ns
76	TdC(BRQr)	Clock FE to BREQ FE Delay		150		150	ns
77	TdBRQ(CTRz)	BREQ FE to Control Bus Float Delay		140		140	ns
78	TdBRQ(ADz)	BREQ FE to AD Bus Float Delay		140		140	ns
79	TdBRQ(BAK)	BREQ RE to BACK RE Required Delay	0		0		ns
80	TsBAK(C)	BACK Valid to Clock RE Setup Time	50		45		ns
81	TdRES(ADz)	RESET FE to A and AD Buses Float Delay		135**		135	ns
82	TdRES(CTRz)	RESET FE to Control Bus Float Delay		100**		100	ns
83	TdRES(DSz)	RESET FE to $\overline{DS}$ Float Delay		90**		90	ns
84	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Setup Time (IOW)	2		2		ns
85	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOW)	−10		−10		ns
86	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Setup Time (IOR)	20		20		ns
87	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOR)	20		20		ns

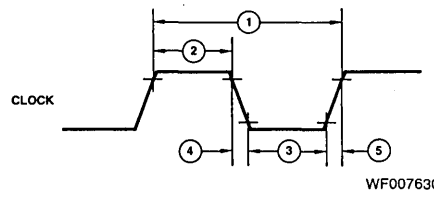
*2000 ns for slow readable registers (worst case)
**Guaranted but not tested.
Note: RE = rising edge
FE = falling edge

UDC-PERIPHERAL INTERFACE

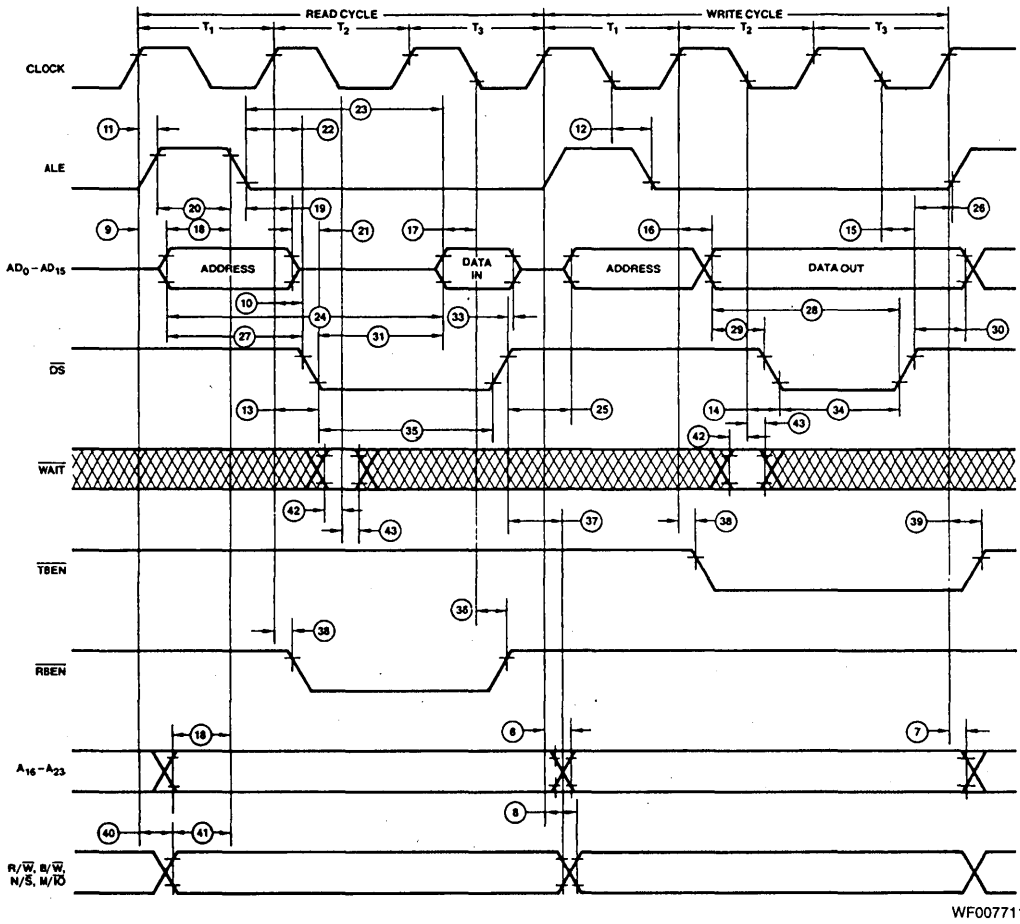
					Preliminary		
			4 MHz		6 MHz		
No.	Parameter Symbol	Parameter Description	Min.	Max.	Min.	Max.	Unit
90	TCHDL	Clock RE to Pulsed $\overline{DACK}$ FE Delay (Flyby Transactions Only)		100		85	ns
91	TCHDH	Clock RE to Pulsed $\overline{DACK}$ RE Delay (To Flyby Transactions Only)		100		85	ns
92	TDSK	$\overline{DS}$ RE to Pulsed $\overline{DACK}$ RE Delay (FROM Flyby Transactions Only)	10		10		ns
93	TDAD	Clock RE to Level $\overline{DACK}$ Valid Delay		100		85	ns
94	TDAH	Clock FE to Level $\overline{DACK}$ Valid Hold Time		100		85	ns
95	TEIDL	Clock FE to Internal $\overline{EOP}$ LOW Delay		110		90	ns
96	TEIDH	Clock FE to Internal $\overline{EOP}$ RE Delay		110		90	ns
97	TES	External $\overline{EOP}$ Valid to Clock RE Setup Time During Operation	10		10		ns
98	TEW	External $\overline{EOP}$ Pulse Width Required During Operation	20		20		ns
99	TES(BH)	External $\overline{EOP}$ Valid to Clock RE Setup Time During Bus Hold	10		10		ns
100	TEW(BH)	External $\overline{EOP}$ Pulse Width Required During Bus Hold	20		20		ns

Note: RE = rising edge
FE = falling edge

AC Timing when UDC is a Bus Master

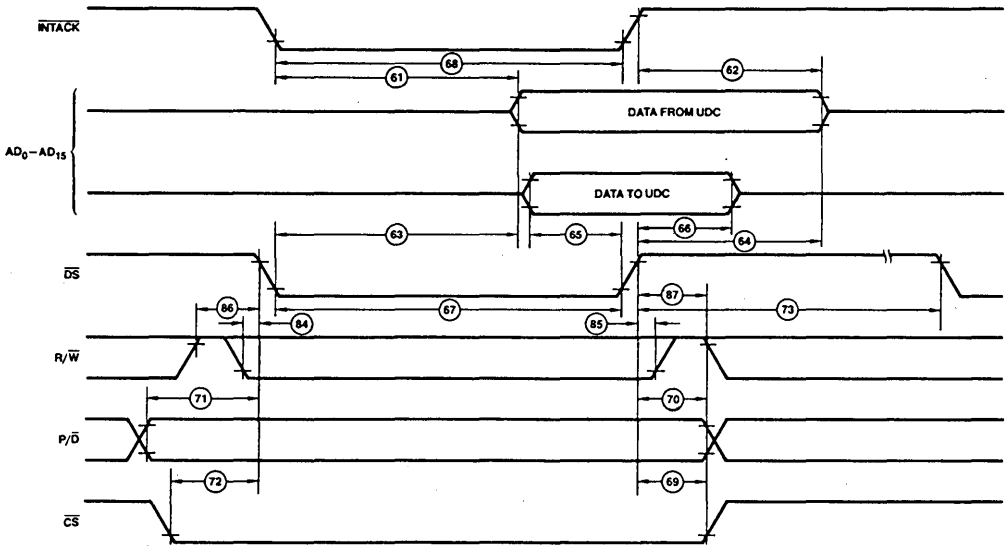


WF007630



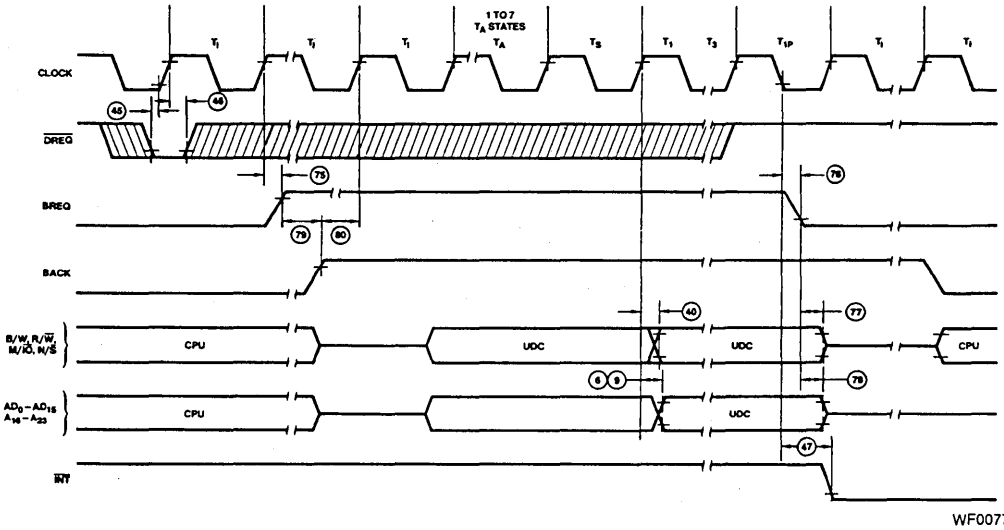
WF007711

AC Timing when UDC is a Bus Slave



WF007720

Bus Exchange Timing



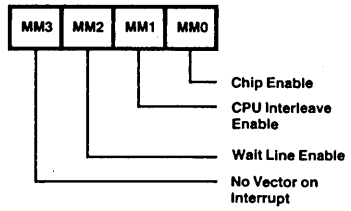
WF007730

APPENDIX A
UDC REGISTER SUMMARY

Master Mode Register

	A7							A0
Address	X	0	1	1	1	0	0	X

Fast Readable
Writable



DF003470

Miscellaneous Registers

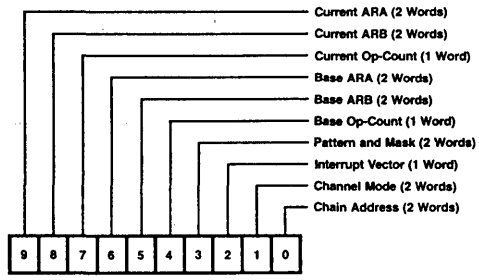
Address	A7							A0		
	X	0	1	1	0	0	1	X	Current Operation Count	CH1
	X	0	1	1	0	0	0	X	Current Operation Count	CH2
	X	0	1	1	0	1	1	X	Base Operation Count	CH1
	X	0	1	1	0	1	0	X	Base Operation Count	CH2
	X	1	0	0	1	0	1	X	Pattern	CH1
	X	1	0	0	1	0	0	X	Pattern	CH2
	X	1	0	0	1	1	1	X	Mask	CH1
	X	1	0	0	1	1	0	X	Mask	CH2

Chain Loadable
Writable
Pattern and Mask – Slow Readable
Operation Count – Fast Readable

D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Chain Control Register

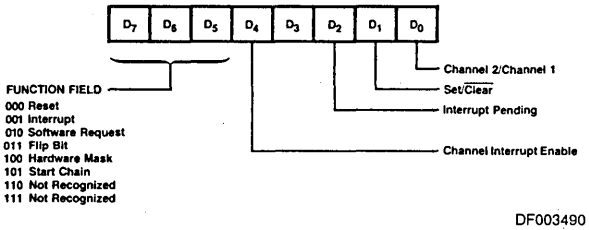
Chain Loadable Only



DF003480

Command Register

	A7							A0	
Address	X	0	1	0	1	1	1	X	CH1
Writable Only	X	0	1	0	1	1	0	X	CH2

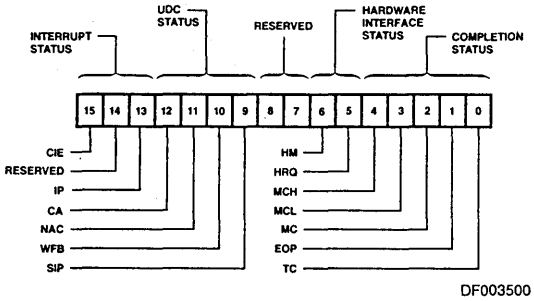


DF003490

Status Register

	A7							A0	
Address	X	0	1	0	1	1	1	X	CH1
	X	0	1	0	1	1	0	X	CH2

Fast Readable



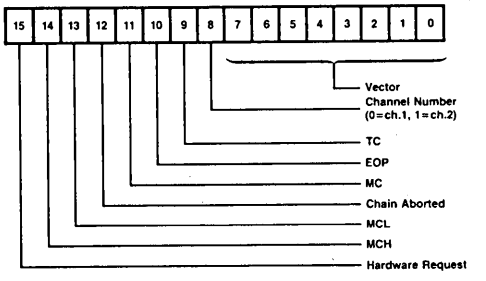
DF003500

2

Interrupt Save Register

	A7						A0	
Address	X	0	1	0	1	0	1	X	CH1
	X	0	1	0	1	0	0	X	CH2

Fast Readable

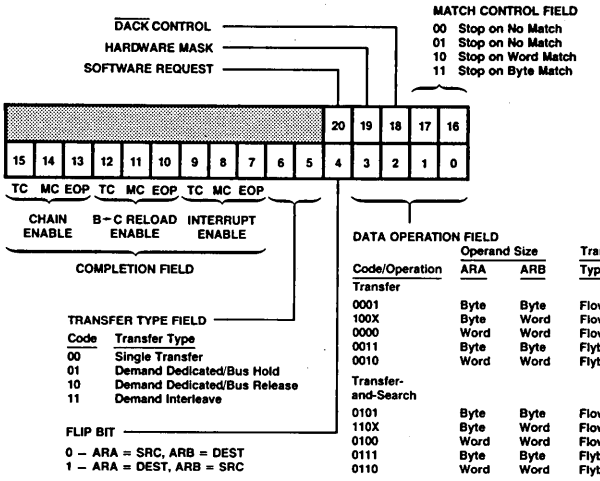


DF003510

Channel Mode Register

A7							A0	
X	1	0	1	0	1	1	X	High CH1
X	1	0	1	0	1	0	X	High CH2
X	1	0	1	0	0	1	X	Low CH1
X	1	0	1	0	0	0	X	Low CH2

Chain Loadable
Writable (Lower 16 bits)
Slow Readable

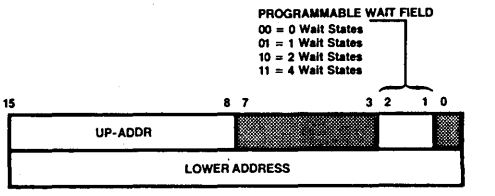


Code/Operation	Operand Size ARA	ARB	Transaction Type
Transfer			
0001	Byte	Byte	Flowthru
100X	Byte	Word	Flowthru
0000	Word	Word	Flowthru
0011	Byte	Byte	Flyby
0010	Word	Word	Flyby
Transfer-and-Search			
0101	Byte	Byte	Flowthru
110X	Byte	Word	Flowthru
0100	Word	Word	Flowthru
0111	Byte	Byte	Flyby
0110	Word	Word	Flyby
Search			
1111	Byte	Byte	N/A
1110	Word	Word	N/A
101X	Illegal		

DF003520

Chain Address Register

	A7						A0	
Address	X	0	1	0	0	1	1	X	Up-Addr CH1
	X	0	1	0	0	1	0	X	Up-Addr CH2
	X	0	1	0	0	0	1	X	Low-Addr CH1
	X	0	1	0	0	0	0	X	Low-Addr CH2

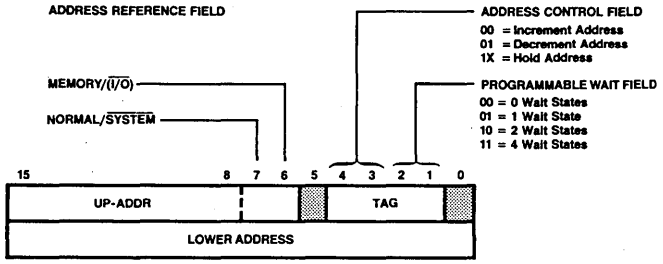


DF003530

Address Registers

	A7						A0	
Address	X	0	0	1	1	0	1	X	Current ARA Up-Addr/Tag CH1
	X	0	0	1	1	0	0	X	Current ARA Up-Addr/Tag CH2
	X	0	0	0	1	0	1	X	Current ARA Low-Addr CH1
	X	0	0	0	1	0	0	X	Current ARA Low-Addr CH2
	X	0	0	1	0	0	1	X	Current ARB Up-Addr/Tag CH1
	X	0	0	1	0	0	0	X	Current ARB Up-Addr/Tag CH2
	X	0	0	0	0	0	1	X	Current ARB Low-Addr CH1
	X	0	0	0	0	0	0	X	Current ARB Low-Addr CH2
	X	0	0	1	1	1	1	X	Base ARA Up-Addr/Tag CH1
	X	0	0	1	1	1	0	X	Base ARA Up-Addr/Tag CH2
	X	0	0	0	1	1	1	X	Base ARA Low-Addr CH1
	X	0	0	0	1	1	0	X	Base ARA Low-Addr CH2
	X	0	0	1	0	1	1	X	Base ARB Up-Addr/Tag CH1
	X	0	0	1	0	1	0	X	Base ARB Up-Addr/Tag CH2
	X	0	0	0	0	1	1	X	Base ARB Low-Addr CH1
	X	0	0	0	0	1	0	X	Base ARB Low-Addr CH2

Chain Loadable
Fast Readable and Writable



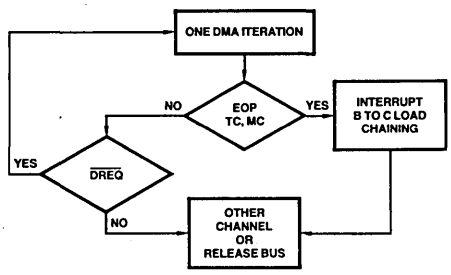
DF003540

APPENDIX B

Flow Charts of DMA Operations:

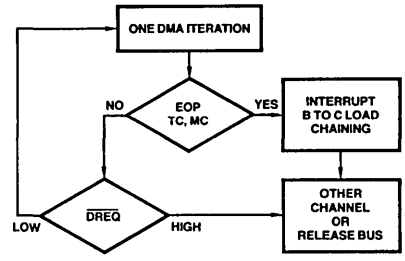
Figure B1 shows the basic DMA operations with software or hardware request. The Demand Interleave operations are shown in Figure B2.

(a) Single Operation



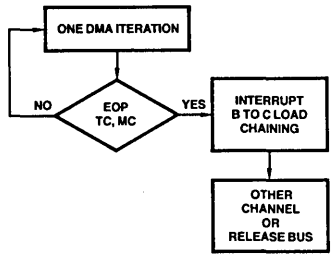
PF001250

(c) Demand Dedicated with Bus Release (Hardware Request)



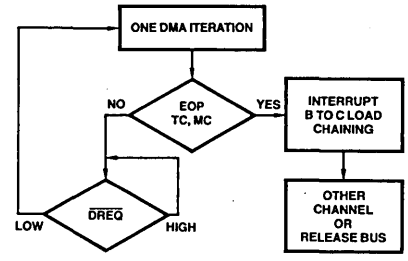
PF001260

(b) Demand Operation when Software Requesting



PF001270

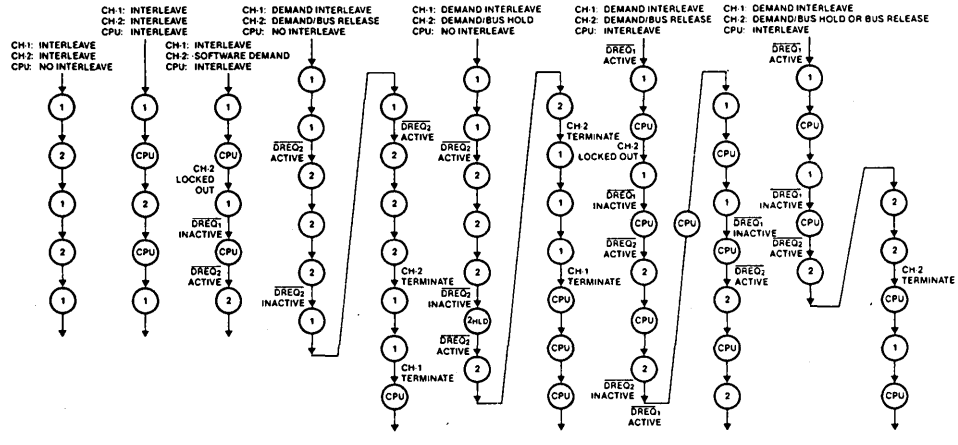
(d) Demand Dedicated with Bus Hold (Hardware Request)



PF001280

Figure B1. Basic DMA Operations of Am9516A UDC

NT**Multibus is a trademark of Intel Corp.

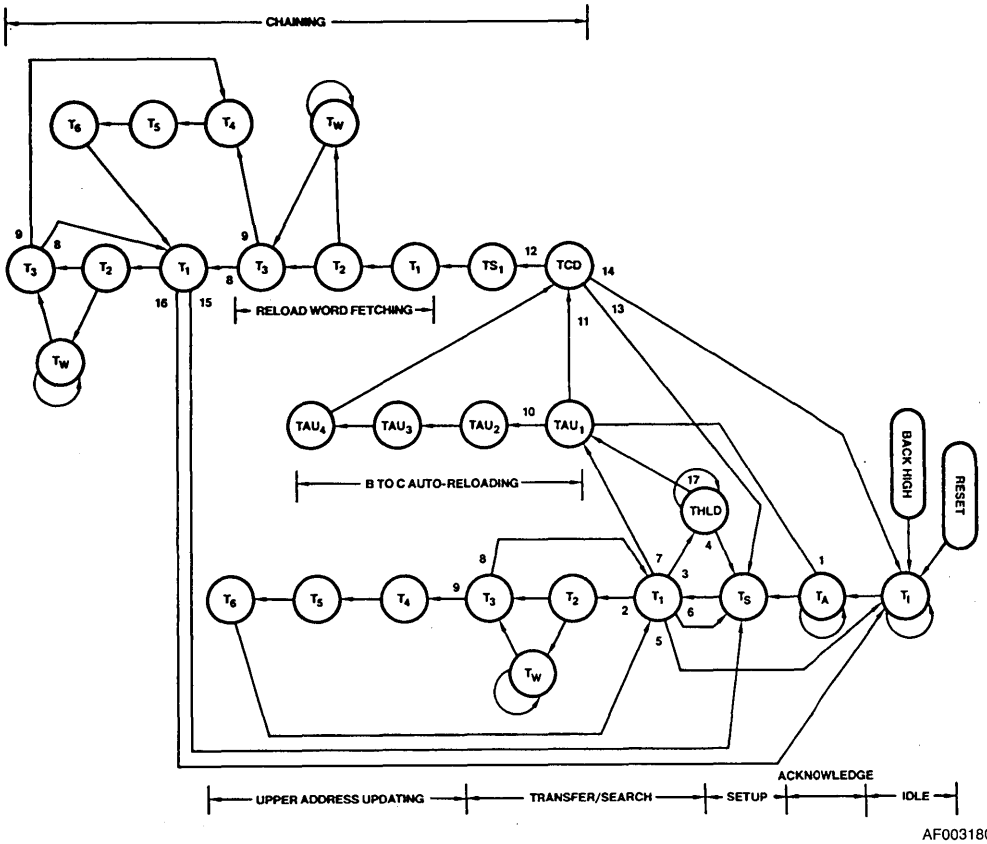


PF001300

Figure B2. Demand Interleave Operations of Am9516A UDC

APPENDIX C

Am9516A STATE DIAGRAM



AF003180

Am9516A INTERNAL OPERATION ROUTINES

1. "Start Chain" command issued or start updating routine* after an interrupt has been served.**
2. Normal DMA operation.
3. Demand with Bus hold while DREQ is inactive.
4. DREQ is active while bus held.
5. Single transfer, CPU interleave enabled, or demand with bus release while current DREQ is inactive and no DMA request is pending.
6. Single Transfer or Demand/Bus release while current DREQ is inactive, but the other DMA request is pending.
7. TC, MC or EOP termination occurs.
8. One DMA or chain transaction is done and the upper address is not changed.
9. One DMA or chain transaction is done and the upper address is changed.
10. Base-to-current auto-reloading is enabled.
11. Base-to-current auto-reloading is disabled.
12. Chaining is enabled.
13. Chaining is disabled and another DMA request is pending.
14. Chaining is disabled and no DMA request is pending.
15. Chaining ends and another DMA request is pending.
16. Chaining ends and no DMA request is pending.
17. EOP termination of Bus Hold.

*Updating routine includes base-to-current auto-reloading and chaining.

**When a second interrupt is to be issued before the first interrupt is acknowledged, the SIP bit of a Status register is set and the channel relinquishes the bus until the first interrupt has been served. If the channel was to perform the updating routine, once the SIP bit is cleared, DTC will reacquire the bus and perform the appropriate operation (i.e., 1).

Am9519A

Universal Interrupt Controller

FINAL

DISTINCTIVE CHARACTERISTICS

- Eight individually maskable interrupt inputs reduce CPU overhead
- Unlimited interrupt channel expansion with no extra hardware
- Programmable 1-byte to 4-byte response provides vector address and message protocol for 8-bit CPUs
- Rotating and fixed priority resolution logic
- Software interrupt request capability
- Common vector and polled mode options
- Automatic hardware clear of in-service interrupts reduces software overhead
- Polarity control of interrupt inputs and outputs
- Reset minimizes software initialization by automatically generating CALL to location zero
- SMD/DESC qualified

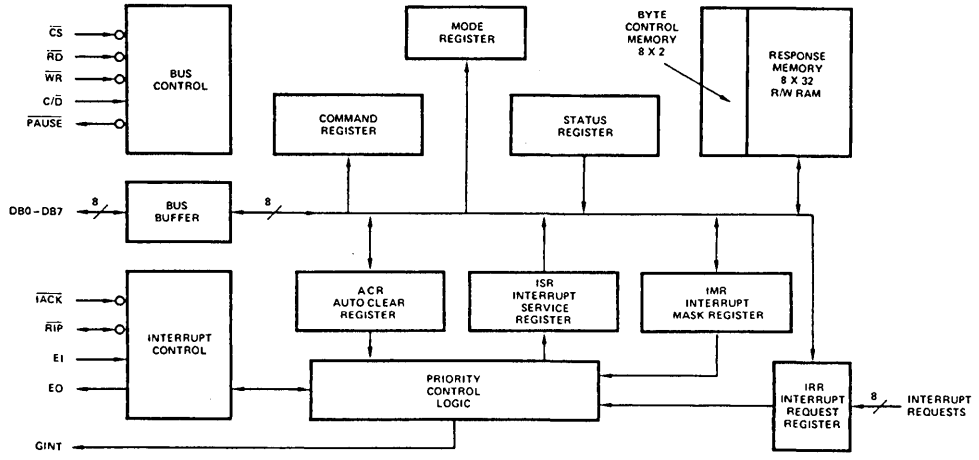
GENERAL DESCRIPTION

The Am9519A Universal Interrupt Controller is a processor support circuit that provides a powerful interrupt structure to increase the efficiency and versatility of microcomputer-based systems. A single Am9519A manages up to eight maskable interrupt request inputs, resolves priorities and supplies up to four bytes of fully programmable response for each interrupt. It uses a simple expansion structure that allows many units to be cascaded for control of large numbers of interrupts. Several programmable control features are provided to enhance system flexibility and optimization.

The Universal Interrupt Controller is designed with a general purpose interface to facilitate its use with a wide range of digital systems, including most popular 8-bit microprocessors. Since the response bytes are fully programmable, any instruction or vectoning protocol appropriate for the host processor may be used.

When the Am9519A controller receives an unmasked interrupt request, it issues a Group Interrupt output to the CPU. When the interrupt is acknowledged, the controller outputs the one-to-four byte response associated with the highest priority unmasked interrupt request. The ability of the CPU to set interrupt requests under software control permits hardware prioritization of software tasks and aids system diagnostic and maintenance procedures.

BLOCK DIAGRAM

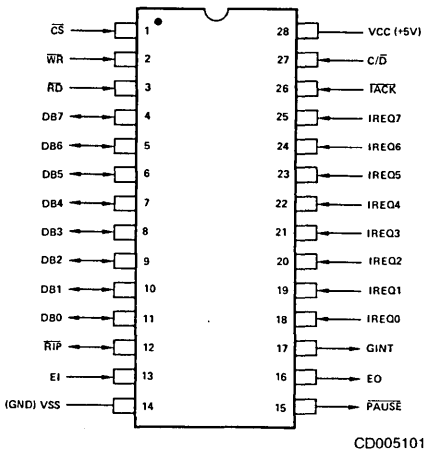


BD003280

Publication # 00147 Rev. D Amendment /0
Issue Date: May 1989

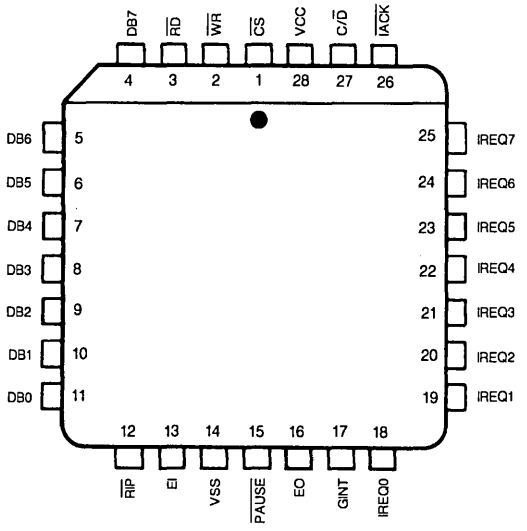
CONNECTION DIAGRAMS
Top View

DIPs



CD005101

PLCC

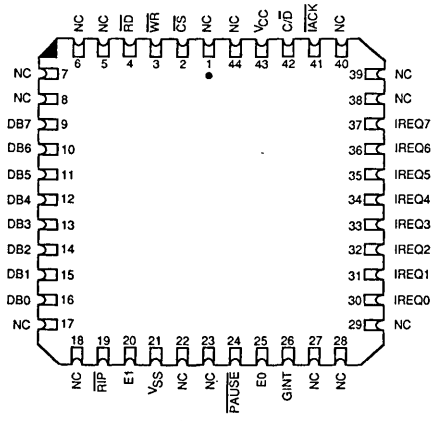


CD010783

Note: Pin 1 is marked for orientation.

LCC

(Military only)

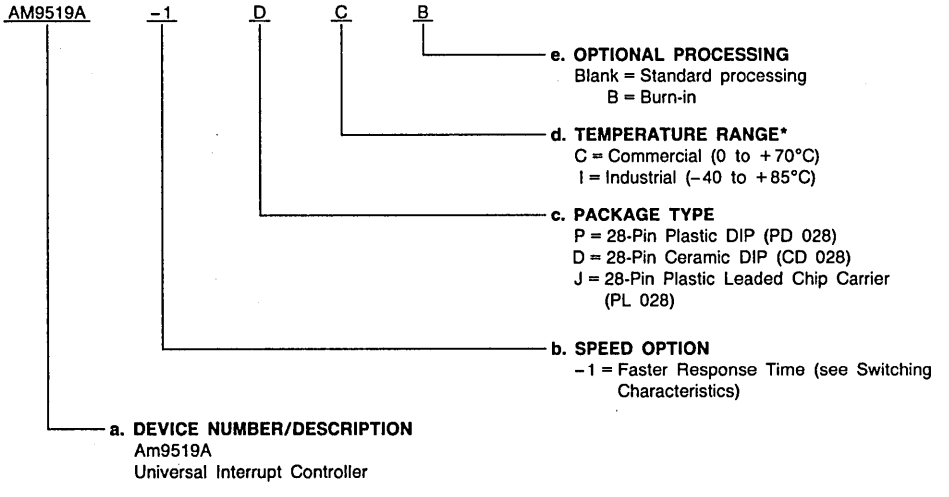


CD010880

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9519A	PC, DC, DCB, DIB, JC
AM9519A-1	PC, DC, DCB, JC

*This device is also available in Military temperature range.

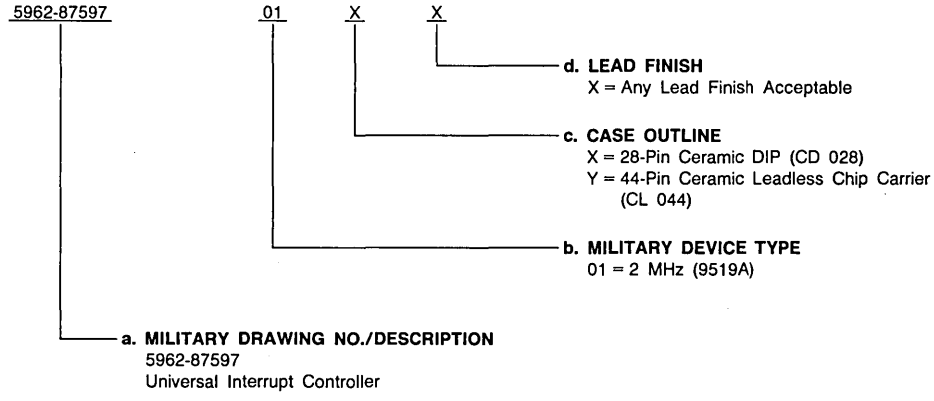
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION (continued)

Standard Military Drawing (SMD)/DESC Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. Standard Military Drawing (SMD)/DESC products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for SMD/DESC products is formed by a combination of: **a. Military Drawing Part Number**
b. Device Type
c. Case Outline
d. Lead Finish



Valid Combinations	
5962-8759701	XX, YX

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

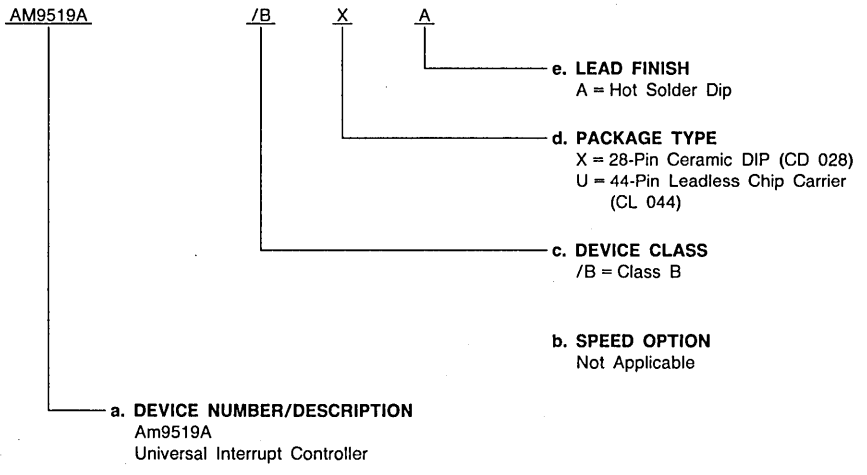
Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

2

ORDERING INFORMATION (continued)

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM9519A	/BXA, /BUA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
28	V_{CC}		+5 Volt Power Supply.
14	V_{SS}		Ground.
11-4	DB0-DB7	I/O	(Data Bus). The eight bidirectional data bus signals are used to transfer information between the Am9519A and the system data bus. The direction of transfer is controlled by the $\overline{IACK}$, $\overline{WR}$ and $\overline{RD}$ input signals. Programming and control information are written into the device; status and response data are output by it.
1	$\overline{CS}$	I	(Chip Select). The active low Chip Select input enables read and write operations on the data bus. Interrupt acknowledge responses are not conditioned by $\overline{CS}$.
3	$\overline{RD}$	I	(Read). The active low Read signal is conditioned by $\overline{CS}$ and indicates that information is to be transferred from the Am9519A to the data bus.
2	$\overline{WR}$	I	(Write). The active low Write signal is conditioned by $\overline{CS}$ and indicates that data bus information is to be transferred from the data bus to a location within the Am9519A.
27	C/$\overline{D}$	I	(Control/Data). The C/$\overline{D}$ control signal selects source and destination locations for data bus read and write operations. Data read or write transfers are made to or from preselected internal registers or memory locations. Control write operations load the command register and control read operations output the status register.
18-25	IREQ0-IREQ7	I	(Interrupt Request). The Interrupt Request signals are used by external devices to indicate that service by the host CPU is desired. IREQ inputs are accepted asynchronously and they may be programmed for either a HIGH-to-LOW or LOW-to-HIGH edge transition. Active inputs are latched internally in the Interrupt Request Register. After the IRR bit is cleared, an IREQ transition of the programmed polarity must occur to initiate another request.
12	$\overline{RIP}$	I/O	(Response In Process). Response In Process is a bidirectional signal used when two or more Am9519A circuits are cascaded. It permits multibyte response transfers to be completed without interference from higher priority interrupts. An Am9519A that is responding to an acknowledged interrupt will treat $\overline{RIP}$ as an output and hold it LOW until the acknowledge response is finished. An Am9519A without an acknowledged interrupt will treat $\overline{RIP}$ as an input and will ignore $\overline{IACK}$ pulses as long as $\overline{RIP}$ is LOW. The $\overline{RIP}$ output is open drain and requires an external pull-up resistor to VCC.
26	$\overline{IACK}$	I	(Interrupt Acknowledge). The active-low Interrupt Acknowledge line indicates that the external system is asking for interrupt response information. Depending on the programmed state of the Am9519A, it will accept 1, 2, 3 or 4 $\overline{IACK}$ pulses; one response byte is transferred per pulse. The first $\overline{IACK}$ pulse causes selection of the highest priority unmasked pending interrupt request and generates a $\overline{RIP}$ output signal.
15	$\overline{PAUSE}$	O	(Pause). The active-low Pause signal is used to coordinate interrupt responses with data bus and control timing. Pause goes LOW when the first $\overline{IACK}$ is received and remains LOW until $\overline{RIP}$ goes LOW. The external system can use Pause to stretch the acknowledge cycle and allow the control timing to automatically adjust to the actual priority resolution delays in the interrupt system. Second, third and fourth response bytes do not cause Pause to go LOW. Pause is an open drain output and requires an external pull-up resistor to VCC.
16	EO	O	(Enable Out). The active-high EO signal is used to implement daisy-chained cascading of several Am9519A circuits. EO is connected to the EI input of the next lower priority chip. On receipt of an interrupt acknowledge, each EO will go inactive until it has been determined that no valid interrupt request is pending on that chip. If an active request is present, EO remains LOW. EO is also held LOW when the master mask bit is active, thus disabling all lower priority chips.
13	EI	I	(Enable in). The active-high EI signal is used to implement daisy-chained cascading of several Am9519A circuits. EI is connected to EO of the next higher priority chip. It may also be used as a hardware disable input for the interrupt system. When EI is LOW, $\overline{IACK}$ inputs will not affect ISR; however, $\overline{PAUSE}$ will go LOW until $\overline{RIP}$ goes LOW. EI is internally pulled up to VCC so that no external pull-up is needed when EI is not used.
17	GINT	O	(Group Interrupt). The Group Interrupt output signal indicates that at least one unmasked interrupt request is pending. It may be programmed for active-high or active-low polarity. When active-low, the output is open drain and requires an external pull-up resistor to VCC. Since a glitch on GINT occurs approximately 100nsec after the last IACK pulse, this pin should not be connected to edge sensitive devices.

PRODUCT OVERVIEW

Register Description

Interrupt Request Register (IRR): The 8-bit IRR is used to store pending interrupt requests. A bit in the IRR is set whenever the corresponding IREQ input goes active. Bits may also be set under program control from the CPU, thus permitting software generated interrupts. IRR bits may be cleared under program control. An IRR bit is automatically cleared when its interrupt is acknowledged. All IRR bits are cleared by a reset function.

Interrupt Service Register (ISR): The 8-bit ISR contains one bit for each IREQ input. It is used to indicate that a pending interrupt has been acknowledged and to mask all lower priority interrupts. When a bit is set by the acknowledge logic in the ISR, the corresponding IRR bit is cleared. If an acknowledged interrupt is not programmed to be automatically cleared, its ISR bit must be cleared by the CPU under program control when it is desired to permit interrupts from lower priority devices. When the interrupt is programmed for automatic clearing, the ISR bit is automatically reset during the acknowledge sequence. All ISR bits are cleared by a reset function.

Interrupt Mask Register (IMR): The 8-bit IMR is used to enable or disable the individual interrupt inputs. The IMR bits correspond to the IREQ inputs, and all eight may be loaded, set or cleared in parallel under program control. In addition, individual IMR bits may be set or cleared by the CPU. Care must be taken therefore when disabling a specific channel by setting its IMR bit. If that bit is causing the GINT pin to be active, a lock-up condition can occur if the CPU recognizes the interrupt and then the Am9519A removes the request. During the $\overline{IACK}$ cycle, $\overline{PAUSE}$ will go LOW and stay LOW. The solution is to disable CPU interrupts prior to writing to the IMR and then re-enable them. A reset function will set all eight

mask bits, disabling all requests. A mask bit that is set does not disable the IRR, and an IREQ that arrives while a corresponding mask bit is set will cause an interrupt later when the mask bit is cleared. Only unmasked interrupt inputs can generate a Group Interrupt output.

Response Memory: An 8 x 32 read/write response memory is included in the Am9519A. It is used to store up to four bytes of response information for each of the eight interrupt request inputs. All bits in the memory are programmable, allowing any desired vector, opcode, instruction or other data to be entered. The Am9519A transfers the interrupt response information for the highest priority unmasked interrupt from the memory to the data bus when the $\overline{IACK}$ input is active.

Auto Clear Register: The 8-bit Auto Clear register contains one bit for each IREQ input and specifies the operating mode for each of the ISR bits. When an auto clear bit is off, the corresponding ISR bit is set when that interrupt is acknowledged and is cleared by software command. When an auto clear bit is on, the corresponding ISR bit is cleared by the hardware by the rising edge of the last acknowledge pulse. A reset function clears all auto clear bits.

Status Register: The 8-bit Status register contains information concerning the internal state of the chip. It is especially useful when operating in the polled mode to identify interrupting devices. Figure 1 shows the status register bit assignments. The polarity of the GINT bit 7 is not affected by the GINT polarity control. Bits S0-S2 are set asynchronously to a status register read operation. It is recommended to read the register twice and to compare the binary vectors for equality prior to proceeding with the device service in polled mode. The polarity of the GINT bit 7 is not affected by the GINT polarity control (Mode bit 3). The Status register is read by executing a read operation ($\overline{CS} = 0$, $RS = 0$) with the control location selected ($C/\overline{D}$ = 1).

Mode Register: The 8-bit Mode register controls the operating options of the Am9519A. Figure 2 shows the bit assignments for the Mode register. The five low order mode bits (0 through 4) are loaded in parallel by command. Bits 5, 6 and 7 are controlled by separate commands. (See Figure 4.) The Mode register cannot be read out directly to the data bus, but Mode bits 0, 2 and 7 are available as part of the Status register.

Command Register: The 8-bit Command register stores the last command entered. Depending upon the command opcode, it may initiate internal actions or precondition the part for subsequent data bus transfers. The Command register is loaded by executing a write operation ($\overline{WR}$ = 0) with the control location selected ($C/\overline{D}$ = 1), as shown in Figure 3.

Byte Count Register: The length in bytes of the response associated with each interrupt is independently programmed so that different interrupts may have different length responses. The byte count for each response is stored in eight 2-bit Byte Count registers. For a given interrupt, the Am9519A will expect to receive a number of $\overline{IACK}$ pulses that equal the corresponding byte count and will hold $\overline{RIP}$ LOW until the count is satisfied.

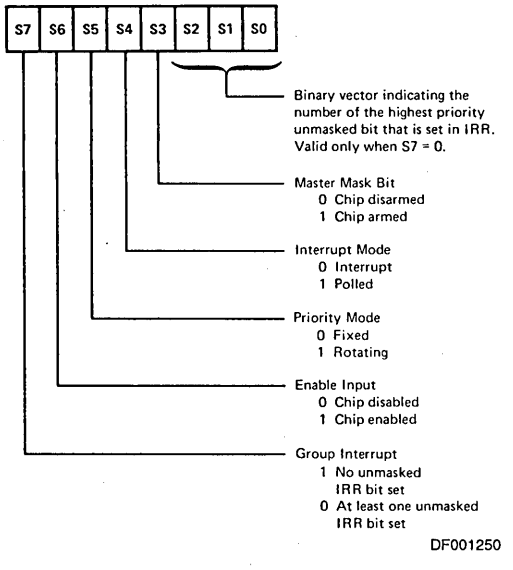


Figure 1. Status Register Bit Assignments

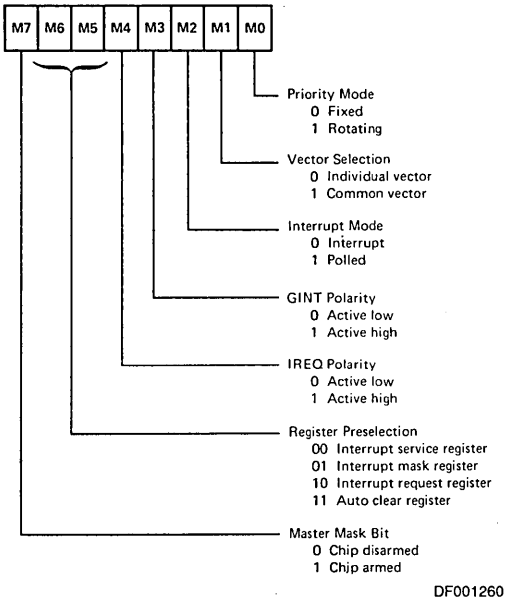


Figure 2. Mode Register Bit Assignments

DETAILED DESCRIPTION

Interrupts are used to improve system throughput and response time by eliminating heavy dependence on software polling procedures. Interrupts allow external devices to asynchronously modify the instruction sequence of a program being executed. In systems with multiple interrupts, vectoring can further improve performance by allowing direct identification of the interrupting device and its associated service routine. The Am9519A Universal Interrupt Controller contains, on one chip, all of the circuitry necessary to detect, prioritize and manage eight vectored interrupts. It includes many

options and operating modes that permit the design of sophisticated interrupt systems.

Reset

The reset function is accomplished by software command or automatically during power-up. The reset command may be issued by the CPU at any time. Internal power-up circuitry is triggered when VCC reaches a predetermined threshold, causing a brief internal reset pulse. In both cases, the resulting internal state of the machine is that all registers are cleared except the Mask register which is set. Thus, no Group Interrupt will be generated, and no interrupt requests will be recognized. The response memory and Byte Count registers are not affected by reset. Their contents after power-up are unpredictable and must be established by the host CPU during initialization.

Operating Sequence

A brief description of a typical sequence of events in an operating interrupt system will illustrate the general interactions among the host CPU, the interrupt controller and the interrupting peripheral.

1. The Am9519A controller is initialized by the CPU to customize its configuration and operation for the application at hand. Both the controller and the CPU are then enabled to accept interrupts.
2. One (or more) of the interrupt request inputs to the controller becomes active indicating that peripheral equipment is asking for service. The controller asynchronously accepts and latches the request(s).
3. If the request is masked, no further action takes place. If the request is not masked, a Group Interrupt output is generated by the controller.
4. The GINT signal is recognized by the CPU which normally will complete the execution of the current instruction, insert an interrupt acknowledge sequence into its instruction execution stream, and disable its internal interrupt structure. The controller expects to receive one or more $\overline{IACK}$ signals from the CPU during the acknowledge sequence.
5. When the controller receives the $\overline{IACK}$ signal, it brings $\overline{PAUSE}$ low and selects the highest priority unmasked pending request. When selection is complete, the $\overline{RIP}$ output is brought low and the first byte in the response memory associated with the selected request is output on the data bus. $\overline{PAUSE}$ stays low until RIP goes low. $\overline{RIP}$ stays low until the last byte of the response has been transferred.
6. During the acknowledge sequence, the IRR bit corresponding to the selected request is automatically cleared, and the corresponding ISR bit is set by the falling edge of $\overline{IACK}$. When the ISR bit is set, the Group Interrupt output is disabled until a higher priority request arrives or the ISR bit is cleared. The ISR bit will be cleared by either hardware or software.
7. If a higher priority request arrives while the current request is being serviced, GINT will be output by the controller, but will be recognized and acknowledged only if the CPU has its interrupt input enabled. If acknowledged, the corresponding higher priority ISR bit will be set and the requests nested.

Information Transfers

Figure 3 shows the control signal configurations for all information transfer operations between the Am9519A and the data bus. The following conventions are assumed: $\overline{RD}$ and $\overline{WR}$ active are mutually exclusive; $\overline{RD}$, $\overline{WR}$ and C/$\overline{D}$ have no meaning unless $\overline{CS}$ is LOW; active $\overline{IACK}$ pulses occur only when $\overline{CS}$ is HIGH.

For reading, the Status register is selected directly by the C/$\overline{D}$ control input. Other internal registers are read by preselecting the desired register with mode bits 5 and 6, and then executing a data read. The response memory can be read only with $\overline{IACK}$ pulses. For writing, the Command register is selected directly by the C/$\overline{D}$ control input. The Mask and Auto Clear registers are loaded following specific commands to that effect. To load each level of the response memory, the response preselect command is issued to select the desired level. An appropriate number of data write operations are then executed to load that level.

CONTROL INPUT					
$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{IACK}$	DATA BUS OPERATION
0	0	0	1	1	Transfer contents of preselected data register to data bus
0	0	1	0	1	Transfer contents of data bus to preselected data register
0	1	0	1	1	Transfer contents of status register to data bus
0	1	1	0	1	Transfer contents of data bus to command register
1	X	X	X	0	Transfer contents of selected response memory location to data bus
1	X	X	X	1	No information transferred

Figure 3. Summary of Data Bus Transfers

The Pause output may be used by the host CPU to ensure that propertiming relationships are maintained with the Am9519A when $\overline{IACK}$ is active. The $\overline{IACK}$ pulse width required depends on several variables, including: operating temperature, internal logic delays, number of interrupt controllers chained together, and the priority level of the interrupt being acknowledged. When delays in these variables combine to delay selection of a request following the falling edge of the first $\overline{IACK}$, the Pause output may be used to extend the $\overline{IACK}$ pulse, if necessary. Pause will remain LOW until a request has been selected, as indicated by the falling edge of $\overline{RIP}$. Typically, the internal interrupt selection process is quite fast, especially for systems with a single Am9519A, and Pause will consequently remain LOW for only a very brief interval and will not cause extension of the $\overline{IACK}$ timing.

Operating Options

The Mode register specifies the various combinations of operating options that may be selected by the CPU. It is cleared by power-up or by a reset command. Mode bit 0 specifies the rotating/fixed priority mode (see Figure 2). In the fixed mode, priority is assigned to the request inputs based upon their physical location at the chip interface, with IREQ0 the highest and IREQ7 the lowest. In the rotating mode, relative priority is the same as for the fixed mode and the most recently serviced request is assigned the lowest priority. In the fixed mode, a lower priority request might never receive service if enough higher priority requests are active. In the rotating mode, any request will receive service within a maximum of seven other service cycles no matter what pattern the request inputs follow.

Mode bit 1 selects the individual/common vector option. Individual vectoring provides a unique location in the response memory for each interrupt request. The common vector option always supplies the response associated with IREQ0 no matter which request is being acknowledged.

Mode bit 2 specifies interrupt or polled operation. In the polled mode, the Group Interrupt output is disabled. The CPU may read the Status register to determine if a request is pending.

Since $\overline{IACK}$ pulses are not normally supplied in polled mode, the IRR bit is not automatically cleared, but may be cleared by command. With no $\overline{IACK}$ input, the ISR and the response memory are not used. An Am9519A in the polled mode has EI connected to EO so that in multichip interrupt systems the polled chip is functionally removed from the priority hierarchy.

Mode bit 3 specifies the sense of the GINT output. When active high polarity is selected, the output is a two-state configuration. For active low polarity, the output is open drain and requires an external pull-up resistor to provide the high logic level. The open drain output allows wired-or configurations with other similar output signals.

Mode bit 4 specifies the sense of the IREQ inputs. When active low polarity is selected, the IRR responds to falling edges on the request inputs. When active high is selected, the IRR responds to rising edges.

Mode bits 5 and 6 specify the register that will read on subsequent data read operations (C/$\overline{D}$ = 0, $\overline{RD}$ = 0). This preselection remains valid until changed by a reset or a command.

Mode bit 7 is the master mask bit that disables all request inputs. It is used to disable all interrupts without modifying the IMR so that the previous IMR contents are valid when interrupts are re-enabled. When the master mask bit is LOW, it causes the EO line to remain disabled (LOW). Thus, for multiple-chip interrupt systems, one master mask bit can disable the whole interrupt structure. Alternatively, portions of the structure may be disabled. The state of the master mask bit is available as bit S3 of the Status register.

Programming

After reset, the Am9519A must be initialized by the CPU to perform useful work. At a minimum, the master mask bit and at least one of the IMR bits should be enabled. If vectoring is to be used, the response memory must be loaded; if not, the mode must be changed to a non-vectored configuration. Normally, the first step will be to modify the Mode register and the Auto clear register to establish the configuration desired for the application. Then the response memory and byte count will be loaded for those request levels that will be in use. The response memory for every channel must be written even if the channel is not used. Every byte need not be written, only those specified by the byte count. Finally, the master mask bit and at least portions of the IMR will be enabled to allow interrupt processing to proceed.

Commands

The host CPU configures, changes and inspects the internal condition of the Am9519A using the set of commands shown in Figure 4. An "X" entry in the table indicates a "don't care" state. All commands are entered by directly loading the Command register as shown in Figure 3 (C/$\overline{D}$ = 1, $\overline{WR}$ = 0). Figure 5 shows the coding assignments for the Byte Count registers. (A detailed description of each command is contained in the Am9519A Application Note AMPUB-071.)

BY1	BY0	COUNT
0	0	1
0	1	2
1	0	3
1	1	4

Figure 5. Byte Count Coding

COMMAND CODE								
7	6	5	4	3	2	1	0	COMMAND DESCRIPTION
0	0	0	0	0	0	0	0	Reset
0	0	0	1	0	X	X	X	Clear all IRR and all IMR bits
0	0	0	1	1	B2	B1	B0	Clear IRR and IMR bit specified by B2, B1, B0
0	0	1	0	0	X	X	X	Clear all IMR bits
0	0	1	0	1	B2	B1	B0	Clear IMR bit specified by B2, B1, B0
0	0	1	1	0	X	X	X	Set all IMR bits
0	0	1	1	1	B2	B1	B0	Set IMR bit specified by B2, B1, B0
0	1	0	0	0	X	X	X	Clear all IRR bits
0	1	0	0	1	B2	B1	B0	Clear IRR bit specified by B2, B1, B0
0	1	0	1	0	X	X	X	Set all IRR bits
0	1	0	1	1	B2	B1	B0	Set IRR bit specified by B2, B1, B0
0	1	1	0	X	X	X	X	Clear highest priority ISR bit
0	1	1	1	0	X	X	X	Clear all ISR bits
0	1	1	1	1	B2	B1	B0	Clear ISR bit specified by B2, B1, B0
1	0	0	M4	M3	M2	M1	M0	Load Mode register bits 0–4 with specified pattern
1	0	1	0	M6	M5	0	0	Load Mode register bits 5, 6 with specified pattern
1	0	1	0	M6	M5	0	1	Load Mode register bits 5, 6 and set mode bit 7
1	0	1	0	M6	M5	1	0	Load Mode register bits 5, 6 and clear mode bit 7
1	0	1	1	X	X	X	X	Preselected IMR for subsequent loading from data bus
1	1	0	0	X	X	X	X	Preselected Auto Clear register for subsequent loading from data bus
1	1	1	BY1	BY0	L2	L1	L0	Load BY1, BY0 into byte count register and preselect response memory level specified by L2, L1, L0 for subsequent loading from data bus

Figure 4. Am9519A Command Summary

APPLICATIONS

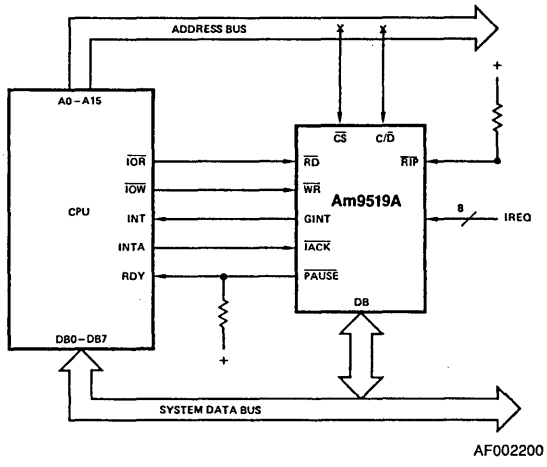


Figure 6. Base Interrupt System Configuration

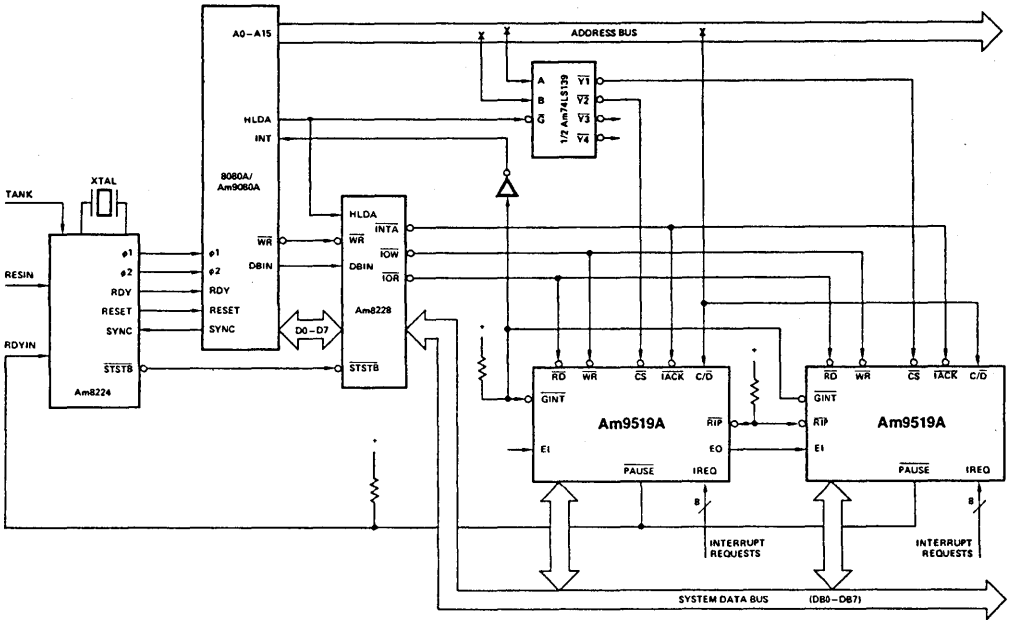


Figure 7. Expanded Interrupt System Configuration

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
VCC with Respect to VSS ..... −0.5 V to +7.0 V
All Signal Voltages
with Respect to VSS ..... −0.5 V to +7.0 V
Power Dissipitation (Package Limitation) ..... 1.5 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

Industrial (I) Devices
Temperature (T_A) ..... −40 to +85°C
Supply Voltage (V_{CC}) ..... 5 V ± 10%

Military (M) Devices
Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... 5 V ± 10%

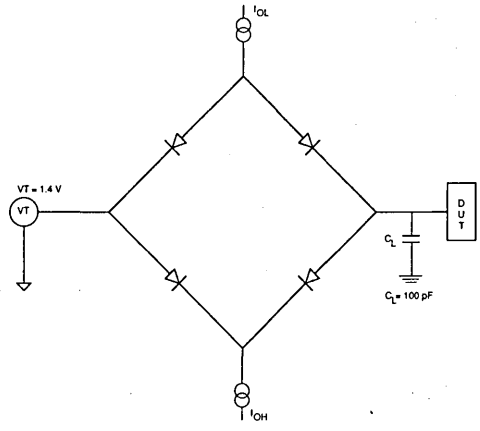
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified (for APL and SMD/DESC Products; Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameters	Description	Test Conditions		Min	Max	Units
VOH	Output High Voltage (Note 8)	IOH = −200μA		2.4		Volts
		IOH = −100μA (EO only)		2.4		
VOL	Output Low Voltage	IOL = 3.2mA			0.4	Volts
		IOL = 1.0mA (EO only)			0.4	
VIH	Input High Voltage			2.0	VCC*	Volts
VIL	Input Low Voltage			−0.5*	0.8	Volts
IIX	Input Load Current	VSS ≤ VIN ≤ VCC	EI Input	−60	10	μA
			Other Inputs	−10	10	
IOZ	Output Leakage Current	VSS ≤ VOUT ≤ VCC, Output Off		COML −10	10	μA
				MIL 150	150	
ICC	CXX Supply Current	Commercial			125	mA
		Industrial			185	
		Military			200	
CO	Output Capacitance	fc = 1.0 MHz			15*	pF
CI	Input Capacitance	T_A = 25°C			10*	
CIO	I/O Capacitance	All pins at 0 V			20*	

*Guaranteed by design — not tested.

SWITCHING TEST CIRCUIT



TC004200

This test circuit is the dynamic load of a Teradyne J941.

SWITCHING TEST INPUT/OUTPUT WAVEFORM

WF007820

SWITCHING CHARACTERISTICS over COMMERCIAL operating ranges (Notes 1, 2)

Parameters	Description	Am9519A		Am9519A-1		Units
		Min	Max	Min	Max	
TAVRL	C/$\overline{D}$ Valid and $\overline{CS}$ LOW to Read LOW	0		0		ns
TAVWL	C/$\overline{D}$ Valid and $\overline{CS}$ LOW to Write LOW	0		0		ns
TCLPH	$\overline{RIP}$ LOW to $\overline{PAUSE}$ HIGH (Note 3)	75	375	75	375	ns
TCLQV	$\overline{RIP}$ LOW to Data Out Valid (Note 4)		50		40	ns
TDVWH	Data in Valid to Write HIGH	250		200		ns
TEHCL	Enable in HIGH to $\overline{RIP}$ LOW (Note 5)	30	300	30	300	ns
TIVGV	Interrupt Request Valid to Group Interrupt Valid	100	800		650	ns
TIVIX	Interrupt Request Valid to Interrupt Request Don't Care (IREQ Pulse Duration)	250		250		ns
TKHCH	$\overline{IACK}$ HIGH to $\overline{RIP}$ HIGH (Note 5)		450		350	ns
TKHKL	$\overline{IACK}$ HIGH to $\overline{IACK}$ LOW ($\overline{IACK}$ Recovery)	140		120		ns
TKHNH	$\overline{IACK}$ HIGH to EO HIGH (Notes 6, 7)		975		750	ns
TKHQX	$\overline{IACK}$ HIGH to Data Out Invalid	20	200	20	100	ns
TKLCL	$\overline{IACK}$ LOW to $\overline{RIP}$ LOW (Notes 5, 9)	75	600	75	450	ns
TKLKH	$\overline{IACK}$ LOW to $\overline{IACK}$ HIGH (1st $\overline{IACK}$) (Note 9)	975		800		ns
TKLNL	$\overline{IACK}$ LOW to EO LOW (Notes 6, 7, 9)		125		100	ns
TKLPL	$\overline{IACK}$ LOW to $\overline{PAUSE}$ LOW (Note 9)	25	175	25	125	ns
TKLQV	$\overline{IACK}$ LOW to Data Out Valid (Notes 4, 9)	25	300	25	200	ns
TKLQV1	1st $\overline{IACK}$ LOW to Data Out Valid (Note 9)	75	650	75	490	ns
TPHKH	$\overline{PAUSE}$ HIGH to $\overline{IACK}$ HIGH	0		0		ns
TRHAX	Read HIGH to C/$\overline{D}$ and $\overline{CS}$ Don't Care	0		0		ns
TRHQX	Read HIGH to Data Out Invalid	20	200	20	100	ns
TRLQV	Read LOW to Data Out Valid		300		200	ns
TRLQX	Read LOW to Data Out Unknown	35		35		ns
TRLRH	Read LOW to Read HIGH ($\overline{RD}$ Pulse Duration)	300		250		ns
TWHAX	Write HIGH to C/$\overline{D}$ and $\overline{CS}$ Don't Care	25		25		ns
TWHDX	Write HIGH to Data in Don't Care	25		25		ns
TWHRW	Write HIGH to Read or Write LOW (Write Recovery)	600		400		ns
TWLWH	Write LOW to Write HIGH ($\overline{WR}$ Pulse Duration)	300		250		ns
TKHIH	$\overline{IACK}$ HIGH to GINT inactive		1000		800	ns

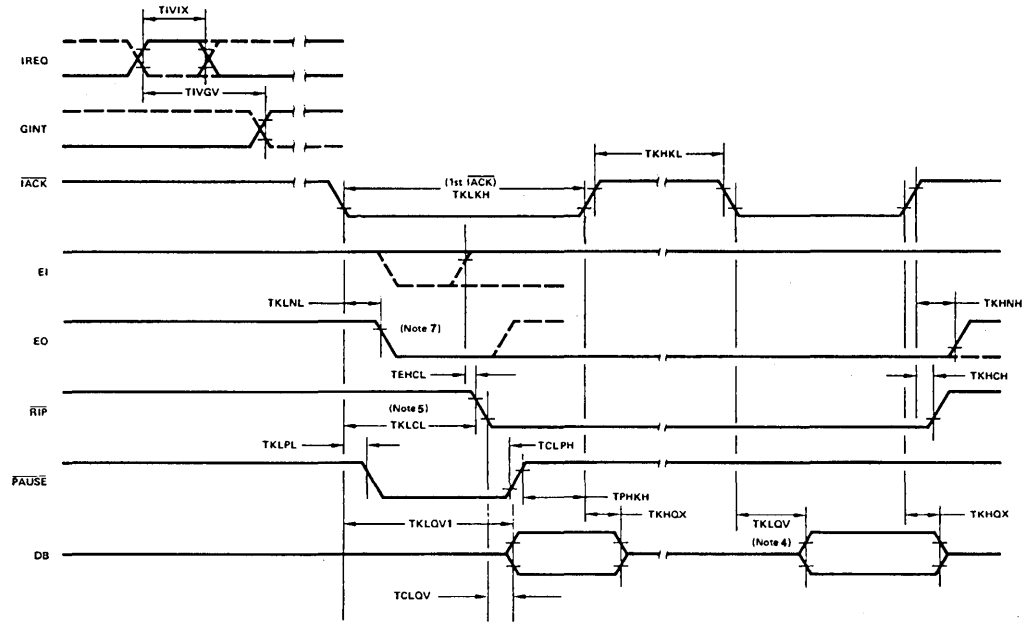
Notes:
1. Transition abbreviations used for the switching parameter symbols include: H = HIGH, L = LOW, V = Valid, X = unknown or don't care, Z = high-impedance.
2. Signal abbreviations used for the switching parameter symbols include: R = Read, W = Write, Q = Data Out, D = Data In, A = Address ($\overline{CS}$ and C/$\overline{D}$), K = Interrupt Acknowledge, N = Enable Out, E = Enable In, P = Pause, C = $\overline{RIP}$.
3. During the first $\overline{IACK}$ pulse, $\overline{PAUSE}$ will be LOW long enough to allow for priority resolution and will not go HIGH until after RIP goes LOW (TCLPH).
4. TKLQV applies only to second, third and fourth $\overline{IACK}$ pulses while $\overline{RIP}$ is LOW. During the first $\overline{IACK}$ pulse, Data Out will be valid following the falling edge of $\overline{RIP}$ (TCLQV).
5. $\overline{RIP}$ is pulled LOW to indicate that an interrupt request has been selected. $\overline{RIP}$ cannot be pulled LOW until EI is HIGH following an internal delay. TKLCL will govern the falling edge of $\overline{RIP}$ when EI is always HIGH or is HIGH early in the acknowledge cycle. The TEHCL will govern when EI goes HIGH later in the cycle. The rising edge of EI will be determined by the length of the preceding priority resolution chain.
   $\overline{RIP}$ remains LOW until after the rising edge of the $\overline{IACK}$ pulse that transfers the last response byte for the selected IREQ.
6. Test conditions for the EO line assume an output loading of IOL = 1.0 mA and IOH = −100 μA. Since EO normally only drives EI of another Am9519A, higher speed operations can be specified with this more realistic test condition.
7. The arrival of $\overline{IACK}$ will cause EO to go LOW, disabling additional circuits that may be connected to EO. If no valid interrupt is pending, EO will return HIGH when EI is HIGH. If a pending request is selected, EO will stay LOW until after the last $\overline{IACK}$ pulse for that interrupt is complete and $\overline{RIP}$ goes HIGH.
8. VOH specifications do not apply to $\overline{RIP}$, $\overline{PAUSE}$, or to $\overline{GINT}$ when active-low. These outputs are open drain, and VOH levels will be determined by external circuitry.
9. $\overline{CS}$ must be HIGH for at least 100ns prior to $\overline{IACK}$ going LOW.

SWITCHING CHARACTERISTICS over **MILITARY** operating ranges (for SMD/DESC and APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted). (Notes 1, 2)

No.	Parameter Symbol	Parameter Description	Am9519A		Unit
			Min.	Max.	
1	TAVRL	C/$\overline{D}$ Valid and $\overline{CS}$ LOW to Read LOW	0		ns
2	TAVWL	C/$\overline{D}$ Valid and $\overline{CS}$ LOW to Write LOW	0		ns
3	TCLPH	$\overline{RIP}$ LOW to $\overline{PAUSE}$ HIGH (Note 3)	75	375	ns
4	TCLQV	$\overline{RIP}$ LOW to Data Out Valid (Note 4)		50	ns
5	TDVWH	Data in Valid to Write HIGH	250		ns
6	TEHCL	Enable in HIGH to $\overline{RIP}$ LOW (Note 5)	30	300	ns
7	TIVGV	Interrupt Request Valid to Group Interrupt Valid	100	800	ns
8	TIVIX	Interrupt Request Valid to Interrupt Request Don't Care (IREQ Pulse Duration)	250		ns
9	TKHCH	$\overline{IACK}$ HIGH to $\overline{RIP}$ HIGH (Note 5)		450	ns
10	TKHKL	$\overline{IACK}$ HIGH to $\overline{IACK}$ LOW ($\overline{IACK}$ Recovery)	140		ns
11	TKHNH	$\overline{IACK}$ HIGH to EO HIGH (Notes 6, 7)		975	ns
12	TKHQX	$\overline{IACK}$ HIGH to Data Out Invalid	20	200	ns
13	TKLCL	$\overline{IACK}$ LOW to $\overline{RIP}$ LOW (Notes 5, 9)	75	650	ns
14	TKLKH	$\overline{IACK}$ LOW to $\overline{IACK}$ HIGH (1st $\overline{IACK}$) (Note 9)	975		ns
15	TKLNL	$\overline{IACK}$ LOW to EO LOW (Notes 6, 7, 9)		125	ns
16	TKLPL	$\overline{IACK}$ LOW to $\overline{PAUSE}$ LOW (Note 9)	25	175	ns
17	TKLQV	$\overline{IACK}$ LOW to Data Out Valid (Notes 4, 9)	25	300	ns
18	TKLQV1	1st $\overline{IACK}$ LOW to Data Out Valid (Note 9)	75	650	ns
19	TPHKH	$\overline{PAUSE}$ HIGH to $\overline{IACK}$ HIGH	0		ns
20	TRHAX	Read HIGH to C/$\overline{D}$ and $\overline{CS}$ Don't Care	0		ns
21	TRHQX	Read HIGH to Data Out Invalid	20	200	ns
22	TRLQV	Read LOW to Data Out Valid		300	ns
23	TRLQX	Read LOW to Data Out Unknown	35		ns
24	TRLRH	Read LOW to Read HIGH ($\overline{RD}$ Pulse Duration)	300		ns
25	TWHAX	Write HIGH to C/$\overline{D}$ and $\overline{CS}$ Don't Care	25		ns
26	TWHDX	Write HIGH to Data in Don't Care	25		ns
27	TWHRW	Write HIGH to Read or Write LOW (Write Recovery)	600		ns
28	TWLWH	Write LOW to Write HIGH ($\overline{WR}$ Pulse Duration)	300		ns
29	TKHIH	$\overline{IACK}$ HIGH to $\overline{GINT}$ Inactive		1000	ns

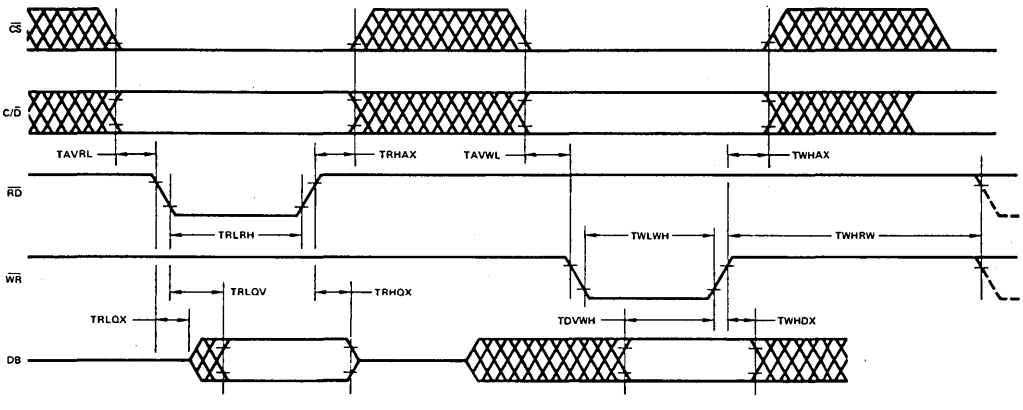
Notes:
1. Transition abbreviations used for the switching parameter symbols include: H = HIGH, L = LOW, V = Valid, X = unknown or don't care, Z = high-impedance.
2. Signal abbreviations used for the switching parameter symbols include: R = Read, W = Write, Q = Data Out, D = Data In, A = Address ($\overline{CS}$ and C/$\overline{D}$), K = Interrupt Acknowledge, N = Enable Out, E = Enable In, P = Pause, C = $\overline{RIP}$.
3. During the first $\overline{IACK}$ pulse, $\overline{PAUSE}$ will be LOW long enough to allow for priority resolution and will not go HIGH until after RIP goes LOW (TCLPH).
4. TKLQV applies only to second, third and fourth $\overline{IACK}$ pulses while $\overline{RIP}$ is LOW. During the first $\overline{IACK}$ pulse, Data Out will be valid following the falling edge of $\overline{RIP}$ (TCLQV).
5. $\overline{RIP}$ is pulled LOW to indicate that an interrupt request has been selected. $\overline{RIP}$ cannot be pulled LOW until EI is HIGH following an internal delay. TKLCL will govern the falling edge of $\overline{RIP}$ when EI is always HIGH or is HIGH early in the acknowledge cycle. The TEHCL will govern when EI goes HIGH later in the cycle. The rising edge of EI will be determined by the length of the preceding priority resolution chain.
   $\overline{RIP}$ remains LOW until after the rising edge of the $\overline{IACK}$ pulse that transfers the last response byte for the selected IREQ.
6. Test conditions for the EO line assume an output loading of IOL = 1.0 mA and IOH = −100 μA. Since EO normally only drives EI of another Am9519A, higher speed operations can be specified with this more realistic test condition.
7. The arrival of $\overline{IACK}$ will cause EO to go LOW, disabling additional circuits that may be connected to EO. If no valid interrupt is pending, EO will return HIGH when EI is HIGH. If a pending request is selected, EO will stay LOW until after the last $\overline{IACK}$ pulse for that interrupt is complete and $\overline{RIP}$ goes HIGH.
8. V_{OH} specifications do not apply to $\overline{RIP}$, $\overline{PAUSE}$, or to $\overline{GINT}$ when active-LOW. These outputs are open drain, and V_{OH} levels will be determined by external circuitry.
9. $\overline{CS}$ must be HIGH for at least 100 ns prior to $\overline{IACK}$ going LOW.

SWITCHING WAVEFORMS



WF003551

Interrupt Operations



WF003560

Data Bus Transfers

Am95C85

Content Addressable Data Manager

FINAL

DISTINCTIVE CHARACTERISTICS

- High-performance sorting, searching, and updating
- 1K byte software-reconfigurable memory array
- Programmable record size
- Cascadable up to 256 devices
- Content-addressable operation, independent of record size
- Intelligent peripheral with sixteen powerful instructions
- Stack mode allows inserting of data without resorting
- Up to 16-MHz operation
- CMOS technology
- 44 lead PLCC

GENERAL DESCRIPTION

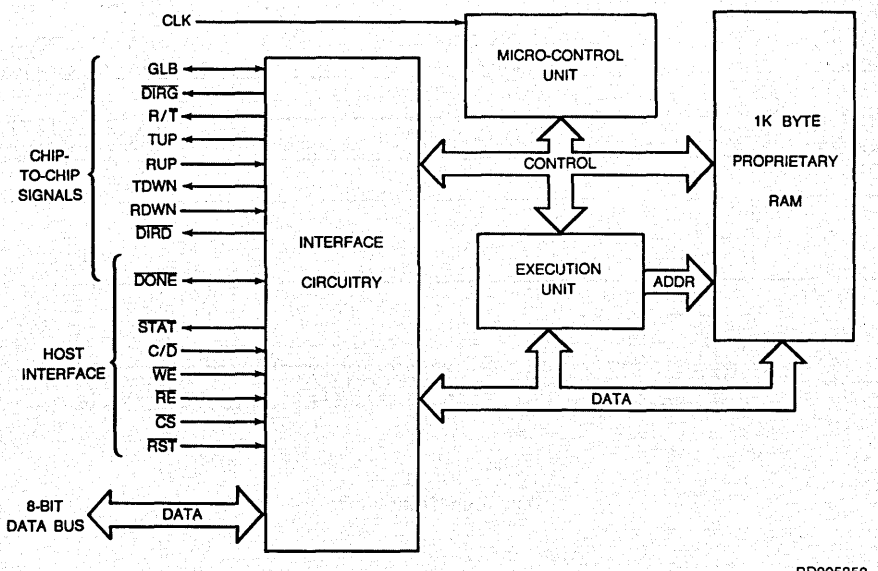
The Am95C85 Content Addressable Data Manager (CADM) is an intelligent CMOS peripheral device designed to enhance the performance of applications involving sorting, searching, and insertion or deletion. Orders of magnitude performance improvement can be seen when compared to the implementation using software algorithms.

The CADM uses an on-chip proprietary 1K byte memory for data manipulation. This specially designed memory can be easily reconfigured to meet different application requirements. The data stored in the CADM are collated into records that consist of a key field and a pointer field. The length of these two fields are software programmable. The sorting and searching of records are based on the values of the key fields. A mask register is also provided to selectively mask out unwanted bits in the key field for comparison. For applications that require large storage area for data manipulation, the CADM can be easily cascaded up to 256 devices.

Content-addressable operation allows the host to retrieve data without having to do extensive searching. Address generation for memory access is done internally, relieving the host from the burden of physical address calculation. Stack-mode operation allows the user to delete records simply by popping the records out of memory, and to insert records by pushing the records into the memory.

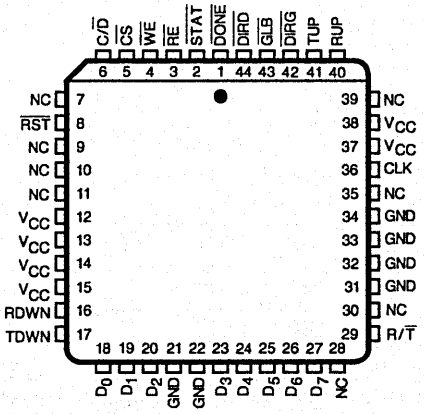
By providing content-addressable searching, automatic sorting, programmable record length, and address-independent operation, the CADM allows the host to off-load repetitive, time-consuming data manipulation. For applications that require substantial sorting, searching, and updating operations, the CADM offers significant improvement in overall performance.

BLOCK DIAGRAM



Publication # 06956 | Rev. C | Amendment /0
Issue Date: August 1989

CONNECTION DIAGRAM
Top View

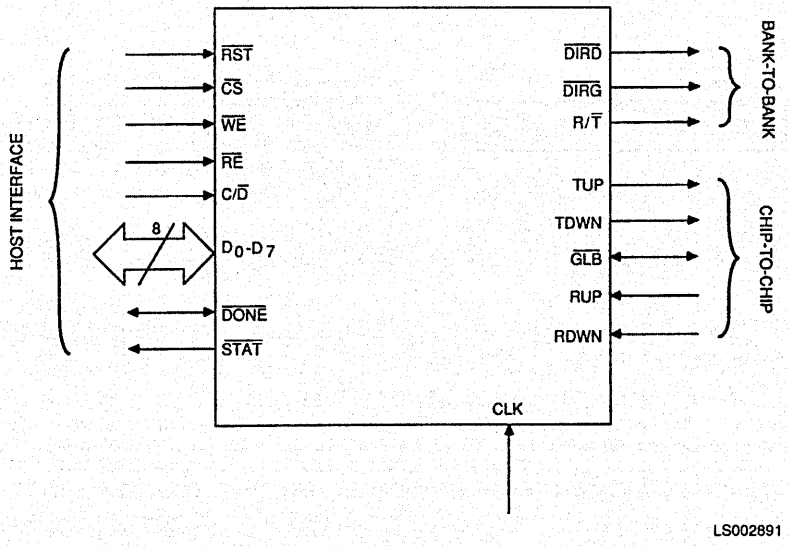


NC = No Connection

CD010442

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



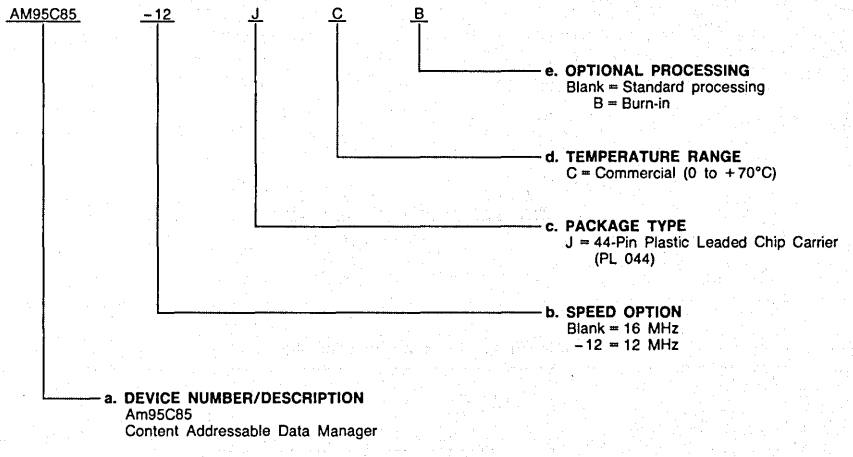
LS002891

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C85	JC
AM95C85-12	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Host Interface

C/$\overline{D}$ Command/$\overline{Data}$ (Input)
A HIGH on this input allows the command register to be loaded with the information on the data bus. A LOW on this input allows the data to be read from, or written into, the internal RAM.

$\overline{CS}$ $\overline{Chip\ Select}$ (Input; Active LOW)
The $\overline{CS}$ input enables the host CPU to perform read or write operations with the Am95C85 devices. The read and write inputs are ignored when $\overline{CS}$ is HIGH.

$D_0 - D_7$ Data Bus (Input/Output; Three State)
The eight bidirectional data pins are used for information exchanges between the Am95C85 (CADM) and the host processor, and between CADM parts themselves. A HIGH on a data line corresponds to a Logic "1," and a LOW corresponds to a Logic "0." These lines act as inputs when $\overline{WE}$ and $\overline{CS}$ are active, and as outputs when $\overline{RE}$ and $\overline{CS}$ are active. D_0 is the least significant bit and D_7 the most significant bit.

$\overline{DONE}$ $\overline{Done}$ (Input/Output; Active LOW, Three State)
This signal indicates the termination of an operation, and is precharged to HIGH at the beginning of a new command, data writes, or data reads. A LOW on this output indicates the device is ready for the next command or data transfer.

$\overline{RE}$ $\overline{Read\ Enable}$ (Input; Active LOW)
The $\overline{RE}$ input, together with $\overline{CS}$ and C/$\overline{D}$ inputs, are used to control data transfer from the Am95C85 to the host. The Am95C85 will put the data onto the data bus when $\overline{RE}$, $\overline{CS}$, and C/$\overline{D}$ inputs are LOW.

$\overline{RST}$ $\overline{Reset}$ (Input; Active LOW)
A LOW on this input will reset the Am95C85. Any command under execution is terminated.

$\overline{STAT}$ $\overline{Status}$ (Output; Active LOW, Three State)
When LOW, the $\overline{STAT}$ output indicates that an exception condition has occurred following the execution of an instruction or data transfer. This pin is precharged to HIGH at the beginning of a new command, or when a write or read is initiated.

$\overline{WE}$ $\overline{Write\ Enable}$ (Input; Active LOW)
The simultaneous occurrence of $\overline{WE}$ and $\overline{CS}$ indicates that information from the data bus is to be transferred to the Am95C85. The C/$\overline{D}$ input determines whether the data will be loaded into the command register or internal RAM.

Chip-to-Chip Communication

These pins are used in chip-to-chip communications in multiple Am95C85 memory configurations. They do not affect the system interface.

$\overline{GLB}$ $\overline{Global}$ (Input/Output; Active LOW, Three State)
This signal is used for part-to-part synchronization during instruction execution. All CADM devices in the same bank should have this pin connected together and pulled up through a resistor to the power supply. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

RDWN Receive from Downward (Input; Active HIGH)
This pin should be connected to TUP of the next lower order CADM in cascade. The last chip in the daisy chain should have this pin pulled up to the power supply through a resistor.

RUP Receive from Upward (Input; Active HIGH)
This pin should be connected to TDWN of the next higher order CADM in cascade. The first chip in the daisy chain should have this pin pulled up to the power supply through a resistor.

TDWN Transmit Downward (Output; Active HIGH)
This signal is issued by the higher order CADM to the next lower order CADM, in cascade, to synchronize the chip-to-chip data transfer. It should be connected to RUP of the next lower order CADM.

TUP Transmit Upward (Output; Active HIGH)
This handshaking signal is issued by the lower order CADM to the next higher order CADM, in cascade, during chip-to-chip data transfer. It should be connected to RDWN of the next higher order CADM.

Bank-to-Bank Control

Bank-to-bank communication is needed when multiple banks of Am95C85 devices are used in a system. The CADM array can be grouped into multiple banks and separated by buffers. The following signals are used to control the direction of buffer signals that separate the banks. They can be left unconnected if only one bank is used.

$\overline{DIRD}$ $\overline{Direction\ of\ Done\ Signal}$ (Output; Active LOW, Three State)
This signal is used to control the direction of the circuit that is buffering the $\overline{DONE}$ signal. All CADMs in the same bank should have this pin connected together and pulled up through a resistor to the power supply. When driven LOW, this signal indicates that the CADM is driving the $\overline{DONE}$ signal. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

$\overline{DIRG}$ $\overline{Direction\ of\ Global\ Signal}$ (Output; Active LOW, Three State)
This signal is used to control the direction of the circuit that is buffering the $\overline{GLB}$ signal. All CADMs in the same bank should have this pin connected together and pulled up through a resistor to the power supply. When driven LOW, this signal indicates that the CADM is driving the $\overline{GLB}$ signal. This pin is precharged to HIGH at the beginning of a new command, or when a Write or Read is initiated.

R/$\overline{T}$ Receive/$\overline{Transmit}$ (Output)
This output is driven LOW when the Am95C85 is driving the data bus. It should be used to control the direction of buffers which isolate the data bus from specific Am95C85 banks.

FUNCTIONAL DESCRIPTION

Introduction

The Am95C85 Content Addressable Data Manager (CADM) is an intelligent peripheral device intended to relieve the host CPU of many of the time-consuming tasks associated with data-list manipulation. Sorting and finding data are tasks implemented by both applications software and operating systems. By providing these functions in hardware, which were previously the responsibility of software, execution time is reduced. This performance improvement can be 100 to 500 times, depending upon the application.

The 44-pin Am95C85 contains 1K byte of RAM whose organization is programmable. It contains a micro-engine, registers, pointers, and an instruction decoder. Most of these functions are transparent to the user.

The Programmer's View

Hardware

The Am95C85 CADM interacts with the host system through the use of a command port, data port, and two status pins called $\overline{STAT}$ and $\overline{DONE}$. Both the command and data ports are accessed through the single 8-bit data bus. The two ports are differentiated by the use of a Command/$\overline{Data}$ pin (C/$\overline{D}$). The familiar signals, $\overline{RE}$, $\overline{WE}$, and $\overline{CS}$ are used to write and read data or commands.

Data Array

The CADM contains 1K byte of internal RAM. It consists of a mask area, a record area, an unused area, and an input buffer area as shown in Figure 1. The mask area exists only if an SMB command is issued. The length of the mask area is the same length as the key, as specified by the KPL command. The data stored in this area is used to select the desired bits in the key field for comparison during the sorting and searching process. Those mask bits with "0" will cause the associated bits in the key field to be ignored during the comparison. The record space stores data as records in the CADM. This area starts from address "K" if the masking option is chosen, or "0" if the masking option is not chosen. It ends at the last address as programmed by the KPL command. The length of this area should be a multiple of (K + P) bytes. The last (K + P) bytes are designated as input buffer area. They are reserved to temporarily store the incoming record. The remaining area between the record area and the input buffer is unused area and is not accessible by the user. This area should be kept as small as possible to optimize the performance of the CADM.

The internal RAM structure, a patented AMD design, is unique in that the record width is controlled by the CPU, using the KPL command. Each record is comprised of two fields, referred to as a key field (K) and a pointer field (P). The KPL command sets the width of these two fields, then partitions the entire array into records, each with a length of K + P bytes. Figure 2 shows the logical model of the CADM data array. The length of K may vary from 1 to 255 bytes, and P may be set between 0 and 255 bytes. The variable record width provides significant flexibility that is very useful for general-purpose data manipulation. It allows complex operations, such as sort and search, to be performed on virtually any type of data. For example, the Am95C85 devices can be used to search a file-allocation table for a particular file address. It may then be reprogrammed to manipulate a disk-directory table. The Am95C85 can sort a database index file and is versatile enough to handle each of the tasks described above, even though each has a differernt record width.

The maximum number of records stored in each CADM depends on the record width (K + P) and the value of Last Address (LA). To efficiently use the memory space of the CADM, the LA should be programmed with the following value:

if mask bytes are used,

$$LA = \{INT[(1024 - 2*K - P)/(K + P)]\} * (K + P) + K - 1$$

if mask bytes are not used,

$$LA = \{INT[(1024 - K - P)/(K + P)]\} * (K + P) - 1$$

The Am95C85 array can be easily expanded if the application requires more record storage; up to 256 CADMs can be cascaded to meet the application requirements. The addition of hardware is transparent to software. The programmer still sees one command port, one data port, one $\overline{STAT}$ pin and one $\overline{DONE}$ pin. The only difference is that there is more record space for data manipulation. The number of CADM devices in cascade can be easily determined by reading the data port after a hardware or software reset.

Addressing Flexibility

To take advantage of the flexibility of the unique memory array, the Am95C85 allows several different addressing modes:

1) Auto-Increment Access
2) Stack Access
3) Indirect Random Access
4) Content-Addressable Access

The programmer will first issue a command that either directly, or by implication, places the Am95C85 CADM in a particular addressing mode. For example, the command AIM allows the host to read or write the currently addressed location, while subsequent reads and writes will be to the next byte (i.e., the CADM auto-increments the address pointer after each data access). Alternatively, STK sets the Stack-Access mode, which means that any subsequent data access physically moves all data below the current location for a read or write. A data read pops the byte at the current location, and moves all the data below up. A data write pushes a byte on the array at the address pointer moving all the data below down. The Stack-Access mode allows for immediate insertion or deletion of records (in previously sorted data), without the need for re-sorting.

The pointer into the memory array, the address pointer, is maintained by the Am95C85, although the programmer can load the address pointer through the use of the LAL (Load Address Long) and LAS (Load Address Short) commands.

The Find (FND) instruction implies a Content-Addressable Access mode. The description of the FND command is "set the address pointer to the key whose value is equal to the following bytes. If not present, point to the next higher value key." Following this instruction, the Am95C85 may be read to acquire the key plus pointer that was found. Since the FND instruction relies on the Am95C85 CADM data being in sorted order, the next section describes how a sort can be accomplished.

If more records matching a particular key value are to be located, additional FND commands without a key following the command can be issued. In this case, the value of the key contained in the input buffer space from the previous FND is used. The Address Pointer is incremented and the key comparisons are performed. This continues with each subsequent FND. To terminate this mode of operation, for instance to allow a new record to be sought, a command other than FND or RRB should be issued. The CADMs will then expect a subsequent FND command to be followed by a new key for which to search.

Host-Independent Sorting

Sorting may be accomplished on data which is in the form of a relational database index file. The programmer sets the length of the key and pointer fields by the KPL command and sets up K bytes of mask if the masking option is used. The data list may be loaded into the Am95C85 devices via DMA or slower programmed I/O. Two methods of sorting are possible:

1) Load data by DMA or I/O and then issue a Sort-Off-Line (SOF) command. This method loads all the data first and then performs the sort. The CPU can be performing other functions during the SOF execution. DMA completion must be detected by software before the SOF command is issued. $\overline{DONE}$ must be detected after the SOF command to signal that the sort has been completed.

2) Sort-On-Line (SON) command, followed by I/O or DMA load, allows each record to be placed in sorted order as it is loaded. If DMA is utilized, the CPU is free toperform other non-CADM tasks during the entire operation. DMA complete, followed by $\overline{DONE}$, defines the end of the sort.

The $\overline{DONE}$ pin signals the acceptance of each byte of data and indicates the device is ready for the next byte. It also signals the completion of the active sort for the SOF command. In the case of SON, after the last byte of each record is received, $\overline{DONE}$ is asserted after the record is merged with existing records. After the last record is sent to the CADM array, the final $\overline{DONE}$ signal represents the end of Sort On Line.

The Hardware Designer's View

Reset

The CADM will go into the reset cycle after the hardware reset is asserted or a software-reset command is issued. Each device in an array will number itself and determine its chip address. The first device with RUP tied to HIGH assumes it has a chip address of 0, the next chip assumes an address of 1, and so on, until the last device with RDWN tied to HIGH numbers itself. Completion of reset is signaled by $\overline{DONE}$ going LOW. After reset, the address pointer is set to the first byte location in the last chip. The key length, K, is set to 1; the pointer length, P, is set to 0, and the last address is set to 1023. Masking is disabled. A hardware reset is required after power-up to bring the internal logic into a known state.

System Interface

All system interface signals are designed to be standard TTL compatible.

The system-control signals, $\overline{RE}$, $\overline{WE}$, $\overline{CS}$, and C/$\overline{D}$ are used to control the interface between the host and the CADM array. The command port access, with $\overline{CS}$ = "LOW" and C/$\overline{D}$ = "HIGH," is used to send commands to the device and is write-only. The data port access, with $\overline{CS}$ = "LOW" and C/$\overline{D}$ = "LOW," is used to transfer data between the host and the CADM array when reading and writing. These control signals should be connected to all the CADM devices in cascade.

The CADM data bus is used for host interface and chip-to-chip data transfer. Because of this, the CADM should not be directly connected to the host data bus. A transceiver is required to isolate the CADM data bus from the host data bus to avoid possible contention.

Two pins indicate the status of the Am95C85. $\overline{DONE}$ is used to indicate the completion of a command execution or data transfer. $\overline{STAT}$ going active indicates an exception condition following the execution of command or data transfer. The host should not drive the CADM data bus when $\overline{DONE}$ is inactive; otherwise, an unexpected outcome may occur. $\overline{DONE}$ may stay inactive forever if an invalid command sequence is issued. In this case, a reset is required to bring $\overline{DONE}$ back to LOW. If there is more than one CADM in cascade, the $\overline{DONE}$ pin from each CADM should be connected together and pulled up through a resistor to the power supply. Similarly, the $\overline{STAT}$ pin from each CADM should be connected together and also tied to the power supply through a pull-up resistor.

The CLK signal should reside between 1 MHz and its maximum rating.

Chip-To-Chip Communications

During the execution of some commands, it may be necessary to transfer data from one chip to another. These signals, TUP, RUP, TDWN, RDWN, and $\overline{GLB}$ are used to perform handshaking between the devices involved in the transfer. RDWN should be connected to the TUP of the next lower order chip, and TDWN should be connected to the RUP of the next lower order chip. The first device should have RUP pulled HIGH through a resistor to the power supply as should RDWN of the last device. Figure 3 shows the signal connections for cascading multiple devices.

Bank-To-Bank Control

As the number of CADMs used in the system increases, the capacitive load seen by each CADM device will increase. Depending on the system environment, up to 16 devices may be cascaded. If the effective load exceeds the specified test load, the designer will have two choices:

1) Reduce the clock frequency to the CADM array.

2) Insert a buffer circuit between banks of CADMs to increase driving capabilities.

If Option 1 is chosen, $\overline{DIRD}$, $\overline{DIRG}$ and $\overline{T}$/R are not used and can be left unconnected. If Option 2 is chosen, $\overline{DIRD}$, $\overline{DIRG}$ and $\overline{T}$/R signals are used to control the direction of buffering circuitry between banks. Even if the designer chooses Option 2, the clock frequency still has to be slowed down from its maximum rating because of buffer delay. The designer must decide which option is best suited to the system.

Command Summary

There are 16 commands to control the operation of the CADM. These commands are used to initialize the CADM, to control the internal pointers, to load the data, and perform sorting and searching. A command is loaded into the command register by writing an operation code into the command port. The command port is used to load the operation code only. For commands that require parameters following the command operation code, the parameters should be loaded through the data port. Commands requiring literal data are: LAS, FND, KPL, SMB, SON, LUD, and LAL. Table 1 summarizes the operation code, mnemonic, and functional description for each command.

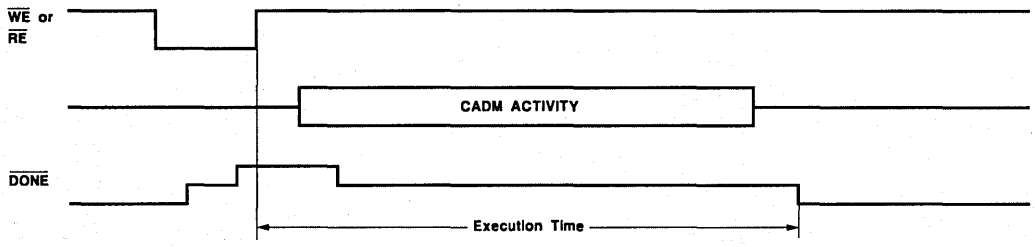
TABLE 1. COMMAND DESCRIPTIONS

OpCode	Mnemonic	Operands	Functional Description
00H	RST		Software reset command.
01H	LAS	Addr (LSB), Addr (MSB)	Load Address Short. Load the following two bytes of data at the address pointer of the currently active device.
02H	DEC		Decrement the address pointer by one.
03H	FND	Key (MSB), ... , Key (LSB) (Note 1)	Find the key specified following this command. Asserted $\overline{STAT}$ LOW if the key is not found.
04H	NXT		Set the address pointer to the first byte of the next record.
05H	RRB		Restore the address pointer to the first byte of the current record.
06H	AIM		Set Auto-Increment Mode. Address pointer is incremented by one after each data read/write.
07H	STK		Set Stack Mode. In Stack-Access mode, a read will pop data out of the data array at the address pointer and a write will push data into data array at the address pointer. The address pointer remains unchanged.
08H	KPL	K, P, LA (LSB), LA (MSB)	Load length of key and pointer fields and set the last address pointer.
09H	SMB	Mask (MSB), ... , Mask (LSB) (Note 1)	Set Mask Byte. The following K bytes of data will be used as mask during sorting and searching.
0AH	SON	Data (MSB), ... , Data (LSB) (Note 2)	Sort On Line. The CADM will insert the record into the data array in sorted order after the last byte of the record is loaded.
0BH	LUD	Data (MSB), ... , Data (LSB), ... (Note 3)	Load Unsorted Data. Data loaded following this command will be placed in the locations after existing meaningful data, if there is any.
0CH	SOF		Sort Off Line. Sort the existing data in the CADM in ascending order.
0DH	LAL	Addr (LSB), Addr (MSB), Chip Addr	Load Address Long. Load the following two bytes at the address pointer of the chip whose number is specified by the third byte.
0EH	PRE		Set the address pointer to the first byte of the previous record.
0FH	GSF		Get Status Full. Asserted $\overline{STAT}$ LOW if the CADM record space is full.

Notes: 1. Requires K Bytes following Opcode.
2. Requires integer multiples of (K + P) Bytes. Execution begins after each (K + P) Bytes are written.
3. Requires integer multiples of (K + P) Bytes.

Command Execution Time

The execution time of each command is expressed in clock cycles per byte of transfer. The execution times are measured from $\overline{WE}$ or $\overline{RE}$ to $\overline{DONE}$ as shown below.



WF024170

Command	Clock Cycles per Byte	Conditions
LUD	6	Command
	6	Per byte within a chip
	7	If crosses chip boundary
SMB	8	Command: first occurrence
	7	Command: all other occurrences
	6	Per mask byte for first k-1 bytes
	8	For last byte
SON	8	Command
	6	For first K + P – 1 bytes
	16 + S	For last byte (where S = binary search time; see FND performance equation)
AIM (Read/Write in AIM mode)	6	Command
	6	If on same chip
	9	If crosses chip boundary
KPL	9	Command; (+ 1 if only one chip in system)
	6	K: (+ 1 if user erroneously sets K = 0)
	7 + 2 * (9 – x)	P: where x = number of lower order zeros in K + P (binary)
	5	LA: (lsb)
	7	LA: (msb)
SOF	(See Sort-Off-Line performance equation)	
RST	8 + 4 * N	Where N = number of chips in system (Note: this applies to hardware and software RESETs).
LAL	4	Command
	5	lsb
	4	msb.
	7	chip
LAS	4	Command
	5	lsb
	7	msb
DEC	8	If on same chip
	10	If crosses chip boundary
PRE	10	If on same chip
	13	If crosses chip boundary
RRB	7	Command
GSF	6	Command
FND	(See FIND performance equation)	
NXT	11	If on same chip
	14	If crosses chip boundary
STK	6	Command
–Push– (data write in Stack mode)	8	If on same chip. Add one clock cycle for every chip boundary crossing.
–Pop– (data read in Stack mode)	14	If on same chip. Add one clock cycle for every chip boundary crossing.

2

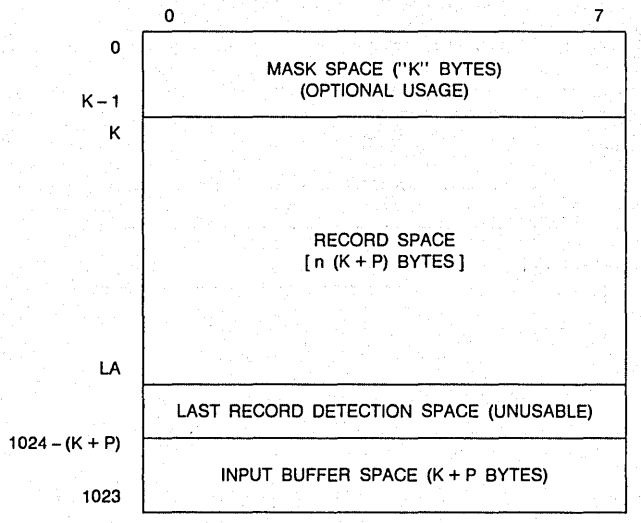


Figure 1. Am95C85 CADM Physical Model

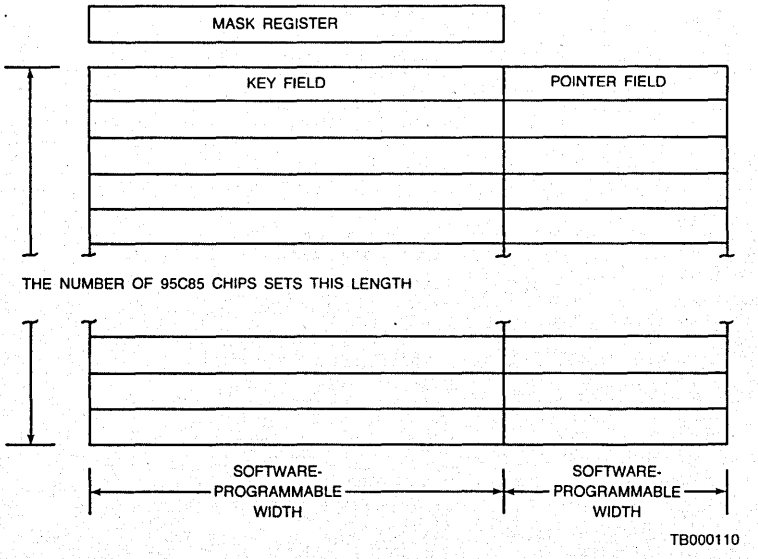


Figure 2. Am95C85 CADM Programmer's Model

Performance

"FND (Find)" Performance Equation

The "Find" performance equation assumes that the non-matching keys are different from the search key (in the most significant byte), and that the first match is found at the end of the longest possible binary search. The Find command's binary search is executed in parallel by all the CADM devices in an array. The device that finds the first occurrence terminates the operation by pulling $\overline{DONE}$ LOW. This explains why, for multi-chip arrays, the "Find" performance is independent of the total number of records. This equation includes the time required to load the search key.

"SOF (Sort-Off-Line)" Performance Equations

The Am95C85 CADM sorting performance is data-dependent. Best-case performance, quickest sort, is achieved from previously sorted data with no matching most significant bytes. The data which takes the longest time to sort is already sorted in reverse or descending order, and contains matching most significant bytes, where only the least significant bytes differ. The following two equations establish performance bounds for these two extremes.

"FIND" PERFORMANCE EQUATION

$$T_F = \frac{39 + 5K + (5.5 + 3K)\ (\lfloor \log_2 (n) \rfloor + 1)}{F}$$

"SORT OFF-LINE" PERFORMANCE EQUATIONS

Best-Case Performance:

$$T_{SB} = \frac{9 + N\left[\ 20 + 6\ (K + P) + 8.5\ (\lfloor \log_2\ (n + 1) \rfloor)\right]}{F}$$

Worst-Case Performance:

$$T_{SW} = \frac{9 + N\left[21 + \left(9 + \left\lceil \frac{N}{n} \right\rceil\right)\ (K + P) + (\lfloor \log_2 (n) \rfloor + 1)\ (5.5 + 3K)\right]}{F}$$

Where:

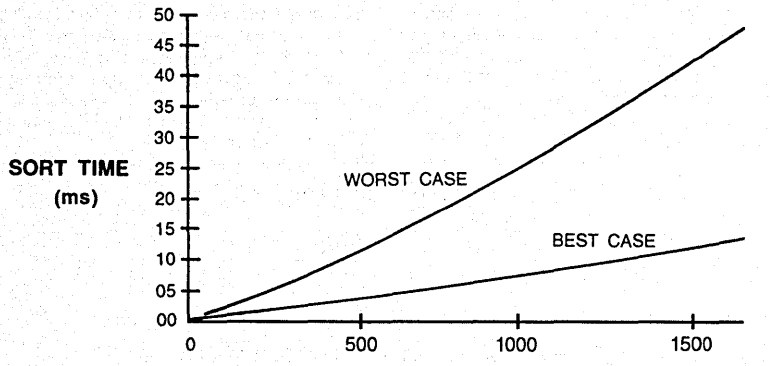
- N = Total no. of records
- n = No. of records in each chip
- K = No. of bytes/key
- P = No. of bytes/pointer
- F = Frequency of Am95C85 clock
- $\lfloor\ \rfloor$ = Truncate notation
- $\lceil\ \rceil$ = Round up notation
- T_{SB} = Time for sort (best case)
- T_{SW} = Time for sort (worst case)
- T_F = Time for find

$$n = \left\lfloor \frac{1024 - K - P - M}{K + P} \right\rfloor$$

M = K (if masking is used)
0 (if masking not used)

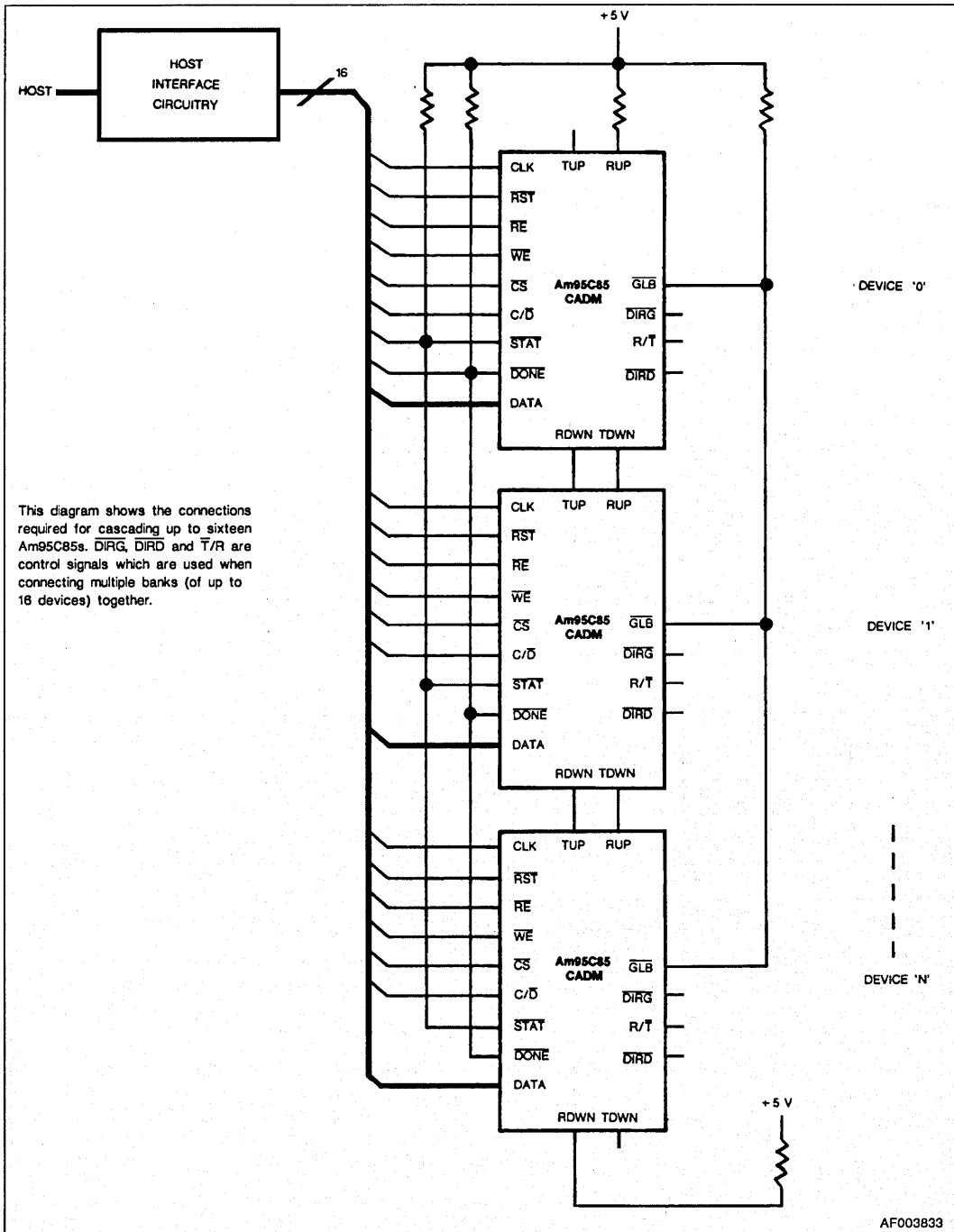
Am95C85 CADM SORT PERFORMANCE

(K = 8, P = 2)



OP001821

For This Case:
$T_F = 6.4\ \mu s$
CLK = 16 MHz



This diagram shows the connections required for cascading up to sixteen Am95C85s. $\overline{DIRG}$, $\overline{DIRD}$ and $\overline{T}/R$ are control signals which are used when connecting multiple banks (of up to 16 devices) together.

Figure 3. Am95C85 Cascade Circuit

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to GND ..... −0.5 to +7

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	16 MHz, 12 MHz		Units
			Min.	Max.	
V_{OH}	Output HIGH Voltage	I_{OH} = −12 mA (I_{OH} = −1 mA for TUP, TDWN)	2.4	V_{CC}	V
V_{OL}	Output LOW Voltage	I_{OL} = 12 mA (I_{OH} = 1 mA for TUP, TDWN)	0	0.45	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + 0.5	V
V_{IL}	Input LOW Voltage		−0.5	0.8	V
I_{IL}	Input Leakage Current	0 V < V_{IN} < V_{CC}		±10	μA
I_{LO}	Output Leakage Current	0 V < V_{OUT} < V_{CC}		±10	μA
I_{CC}	Maximum Average Power Supply Current	V_{CC} = 5.5 V, 16 MHz, Outputs loaded, worst-case data shifts during Push		200	mA
I_{CCS}	Maximum Average Power Supply Standby Current	V_{CC} = 5.5 V, 16 MHz, Outputs unloaded, No-ops		125	mA
V_{CC}	Power Supply Voltage		4.5	5.5	V

CAPACITANCE*

Parameter Symbol	Parameter Description	Test Conditions	16 MHz, 12 MHz		Units
			Min.	Max.	
C_i	Input Capacitance ($\overline{RE}$, $\overline{WE}$, $\overline{CS}$, C/$\overline{D}$, $\overline{RST}$, RUP, RDWN, CLK)	F_C = 1 MHz, V_{CC} = 0 V, GND = 0 V, unmeasured pins floating		8	pF
C_o	Output Capacitance (TUP, TDWN, R/$\overline{T}$)	F_C = 1 MHz, V_{CC} = 0 V, GND = 0 V, unmeasured pins floating		10	pF
C_{io}	I/O Capacitance ($\overline{DONE}$, $\overline{DIRD}$, $\overline{GLB}$, $\overline{DIRG}$, $\overline{STAT}$, DATABUS)	F_C = 1 MHz, V_{CC} = 5 V, GND = 0 V, unmeasured pins floating		12	pF

*The capacitance values are guaranteed by design and are not tested.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

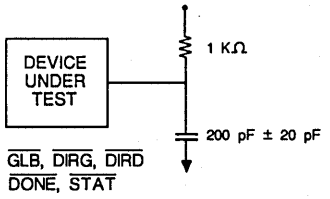
No.	Parameter Symbol	Parameter Description	16 MHz		12 MHz		Units
			Min.	Max.	Min.	Max.	
SYSTEM-TO-CADM TIMINGS							
1	t_{CC}	CLK Period	62	1000	82	1000	ns
2	t_{CH}	CLK HIGH Time	26		36		ns
3	t_{CL}	CLK LOW Time	26		36		ns
4	t_{DS}	Data Setup Before CLK LOW for Write	5		5		ns
5	t_{DH}	Data Hold After CLK LOW for Write	24		24		ns
6	t_{WS}	$\overline{WE}$ Setup Before CLK LOW	10		10		ns
7	t_{WW}	$\overline{WE}$ Pulse Width	86		106		ns
8	t_{WH}	$\overline{WE}$ Hold After CLK LOW	14		14		ns
9	t_{RS}	$\overline{RE}$ Setup Before CLK LOW	10		10		ns
10	t_{RR}	$\overline{RE}$ Pulse Width	86		106		ns
11	t_{RH}	$\overline{RE}$ Hold After CLK LOW	14		14		ns
12	t_{CSS}	$\overline{CS}$ Setup Before CLK LOW	10		10		ns
13	t_{CSW}	$\overline{CS}$ Pulse Width	86		106		ns
14	t_{CSH}	$\overline{CS}$ Hold After CLK LOW	14		14		ns
15	t_{CDS}	C/$\overline{D}$ Setup Before CLK LOW	10		10		ns
16	t_{CDW}	C/$\overline{D}$ Pulse Width	86		106		ns
17	t_{CDH}	C/$\overline{D}$ Hold After CLK LOW	14		14		ns
18	t_{SS}	$\overline{RST}$ Setup Before CLK LOW	10		10		ns
19	t_{SW}	$\overline{RST}$ Pulse Width	210		270		ns
20	t_{SH}	$\overline{RST}$ Hold After CLK LOW	14		14		ns
CADM-TO-SYSTEM TIMINGS							
21	t_{LDVR}	CLK LOW to Data Valid for Read		26		37	ns
22	t_{HDTR}	CLK HIGH to Data Three-State for Read		20		30	ns
23	t_{CHDH}	CLK HIGH to $\overline{DONE}$, $\overline{DIRD}$ HIGH		20		30	ns
24	t_{CLDT}	CLK LOW to $\overline{DONE}$, $\overline{DIRD}$ Three State		20		30	ns
25	t_{CHDL}	CLK HIGH to $\overline{DONE}$, $\overline{DIRD}$ LOW		20		30	ns
26	t_{CHSH}	CLK HIGH to $\overline{STAT}$ HIGH		20		30	ns
27	t_{CLST}	CLK LOW to $\overline{STAT}$ Three State		20		30	ns
28	t_{CHSL}	CLK HIGH to $\overline{STAT}$ LOW		20		30	ns
29	t_{CLRL}	CLK LOW to R/$\overline{T}$ LOW		26		37	ns
30	t_{CHRH}	CLK HIGH to R/$\overline{T}$ HIGH		20		30	ns

SWITCHING CHARACTERISTIC (continued)

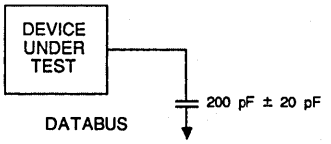
No.	Parameter Symbol	Parameter Description	16 MHz		12 MHz		Units
			Min.	Max.	Min.	Max.	
CADM-TO-CADM TIMINGS							
31	t_{CHRL}	CLK HIGH to R/$\overline{T}$ LOW for Interchip Data Move		20		30	ns
32	t_{HDVI}	CLK HIGH to Data Valid for Interchip Data Move	5	20	5	30	ns
33	t_{HDTI}	CLK HIGH to Data Three State for Interchip Data Move		20		30	ns
34	t_{CHGH}	CLK HIGH to $\overline{GLB}$ HIGH		20		30	ns
35	t_{CLGT}	CLK LOW to $\overline{GLB}$ Three State		20		30	ns
36	t_{CHGL}	CLK HIGH to $\overline{GLB}$, $\overline{DIRG}$ LOW		20		30	ns
37	t_{LDGH}	CLK LOW to $\overline{DIRG}$ HIGH		26		37	ns
38	t_{HDGT}	CLK HIGH to $\overline{DIRG}$ Three State		20		30	ns
39	t_{HTUH}	CLK HIGH to TUP HIGH		20		30	ns
40	t_{HTUL}	CLK HIGH to TUP LOW		20		30	ns
41	t_{LTUL}	CLK LOW to TUP LOW		15		20	ns
42	t_{RDSL}	RDWN Setup Before CLK LOW	6		6		ns
43	t_{RDHL}	RDWN Hold After CLK LOW	26		36		ns
44	t_{RUSL}	RUP Setup Before CLK LOW	6		6		ns
45	t_{RUHL}	RUP Hold After CLK LOW	26		36		ns
46	t_{HTDH}	CLK HIGH to TDWN HIGH		20		30	ns
47	t_{HTDL}	CLK HIGH to TDWN LOW		20		30	ns
48	t_{LTDH}	CLK LOW to TDWN HIGH		15		20	ns
49	t_{RUDT}	RUP HIGH to Data Three State	0	15	0	20	ns
50	t_{RURH}	RUP HIGH to R/$\overline{T}$ HIGH (for Pop)	0	15	0	25	ns
51	t_{RDDT}	RDWN LOW to Data Three State	0	15	0	20	ns
52	t_{RDRH}	RDWN LOW to R/$\overline{T}$ HIGH (for Push)	0	15	0	25	ns
53	t_{DHTD}	Data Hold After TDWN HIGH for Interchip Data Move (for Pop)	0		0		ns
54	t_{DHTU}	Data Hold After TUP LOW for Interchip Data Move (for Push)	0		0		ns
55	t_{DSL}	$\overline{DONE}$ Setup Before CLK LOW	6		6		ns
56	t_{DHL}	$\overline{DONE}$ Hold After CLK LOW	26		36		ns
57	t_{GSL}	$\overline{GLB}$ Setup Before CLK LOW	6		6		ns
58	t_{GHL}	$\overline{GLB}$ Hold After CLK LOW	26		36		ns

SWITCHING TEST CIRCUITS

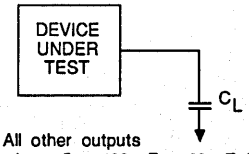
AC Loading



TC003982



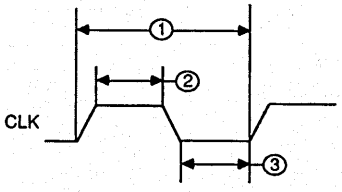
TC004002



All other outputs
where C_L= 100 pF ± 20 pF for R/$\overline{T}$
C_L= 20 pF for TUP, TDWN

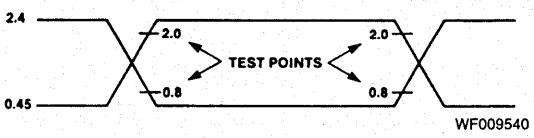
TC003991

SWITCHING TEST WAVEFORMS



WF023670

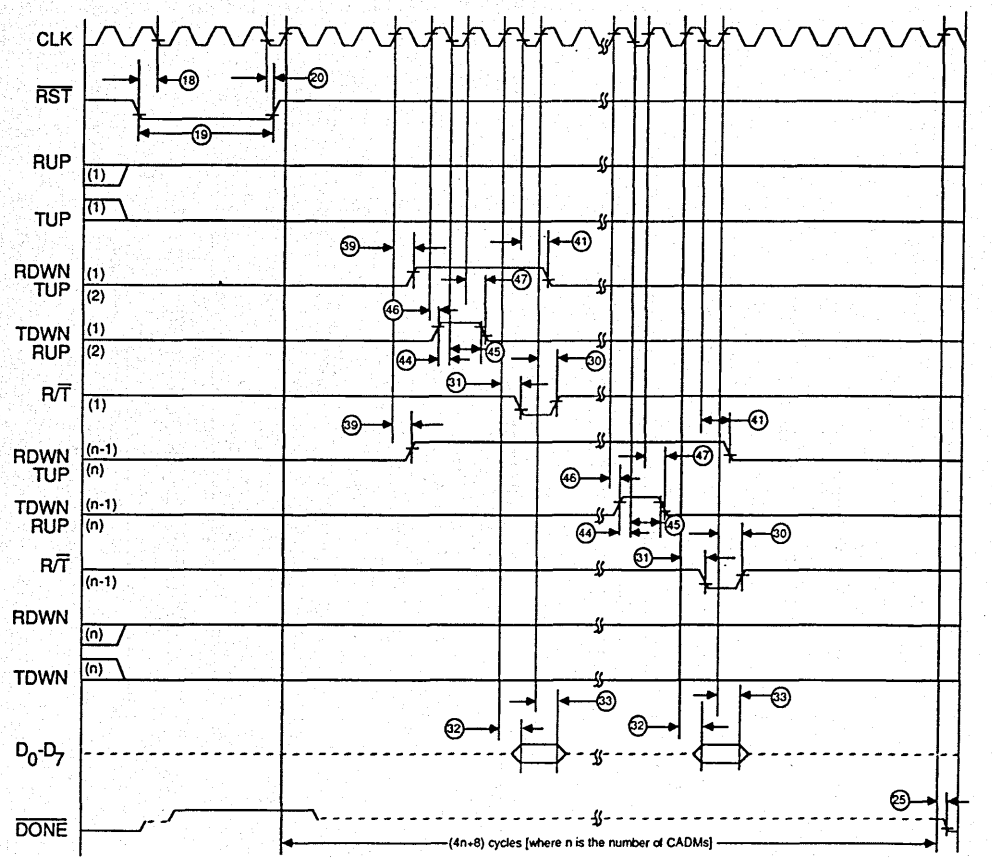
Clock



WF009540

Input/Output

SWITCHING WAVEFORMS

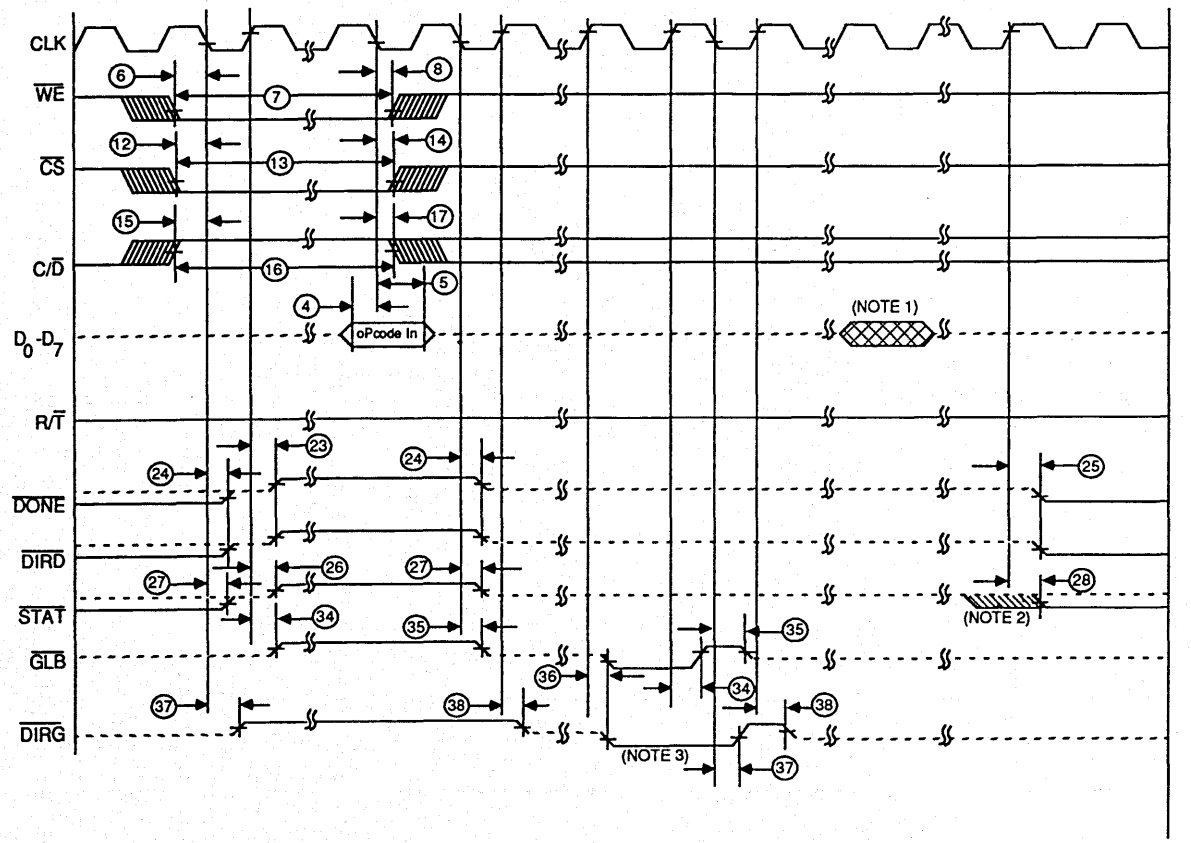


WF024040

Hardware Reset Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

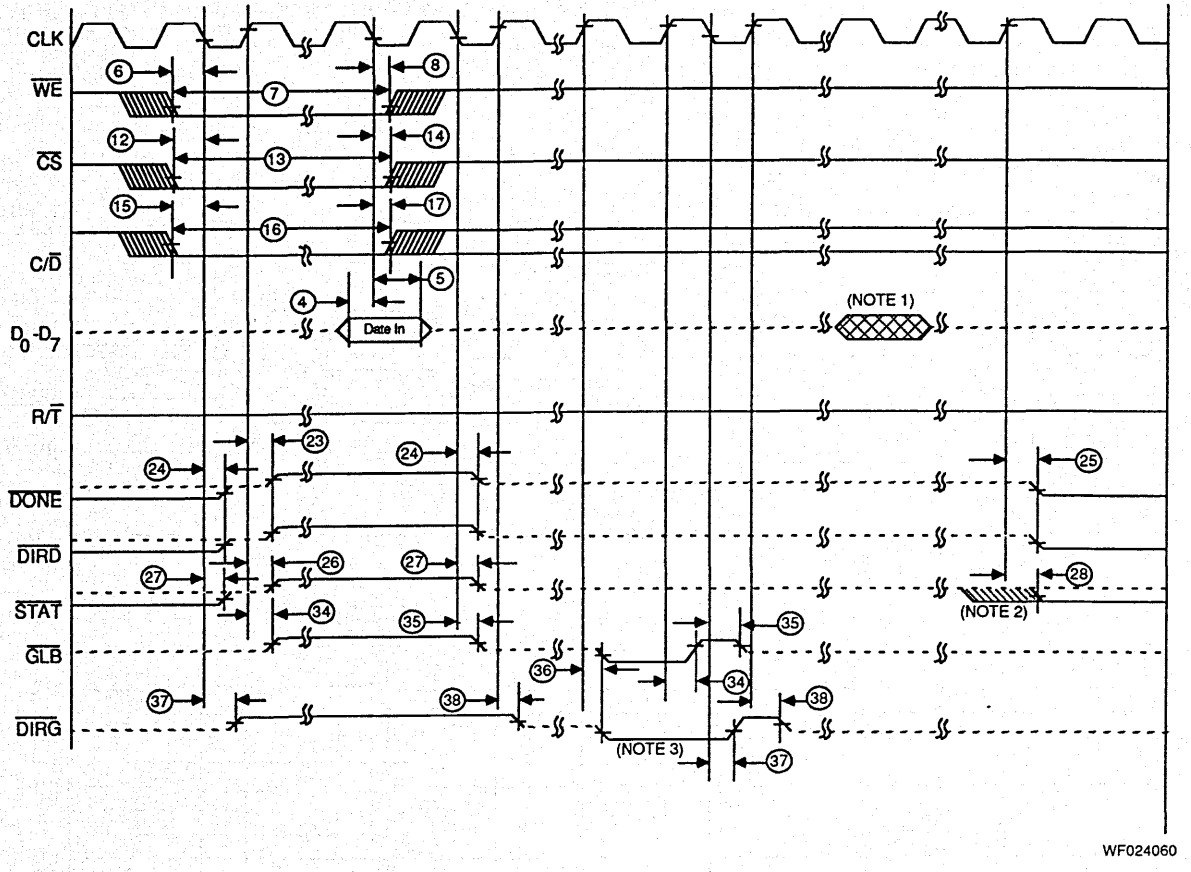
SWITCHING WAVEFORMS (continued)



Command Write Timing

Notes: 1. Any CADM may drive the data bus at any time during the command.
2. $\overline{STAT}$ is indeterminate one cycle before $\overline{DONE}$. It must be qualified with $\overline{DONE}$ being asserted.
3. $\overline{GLB}$ and $\overline{DIRG}$ may occur multiple times or not at all during the command.

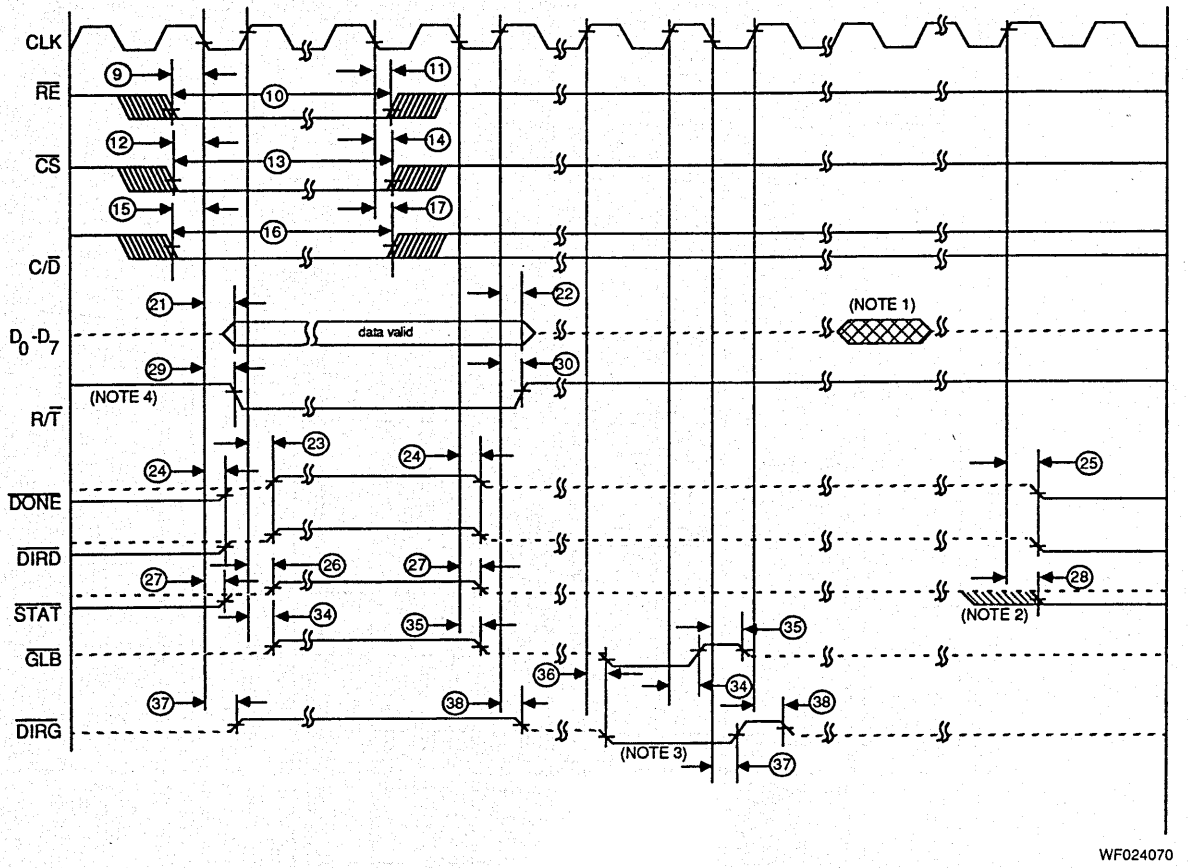
SWITCHING WAVEFORMS (continued)



Data Write Timing

Notes: 1. Any CADM may drive the data bus at any time during the write.
2. STAT is indeterminate one cycle before DONE. It must be qualified with DONE being asserted.
3. GLB and DIRG may occur multiple times or not at all during the write.

SWITCHING WAVEFORMS (continued)

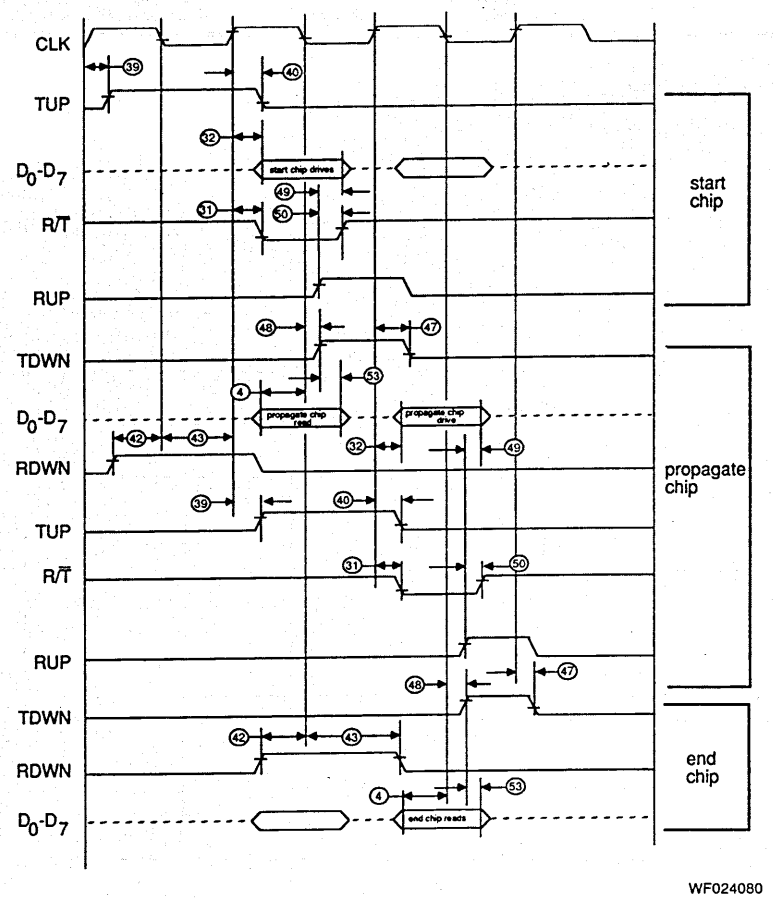


WF024070

Data Read Timing

Notes: 1. Any CADM may drive the data bus at any time during the read.
2. $\overline{STAT}$ is indeterminate one cycle before $\overline{DONE}$. It must be qualified with $\overline{DONE}$ being asserted.
3. $\overline{GLB}$ and $\overline{DIRG}$ may or may not occur during the read.
4. TUP and TDWN are not guaranteed to be LOW during the cycle when $\overline{T}$/R switches LOW during a read.

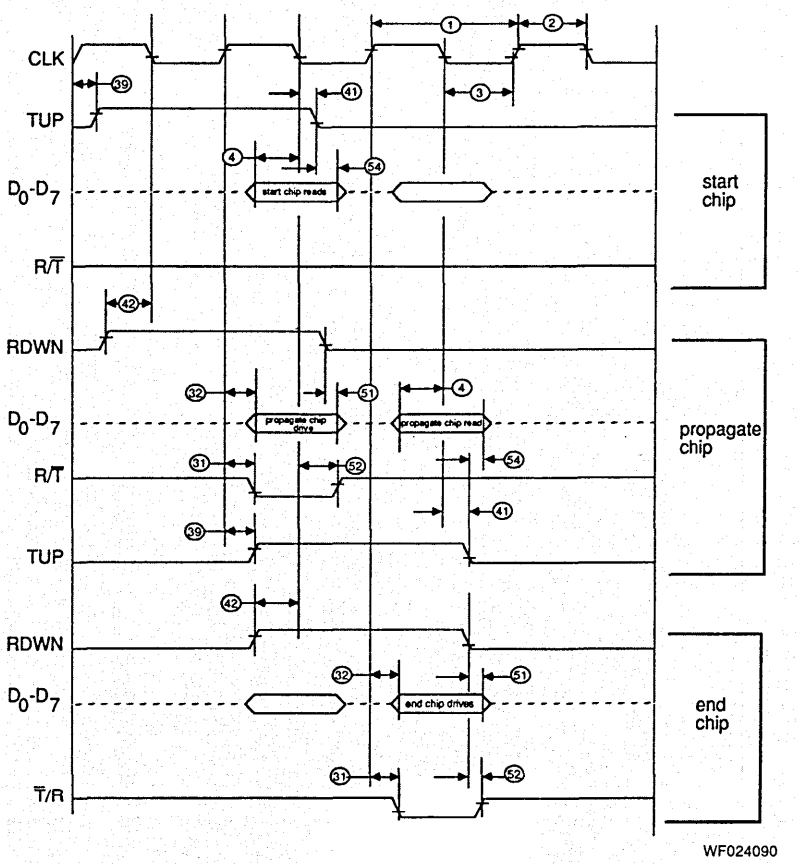
SWITCHING WAVEFORMS (continued)



Pop Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

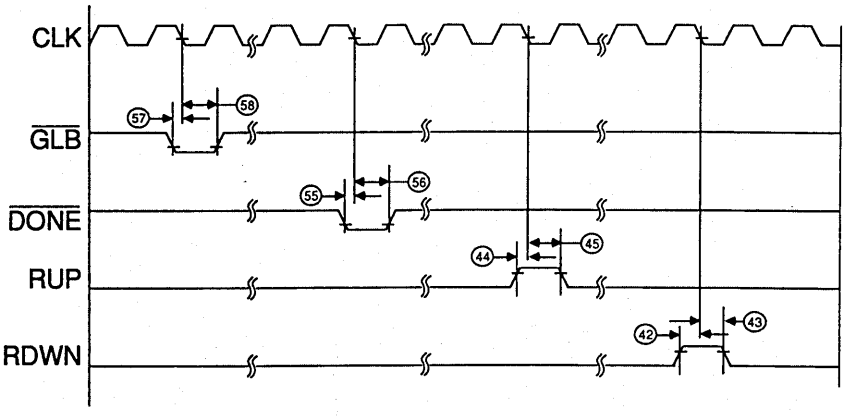
SWITCHING WAVEFORMS (continued)



Push Timing

Note: TUP and TDWN are guaranteed to be LOW only when the CLK is LOW.

SWITCHING WAVEFORMS (continued)

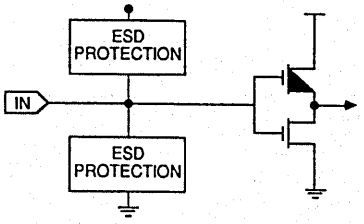


WF024100

Drive $\overline{\text{GLB}}$, $\overline{\text{DONE}}$, RUP, RDWN Timing

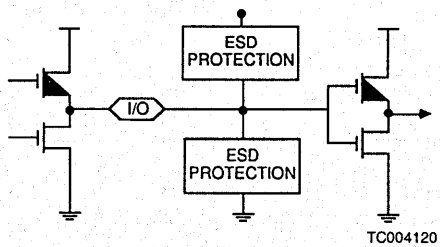
INPUT/OUTPUT CIRCUIT DIAGRAMS

Inputs: $\overline{RE}$, $\overline{WE}$, $\overline{CS}$, C/$\overline{D}$, $\overline{RST}$, RDWN, RUP



TC004110

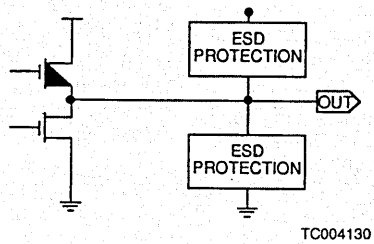
Bi-Directional: $\overline{DONE}$, D0-7, $\overline{GLB}$, TUP, TDWN*



TC004120

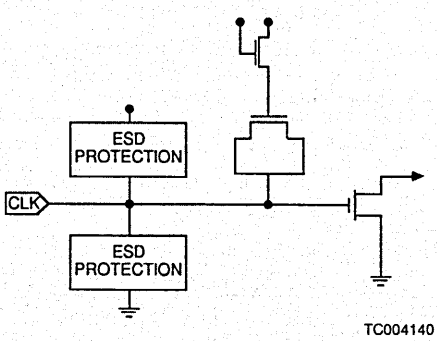
*TUP and TDWN are inputs for test mode only

Outputs: $\overline{STAT}$, $\overline{DIRG}$, $\overline{DIRD}$, R/$\overline{T}$



TC004130

Clocks: CLK



TC004140

Final

Z8530H

Serial Communications Controller (SCC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Two 0- to 2- Mbps full duplex serial channels**
  Each channel has independent oscillator, baud-rate generator, and PLL for clock recovery, dramatically reducing external components.
- **Programmable protocols**
  NRZ, NRZI, and FM data encoding are supported under program control.
- **Programmable Asynchronous Modes**
  Five- to 8-bit characters with programmable stop bits, clock, break detect, and error conditions.
- **Programmable Synchronous Modes**
  SDLC and HDLC and SDLC loop supported with frame control, zero insertion and deletion, abort, and residue handling. CRC-16 and CCITT generators and checkers.
- **Compatible with non-multiplexed bus**
  The Z8530H interfaces easily to most other CPUs.
- **Enhanced Version**
  The Z8530H is an enhanced version whose features include 8-MHz operation and an improved Valid Access Recovery Time (t_{RC}) specification.

GENERAL DESCRIPTION

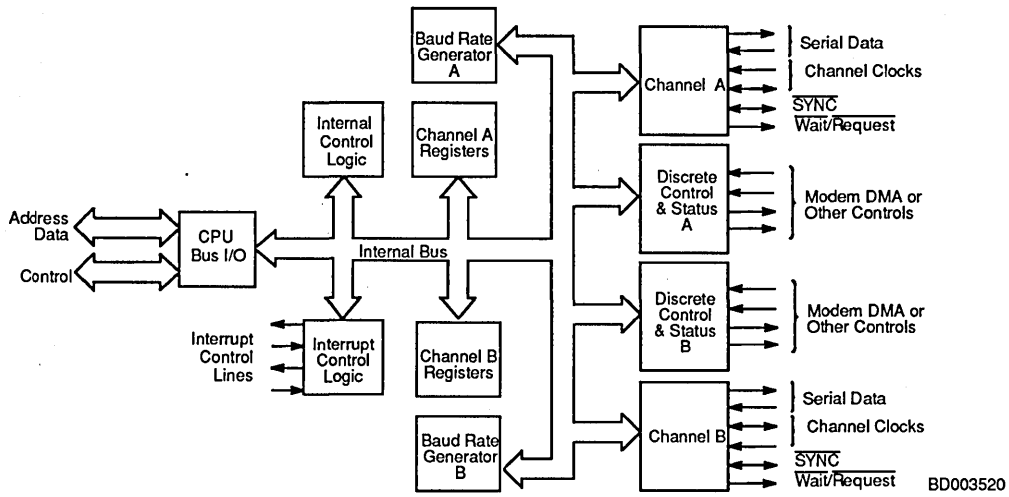
The SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with 8- and 16-bit microprocessors. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions, including on-chip baud rate generators, digital phase-locked loops, and crystal oscillators, which dramatically reduce the need for external logic.

The SCC handles asynchronous formats, synchronous byte-oriented protocols, such as IBM® Bisync, and synchronous bit-oriented protocols, such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (cassette, diskette, tape drivers, etc.).

The device can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The Z8530H is designed for non-multiplexed buses and is easily interfaced with most other CPUs, such as 8080, Z80, 6800, 68000, and MULTIBUS.™

BLOCK DIAGRAM



Publication # 00970 Rev. F Amendment /0
Issue Date: May 1989

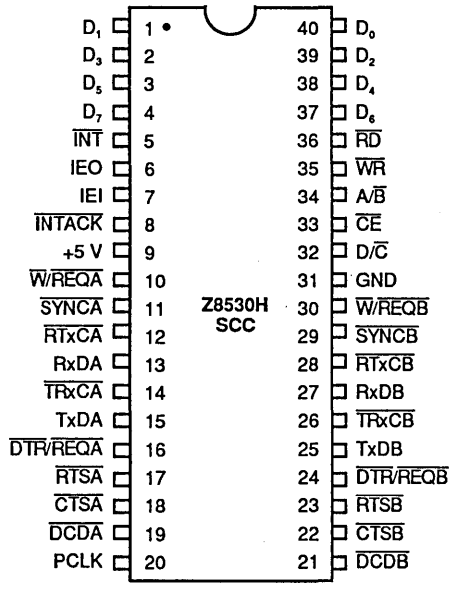
RELATED AMD PRODUCTS

Part No.	Description	Part No.	Description
Am79C12	Full Duplex 1200- bps Modem	80286	High-Performance 16-Bit Microprocessor
Am7960	Coded Data Transceiver		
80186	Highly Integrated 16-Bit Microprocessor	Am9517A	DMA Controller
80188	Highly Integrated 8-Bit Microprocessor		

CONNECTION DIAGRAMS

Top View

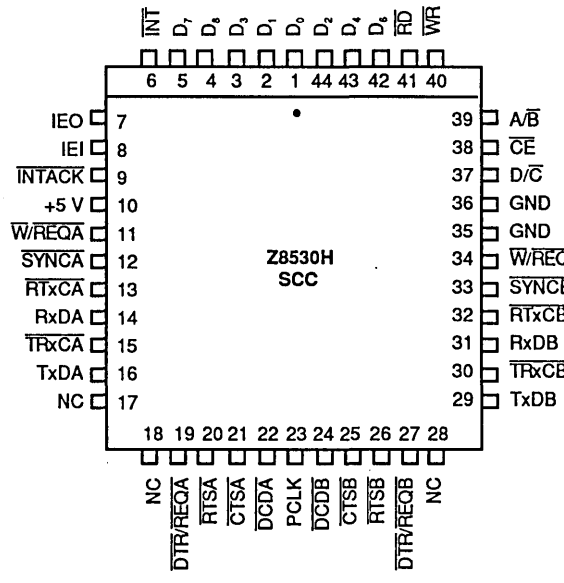
DIP



CD005361

Note: Pin 1 is marked for orientation.

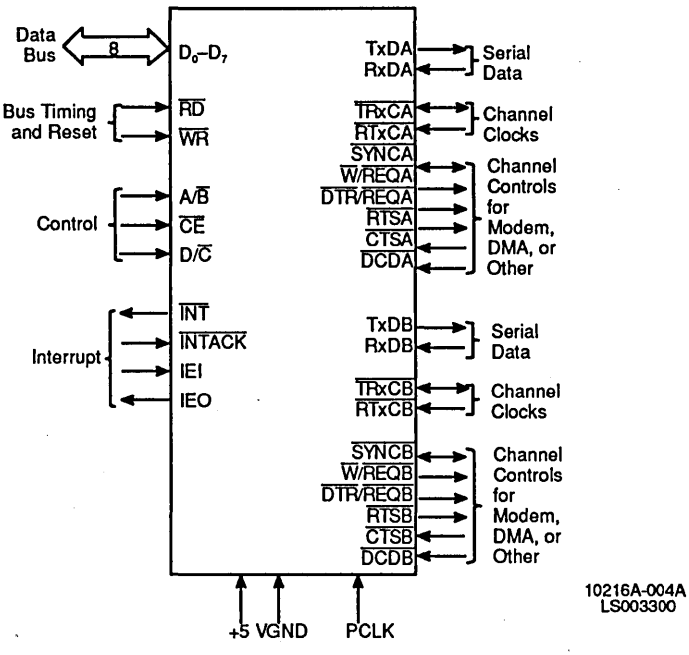
PLCC



CD010931

2

LOGIC SYMBOL

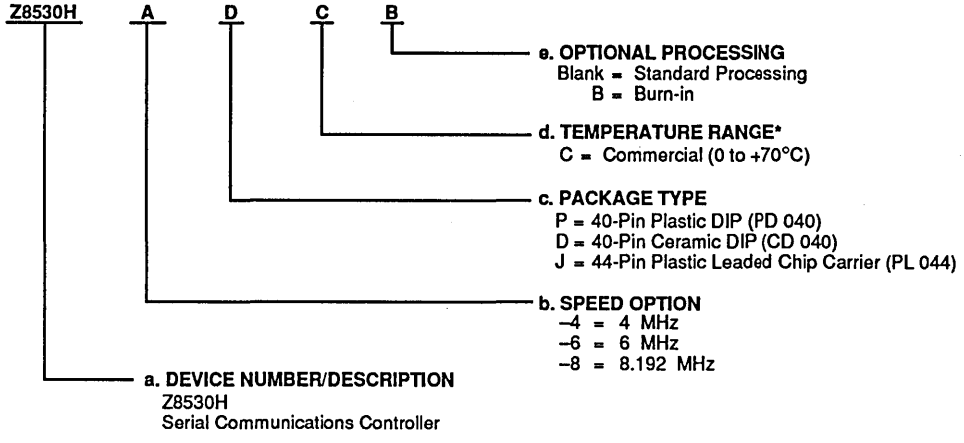


ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Z8530H-4 Z8530H-6 Z8530H-8	PC, DC, DCB, JC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/O) for electrical performance characteristics.

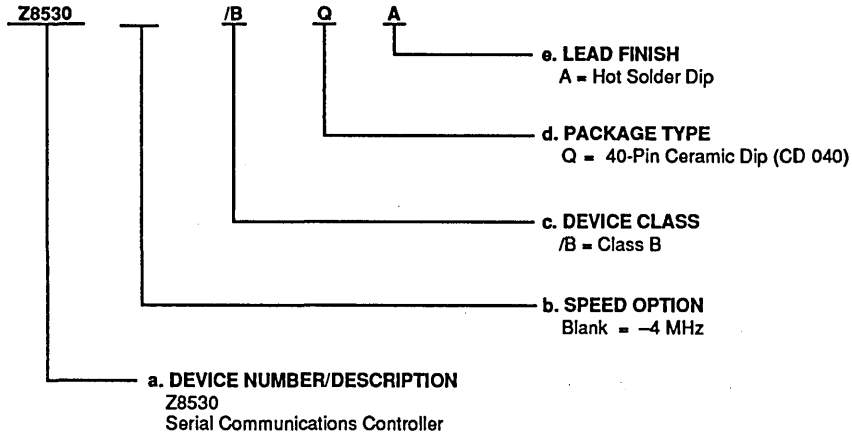
2

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
Z8530	/BQA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

V_{cc}
+5-V Power Supply

GND
Ground

A/$\overline{B}$
Channel A/$\overline{\text{Channel B}}$ Select (Input)

This signal selects the channel in which the read or write operation occurs.

$\overline{CE}$
$\overline{\text{Chip Enable}}$ (Input; Active Low)

This signal selects the SCC for a read or write operation.

$\overline{CTSA}$, $\overline{CTSB}$
$\overline{\text{Clear to Send}}$ (Inputs; Active Low)

If these pins are programmed as Auto Enables, a Low on the inputs enables their respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

D/$\overline{C}$
Data/$\overline{\text{Control}}$ Select (Input)

This signal defines the type of information transferred to or from the SCC. A High means data is transferred; a Low indicates a command.

$\overline{DCDA}$, $\overline{DCDB}$
$\overline{\text{Data Carrier Detect}}$ (Inputs; Active Low)

These pins function as receiver enables if they are programmed for Auto Enables; otherwise, they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

D_0–D_7
Data Bus (Input/Output; Three-State)

These lines carry data and commands to and from the SCC.

$\overline{DTR/REQA}$, $\overline{DTR/REQB}$
$\overline{\text{Data Terminal Ready/Request}}$
(Outputs; Active Low)

These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request lines for a DMA controller.

IEI
Interrupt Enable In (Input; Active High)

IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

IEO
Interrupt Enable Out (Output; Active High)

IEO is High only if IEI is High and the CPU is not servicing an SCC interrupt or the SCC is not requesting an interrupt (Interrupt Acknowledge Cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

$\overline{INT}$
$\overline{\text{Interrupt Request}}$ (Output; Active Low, Open Drain)

This signal is activated when the SCC requests an interrupt.

$\overline{INTACK}$
$\overline{\text{Interrupt Acknowledge}}$ (Input; Active Low)

This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of PCLK.

PCLK
Clock (Input)

This is the master SCC clock used to synchronize internal signals; PCLK is a TTL- level signal.

$\overline{RD}$
$\overline{\text{Read}}$ (Input; Active Low)

This signal indicates a read operation and, when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

RxDA, RxDB Receive Data (Inputs; Active High)

These input signals receive serial data at standard TTL levels.

$\overline{RTxCA}$, $\overline{RTxCB}$
$\overline{\text{Receive/Transmit Clocks}}$
(Inputs; Active Low)

These pins can be programmed in several different modes of operation. In each channel, $\overline{RTxC}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the digital phase-locked loop. These pins can also be programmed for use with the respective $\overline{SYNC}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

$\overline{RTSA}$, $\overline{RTSB}$
$\overline{\text{Request to Send}}$ (Outputs; Active Low)

When the Request to Send (RTS) bit in Write Register 5 is set, the $\overline{RTS}$ signal goes Low. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. In SYNC mode or in asynchronous mode with Auto Enable off, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

$\overline{SYNCA}$, $\overline{SYNCB}$
$\overline{\text{Synchronization}}$ (Inputs/Outputs; Active Low)

These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transi-

tions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 but have no other function.

In the External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{SYNC}$ must be driven Low two receive clock cycles after the last bit in the SYNC character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{SYNC}$.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which SYNC characters are recognized. The SYNC condition is not latched, so these outputs are active each time a SYNC pattern is recognized (regardless of character boundaries). In the SDLC mode, these pins act as outputs and are valid on receipt of a flag.

TxDA, TxDB
Transmit Data (Outputs; Active High)

These output signals transmit serial data at standard TTL levels.

$\overline{TRxCA}$, $\overline{TRxCB}$
Transmit/Receive Clocks
(Inputs/Outputs; Active Low)

These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

$\overline{WR}$
Write (Input; Active Low)

When the SCC is selected, this signal indicates a write operation. The coincidence of $\overline{RD}$ and $\overline{WR}$ is interpreted as a reset.

$\overline{W/REQA}$, $\overline{W/REQB}$
Wait/Request (Outputs; Open drain when programmed for a Wait function, driven High or Low when programmed for a Request function)

These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

ARCHITECTURE

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a non-multiplexed CPU bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface with modems or other external devices (see Block Diagram).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two SYNC character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a write-only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (A only), and one containing the Interrupt Pending bits (A only).

The registers for each channel are designated as follows:

WR0–WR15—Write Registers 0 through 15.

RR0–RR3, RR10, RR12, RR13, RR15—Read Registers 0 through 3, 10, 12, 13, 15.

Table 1 lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel.

Data Path

The transmit and receive data path illustrated in Figure 1 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data are routed through one of several paths (data or CRC) depending on the selected mode (the character length in asynchronous modes also determines the data path).

The transmitter has an 8-bit transmit data buffer register loaded from the internal data bus and a 20-bit transmit shift register that can be loaded either from the sync-character registers or from the transmit data register. Depending on the operational mode, outgoing data are routed through one of four main paths before being transmitted from the Transmit Data output (TxD).

2

Table 1. Read and Write Register Functions

Read Register Functions		Write Register Functions	
RR0	Transmit/Receive buffer status and External status	**WR0**	CRC initialize, initialization commands for the various modes, shift right/shift left command
RR1	Special Receive Condition status	**WR1**	Transmit/Receive Interrupt and data transfer mode definition
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)	**WR2**	Interrupt vector (accessed through either channel)
RR3	Interrupt Pending bits (Channel A only)	**WR3**	Receive parameters and control
RR8	Receive buffer	**WR4**	Transmit/Receive miscellaneous parameters and modes
RR10	Miscellaneous status	**WR5**	Transmit parameters and controls
RR12	Lower byte of baud rate generator time constant	**WR6**	Sync characters or SDLC address field
RR13	Upper byte of baud rate generator time constant	**WR7**	Sync character or SDLC flag
RR15	External/Status interrupt information	**WR8**	Transmit buffer
		WR9	Master interrupt control and reset (accessed through either channel)
		WR10	Miscellaneous transmitter/receiver control bits
		WR11	Clock mode control
		WR12	Lower byte of baud rate generator time constant
		WR13	Upper byte of baud rate generator time constant
		WR14	Miscellaneous control bits
		WR15	External/Status interrupt control

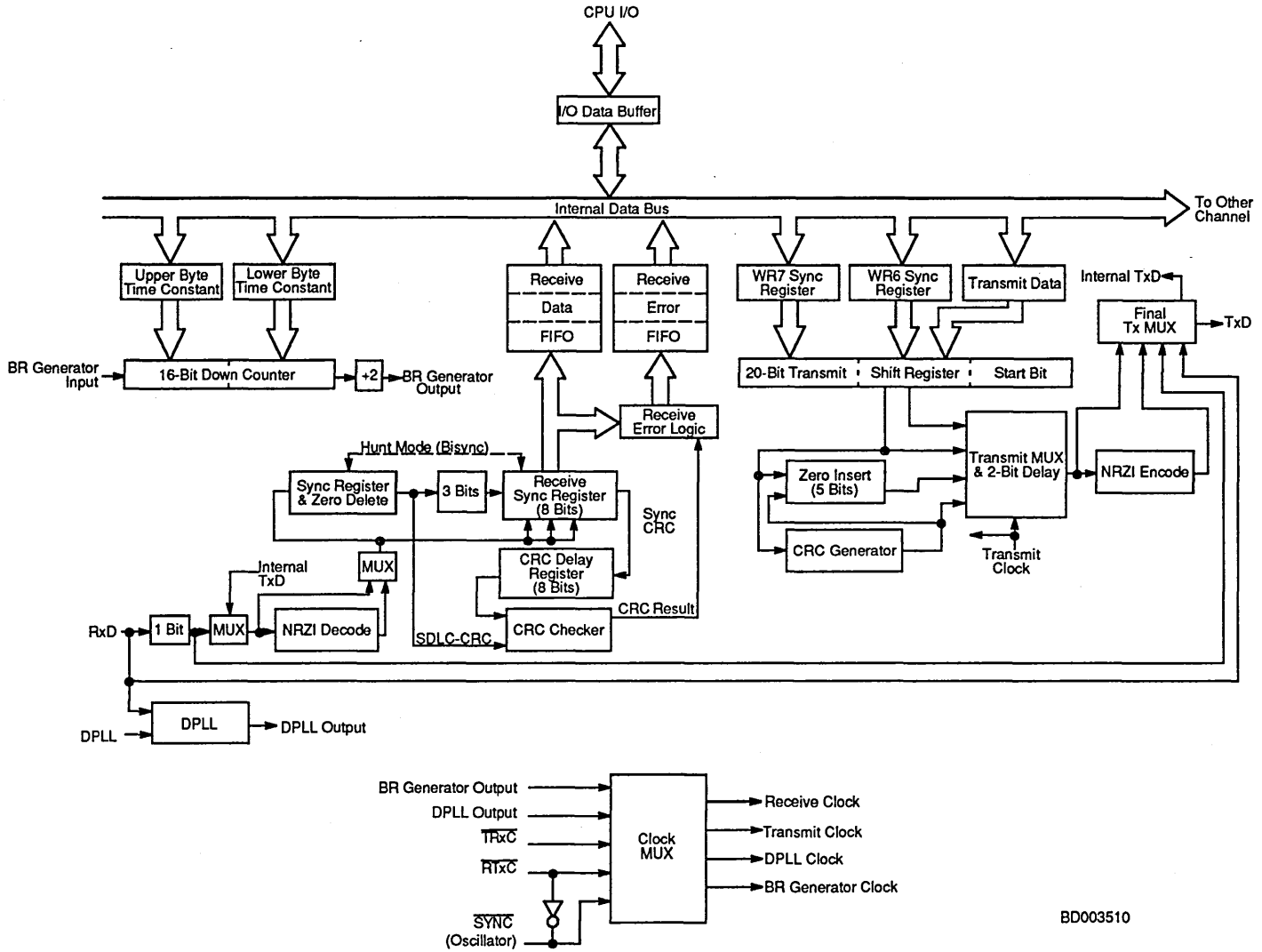


Figure 1. Data Path

DETAILED DESCRIPTION

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, it interacts with the CPU and provides vectored interrupts and handshaking signals.

Data Communications Capabilities

The SCC provides two independent full-duplex channels programmable for use in any common asynchronous or SYNC data-communication protocol. Figure 2 and the following description briefly detail these protocols.

Asynchronous Modes

Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in the Z8530H Logic Symbol). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing of error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of framing error as a new start bit; a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals—a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In asynchronous modes, the $\overline{SYNC}$ pin may be programmed as an input used for functions, such as monitoring a ring indicator.

Synchronous Modes

The SCC supports both byte-oriented and bit-oriented synchronous communication. SYNC byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit SYNC character (Monosync), any 12-bit SYNC pattern (Bisync), or with an external SYNC signal. Leading SYNC characters can be removed without interrupting the CPU.

Five- or 7-bit SYNC characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming SYNC characters as shown in Figure 3.

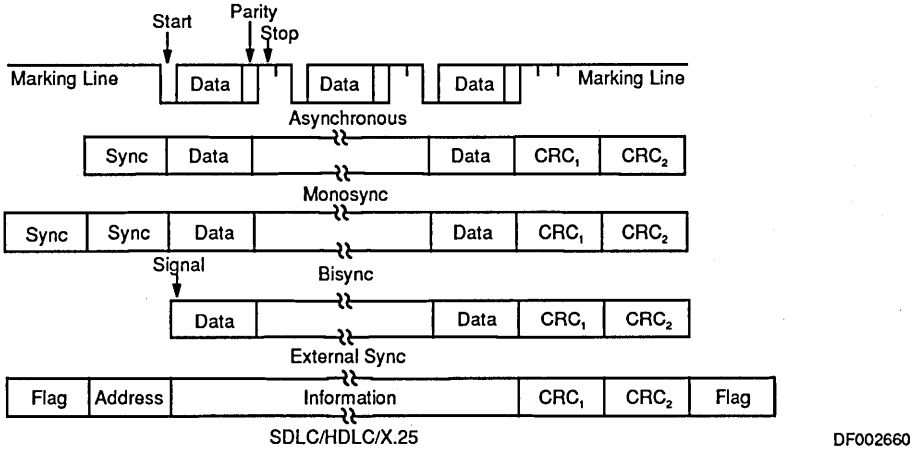


Figure 2. SCC Protocols

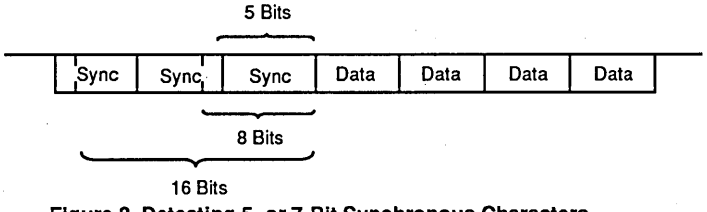


Figure 3. Detecting 5- or 7-Bit Synchronous Characters

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols, such as IBM BISYNC.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. Either polynomial may be selected in all synchronous modes. Users may preset the CRC generator and checker to all "1"s or all "0"s. The SCC also provides a feature that automatically transmits CRC data when no other data are available for transmission. This allows for high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there are no data or CRC to send in SYNC modes, the transmitter inserts 6-, 8-, or 16-bit SYNC characters, regardless of the programmed character length.

The SCC supports SYNC bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{SYNC}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all "0"s inserted by the transmitter during character assembly. CRC is also calculated and automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC mustbe programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all "1"s or all "0"s. The CRC is inverted before transmission and the receiver checks against the bit pattern "0001110100001111."

NRZ, NRZI or FM coding may be used in any 1X mode. The parity options available in asynchronous modes are available in synchronous modes.

The SCC can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC Loop Mode

The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 4).

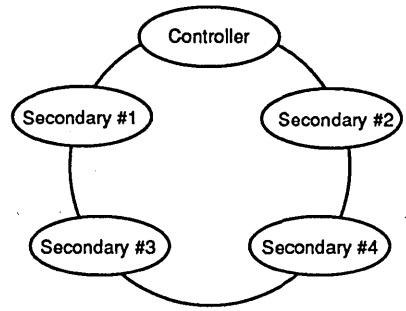


Figure 4. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, must pass these messages to the rest of the loop by retransmitting them with a one-bit time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern "11111110." Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary one of the EOP to a zero before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in the SDLC Loop mode.

Baud Rate Generator

Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On start-up, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching zero, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the digital phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate. (The baud rate is in bits/second and the BR clock period is in seconds.)

$$\text{baud rate} = \frac{1}{2\,(\text{Time Constant} + 2) \times (\text{BR Clock Period})}$$

Time-Constant Values for Standard Baud Rates at BR Clock = 3.9936 MHz

Rate (Baud)	Time Constant (decimal notation)	Error
19200	102	–
9600	206	–
7200	275	0.12%
4800	414	–
3600	553	0.06%
2400	830	–
2000	996	0.04%
1800	1107	0.03%
1200	1662	–
600	3326	–
300	6654	–
150	13310	–
134.5	14844	0.0007%
110	18151	0.0015%
75	26622	–
50	39934	–

Digital Phase-Locked Loop

The SCC contains a digital phase-locked loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1/0 or 0/1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next count cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15/16 counting transition.

The 32X clock for the DPLL can be programmed to come from either the $\overline{RTxC}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

Data Encoding

The SCC may be programmed to encode and decode the serial data in four different ways (Figure 5). In NRZ encoding, a "1" is represented by a High level, and a "0" is represented by a Low level. In NRZI encoding, a "1" is represented by no change in level, and a "0" is represented by a change in level. In FM_1 (more properly, biphase mark), a transition occurs at the beginning of every bit cell. A "1" is represented by an additional transition at the center of the bit cell, and a "0" is represented by no additional transition at the center of the bit cell. In FM_0 (biphase space), a transition occurs at the beginning of every bit cell. A "0" is represented by an additional transition at the center of the bit cell, and a "1" is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (biphase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0/1, the bit is a "0." If the transition is 1/0, the bit is a "1."

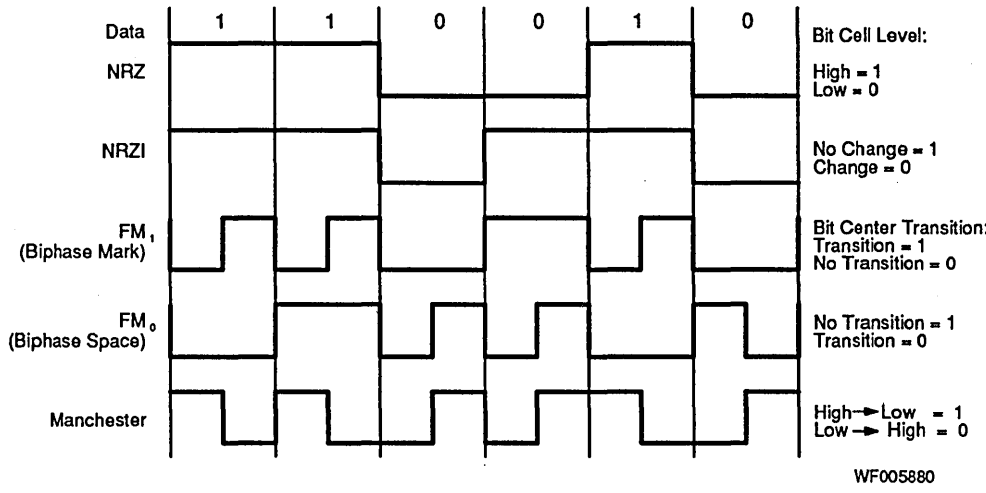


Figure 5. Data Encoding Methods

Auto Echo and Local Loopback

The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes but works in SYNC and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{CTS}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed, and the programmer is responsible for disabling transmitter interrupts and $\overline{Wait/Request}$ on transmit.

The SCC is also capable of local loopback. In this mode, TxD is RxD just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data, and RxD is ignored (except to be echoed out via TxD). The CTS and DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local loopback works in asynchronous, SYNC and SDLC modes with NRZ, NRZI or FM coding of the data stream.

I/O Interface Capabilities

The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

Polling

All interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts

When an SCC responds to an Interrupt Acknowledge signal ($\overline{INTACK}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 7 and 8).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write-only.

The other two bits are related to the Z-Bus interrupt priority chain (Figure 6). As a Z-Bus peripheral, the SCC may request an interrupt only when no higher priority device is requesting one, for example, when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the A/D bus.

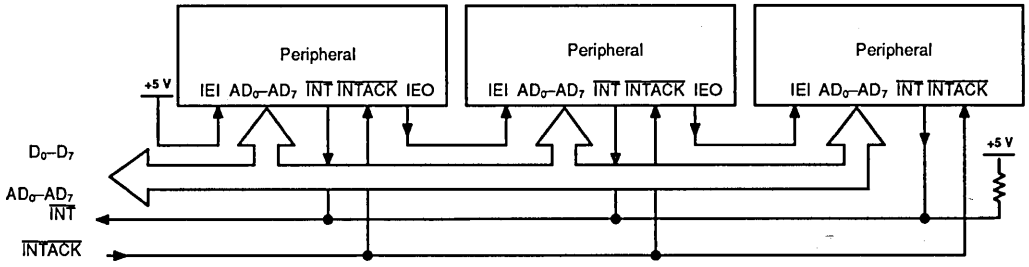
In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is set to "1" and the IEI input is High, the $\overline{INT}$ output is pulled Low, requesting an interrupt.

In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and external to the SCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the SCC being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receive, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the Receive can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive condition
- Interrupt on all Receive Characters or Special Receive condition
- Interrupt on Special Receive condition only



AF002770

Figure 6. Z-Bus Interrupt Schedule

Interrupt on First Character or Special Condition, and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in asynchronous mode, end-of-frame in SDLC mode, and optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive conditions any time after the first Receive Character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{SYNC}$ pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, a zero count in the baud rate generator, or the detection of a Break (asynchronous mode), Abort (SDLC mode), or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer

The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{Wait/Request}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{Wait/Request}$ output can be defined under software control as a $\overline{Wait}$ line in the CPU BlockTransfer mode or as a $\overline{Request}$ line in the DMA BlockTransfer mode.

To a DMA controller, the SCC $\overline{Request}$ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{Wait}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{DTR/Request}$ line, allows full-duplex operation under DMA control.

PROGRAMMING INFORMATION

Each channel has fifteen Write registers that are individually programmed from the system bus to configure the functional personality of each channel. Each channel also has eight Read registers from which the system can read Status, Baud rate, or Interrupt information.

In the Z8530H, only the four data registers (Read and Write for Channels A and B) are directly selected by a High on the D/$\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and A/$\overline{B}$ pins. All other registers are addressed indirectly by the content of Write Register 0 in conjunction with a Low on the D/$\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and A/$\overline{B}$ pins. If bit D3 in WR0 is "1" and bits 5 and 6 are "0," then bits 0, 1, 2 address the higher registers 8 through 15. If bits 4, 5, 6 contain a different code, bits 0, 1, 2 address the lower registers 0 through 7 as shown in Table 2.

Writing to or reading from any register except RR0, WR0 and the Data Registers thus involves two operations: First, write the appropriate code into WR0, then follow this by a write or read operation on the register thus specified. Bits 0 through 4 in WR0 are automatically cleared after this operation, so that WR0 then points to WR0 or RR0 again.

Channel A/Channel B selection is made by the A/$\overline{B}$ input (High = A, Low = B).

In the Z8530H, the system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, or even or odd parity might be set first. Then the interrupt mode would be set and, finally, receiver or transmitter enable.

Table 2. Register Addressing

D/$\overline{C}$	"Point High" Code in WR0	D2, D1, D0 in WR0			Write Register	Read Register
High	Either Way	X	X	X	Data	Data
Low	Not True	0	0	0	0	0
Low	Not True	0	0	1	1	1
Low	Not True	0	1	0	2	2
Low	Not True	0	1	1	3	3
Low	Not True	1	0	0	4	(0)
Low	Not True	1	0	1	5	(1)
Low	Not True	1	1	0	6	(2)
Low	Not True	1	1	1	7	(3)
Low	True	0	0	0	Data	Data
Low	True	0	0	1	9	–
Low	True	0	1	0	10	10
Low	True	0	1	1	11	(15)
Low	True	1	0	0	12	12
Low	True	1	0	1	13	13
Low	True	1	1	0	14	(10)
Low	True	1	1	1	15	15

Read Registers

The SCC contains eight read registers (actually nine, counting the receive buffer [RR8] in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 7 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring, for example, when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

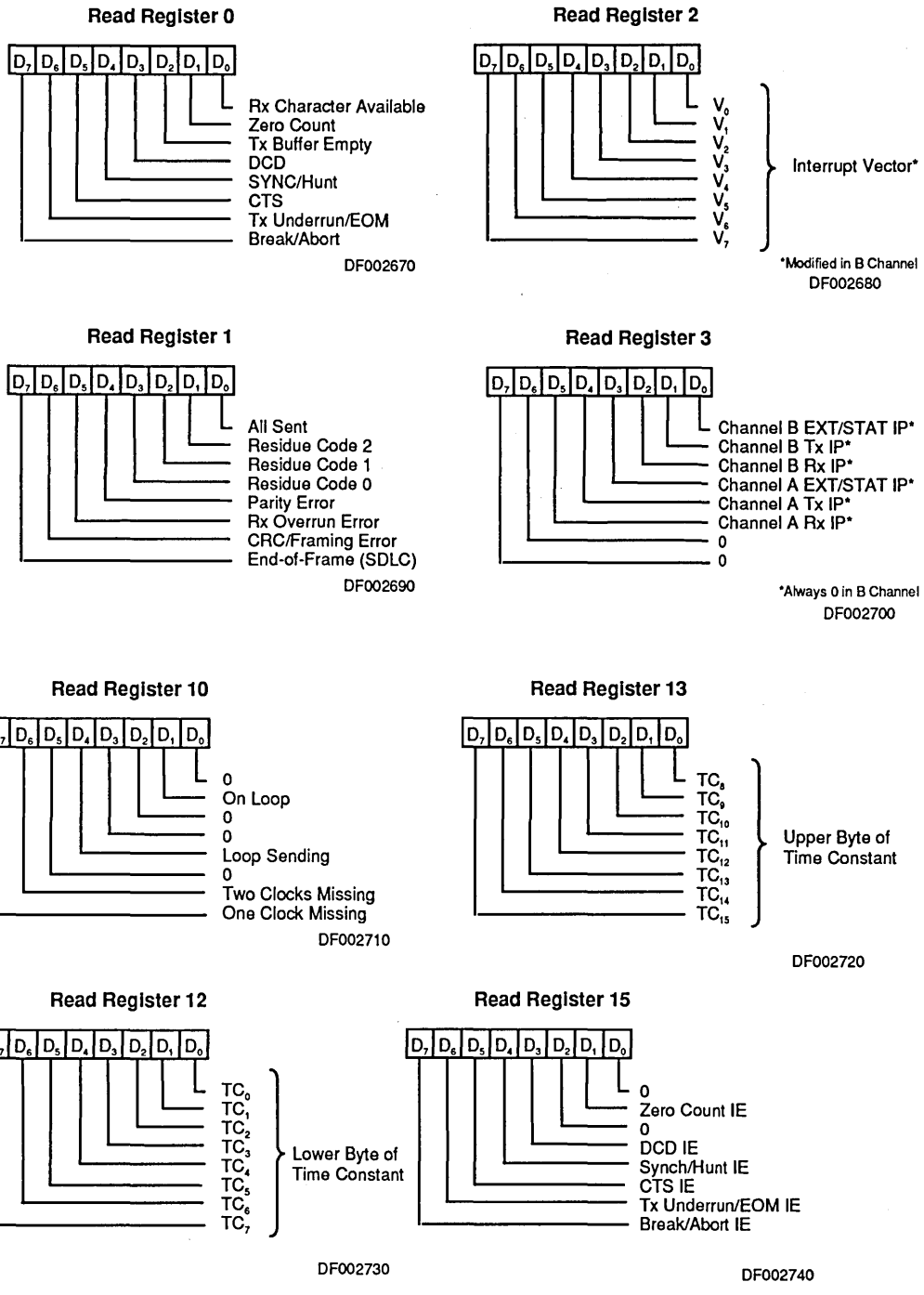


Figure 7. Read Register Bit Functions

Write Registers

The SCC contains 15 write registers (16 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. Two registers (WR2 and WR9) are shared by the two channels and may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 8 shows the format of each write register.

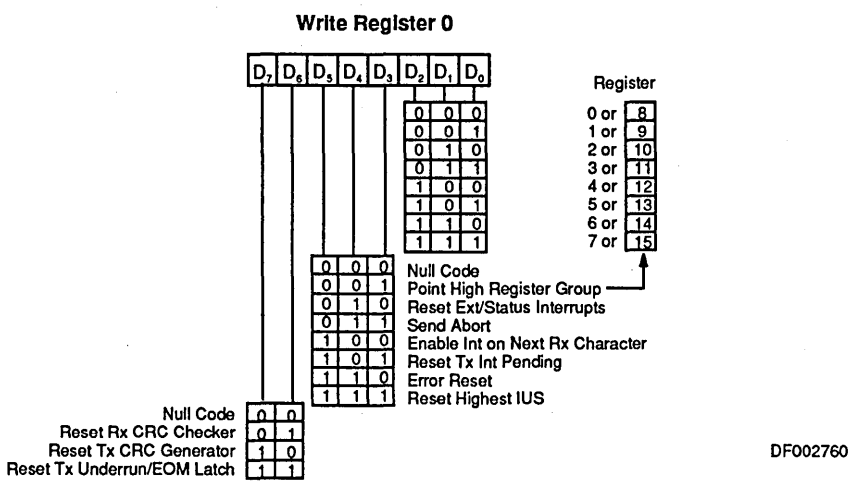


Figure 8. Write Register Bit Functions

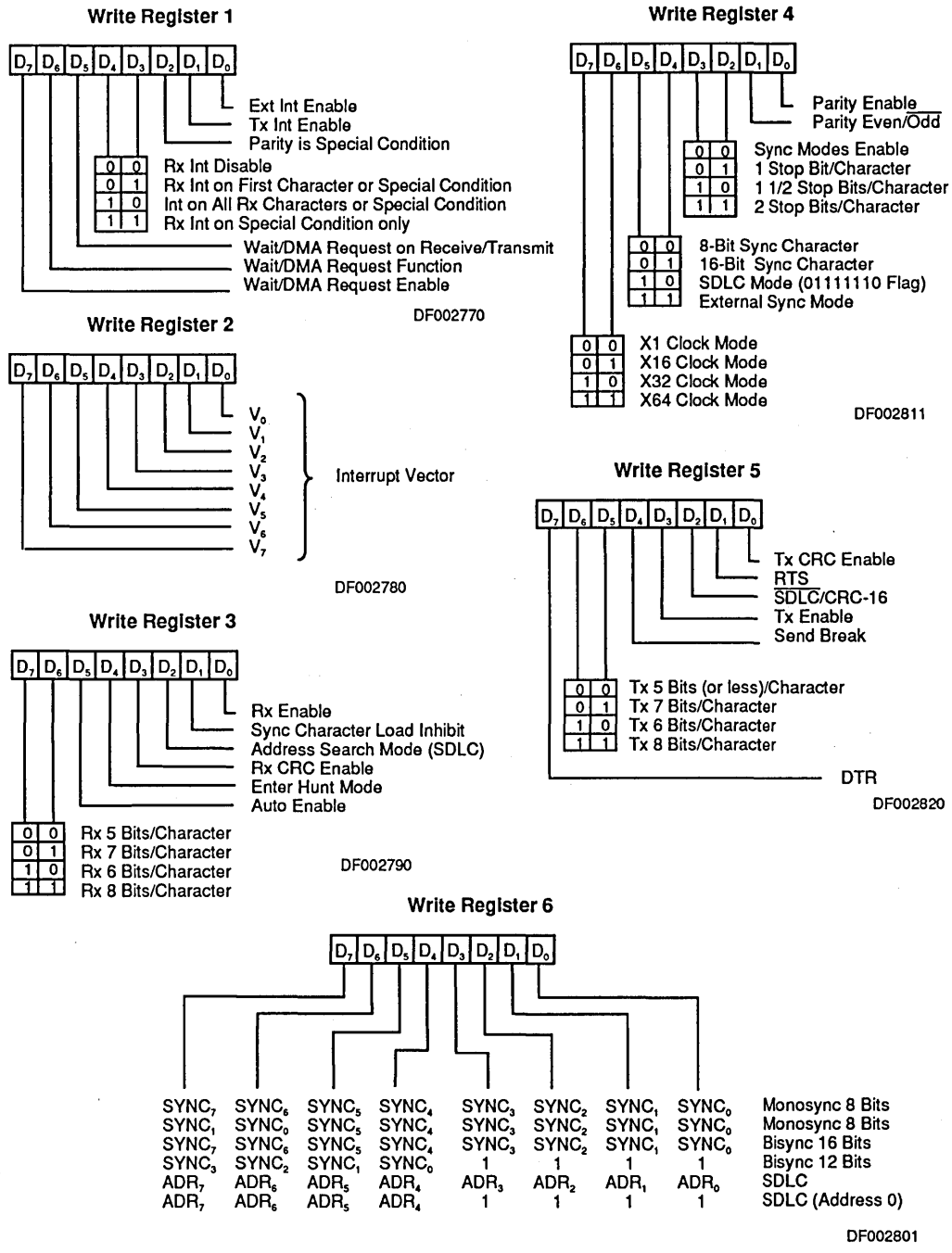


Figure 8. Write Register Bit Functions (continued)

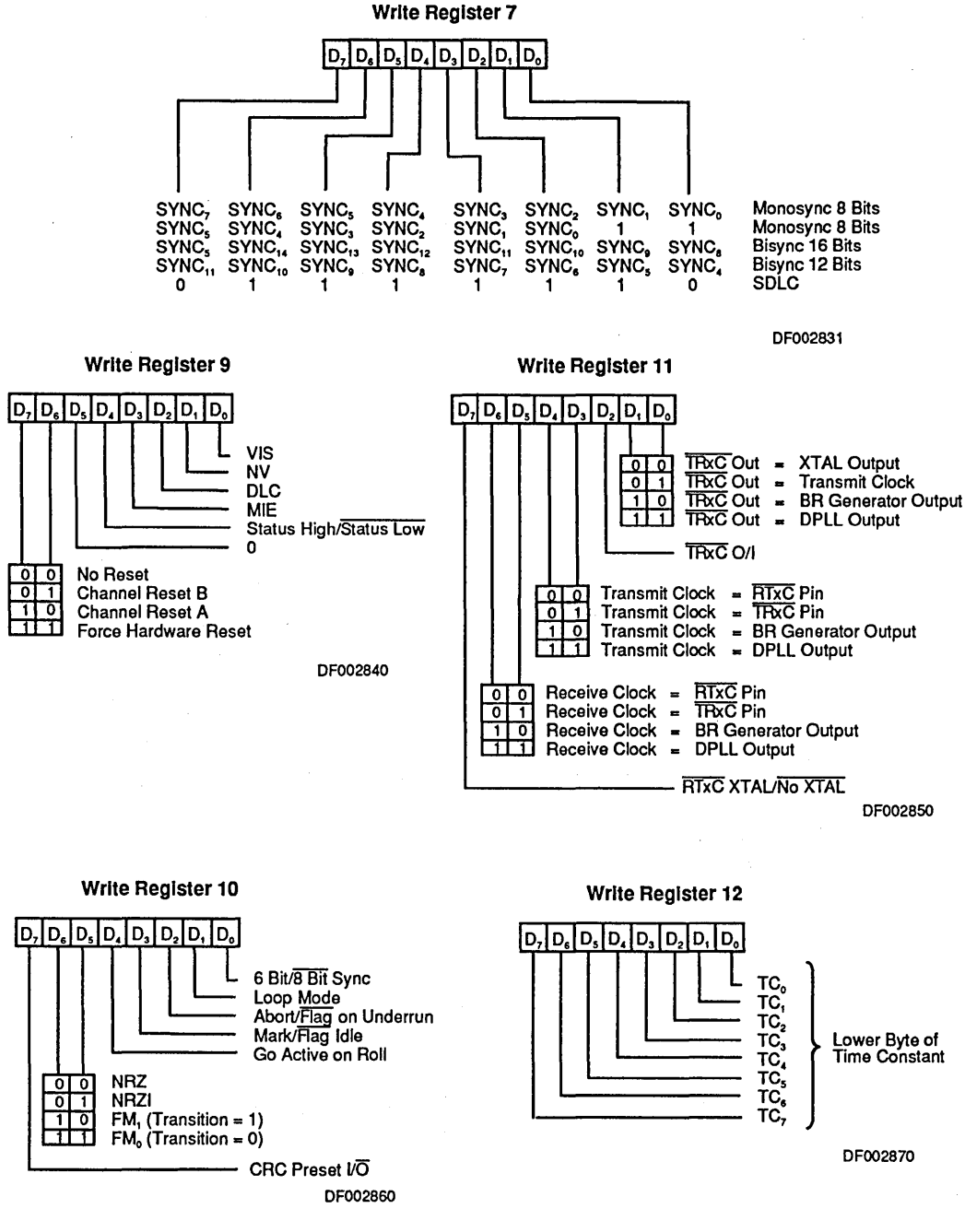


Figure 8. Write Register Bit Functions (continued)

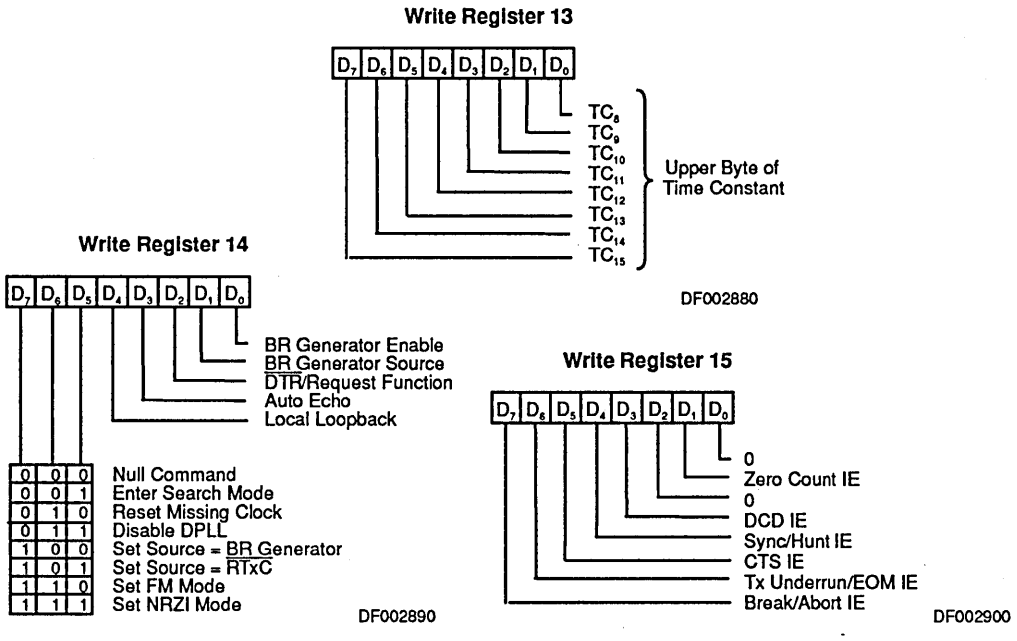


Figure 8. Write Register Bit Functions (continued)

2

Z8530H Timing

The SCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the rising edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the SCC to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the SCC. This time must be at least six PCLK cycles plus 200nsec.

Read Cycle Timing

Figure 9 illustrates Read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

Write Cycle Timing

Figure 10 illustrates Write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened.

Interrupt Acknowledge Cycle Timing

Figure 11 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes Low and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is High when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on D_0–D_7, and it then sets the appropriate Interrupt-Under-Service internally.

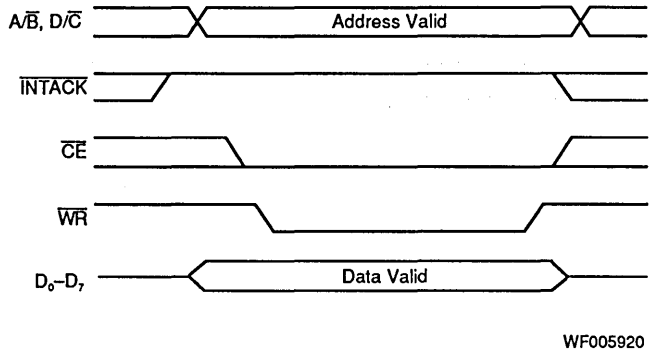


Figure 9. Read Cycle Timing

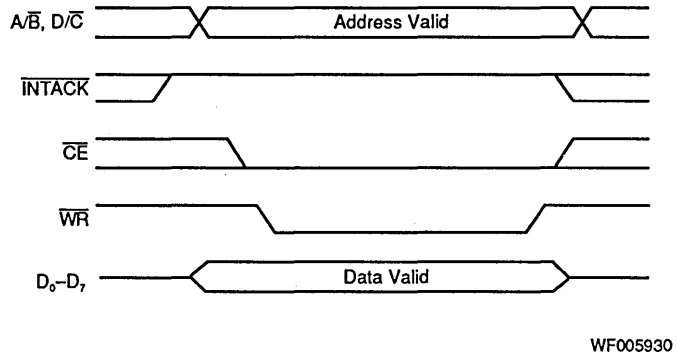


Figure 10. Write Cycle Timing

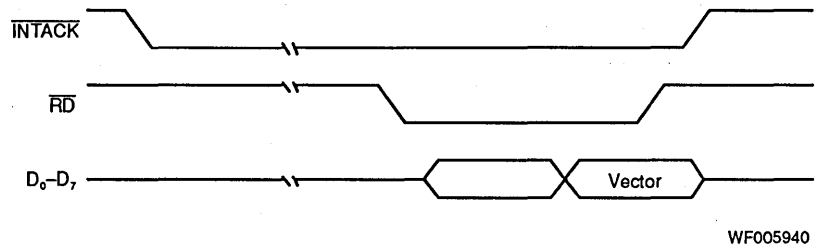


Figure 11. Interrupt Acknowledge Cycle Timing

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	−65 to +150°C
Voltage at any Pin Relative to V_{SS}	−0.5 to +7.0 V
Power Dissipation	1.8 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A)	0 to +70°C
Supply Voltage (V_{CC})	5 V ± 5%

Military (M) Devices

Temperature (T_C)	−55 to 125°C
Supply Voltage (V_{CC})	5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IH}	Input High Voltage	Commercial	2.0	V_{CC} +0.3	V
V_{IH}	Input High Voltage	Military	2.2	V_{CC} +0.3	V
V_{IL}	Input Low Voltage		−0.3	0.8	V
V_{OH}	Output High Voltage	I_{OH} = −250 µA	2.4		V
V_{OL}	Output Low Voltage	I_{OL} = +2.0 mA		0.4	V
I_{IL}	Input Leakage	0.4 V ≤ V_{IN} ≤ 2.4 V		±10.0	µA
I_{OL}	Output Leakage	0.4 V ≤ V_{OUT} ≤ 2.4 V		±10.0	µA
I_{CC}	V_{CC} Supply Current			250	mA
C_{IN}	Input Capacitance	Unmeasured pins returned		10	pF
C_{OUT}	Output Capacitance	to ground. f = 1 MHz at		15	pF
C_{IO}	Bidirectional Capacitance	T_A = 25°C.		20	pF

2

Standard Test Conditions

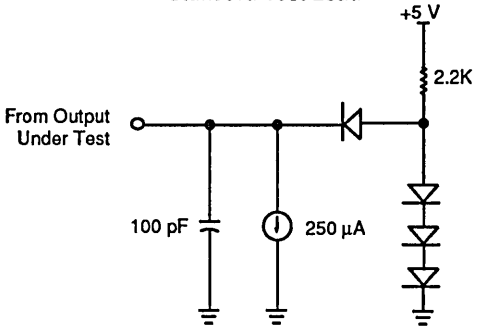
The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

Commercial (Z8530H)
+4.75 V ≤ V_{CC} ≤ +5.25 V
GND = 0 V
0°C ≤ T_A ≤ 70°C

Military (Z8530)
+4.5 V ≤ V_{CC} ≤ +5.5 V
GND = 0 V
−55°C ≤ T_C ≤ 125°C

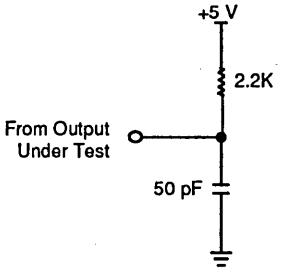
SWITCHING TEST CIRCUITS

Standard Test Load



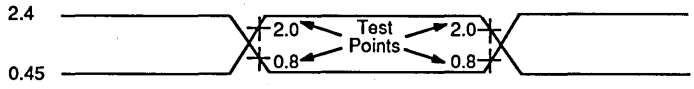
TC001821

Open-Drain Test Load



TC001830

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WR006352

AC testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0."
Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for logic "0."

SWITCHING CHARACTERISTICS over COMMERCIAL operating range

General Timing

No.	Parameter Symbol	Parameter Description	4 MHz Min.	4 MHz Max.	6 MHz Min.	6 MHz Max.	8.192 MHz Min.	8.192 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to W/$\overline{REQ}$ Valid Delay		250		250		250	nsec
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		350		350	nsec
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4)	80	T_{WPCL}	70	T_{WPCL}	60	T_{WPCL}	nsec
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1)	0		0		0		nsec
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1)	150		150		150		nsec
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		0		nsec
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5)	150		150		150		nsec
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1)	−200		−200		−200		nsec
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1)	3TcPC +400		3TcPC +320		3TcPC +250		nsec nsec
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4)	0		0		0		nsec
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2)		300		230		200	nsec
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5)		300		230		200	nsec
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		200		200		200	nsec
14	TwRTXh	$\overline{RTxC}$ High Width (Note 6)	180		180		150		nsec
15	TwRTXl	$\overline{RTxC}$ Low Width (Note 6)	180		180		150		nsec
16	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	1000		660		488		nsec
17	TcRTXX	Crystal Oscillator Period (Note 3)	250	1000	165	1000	122	1000	nsec
18	TwTRXh	$\overline{TRxC}$ High Width (Note 6)	180		180		150		nsec
19	TwTRXl	$\overline{TRxC}$ Low Width (Note 6)	180		180		150		nsec
20	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	1000		660		488		nsec
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	200		200		200		nsec
22	TwSY	$\overline{SYNC}$ Pulse Width	200		200		200		nsec

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ have 18 pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.

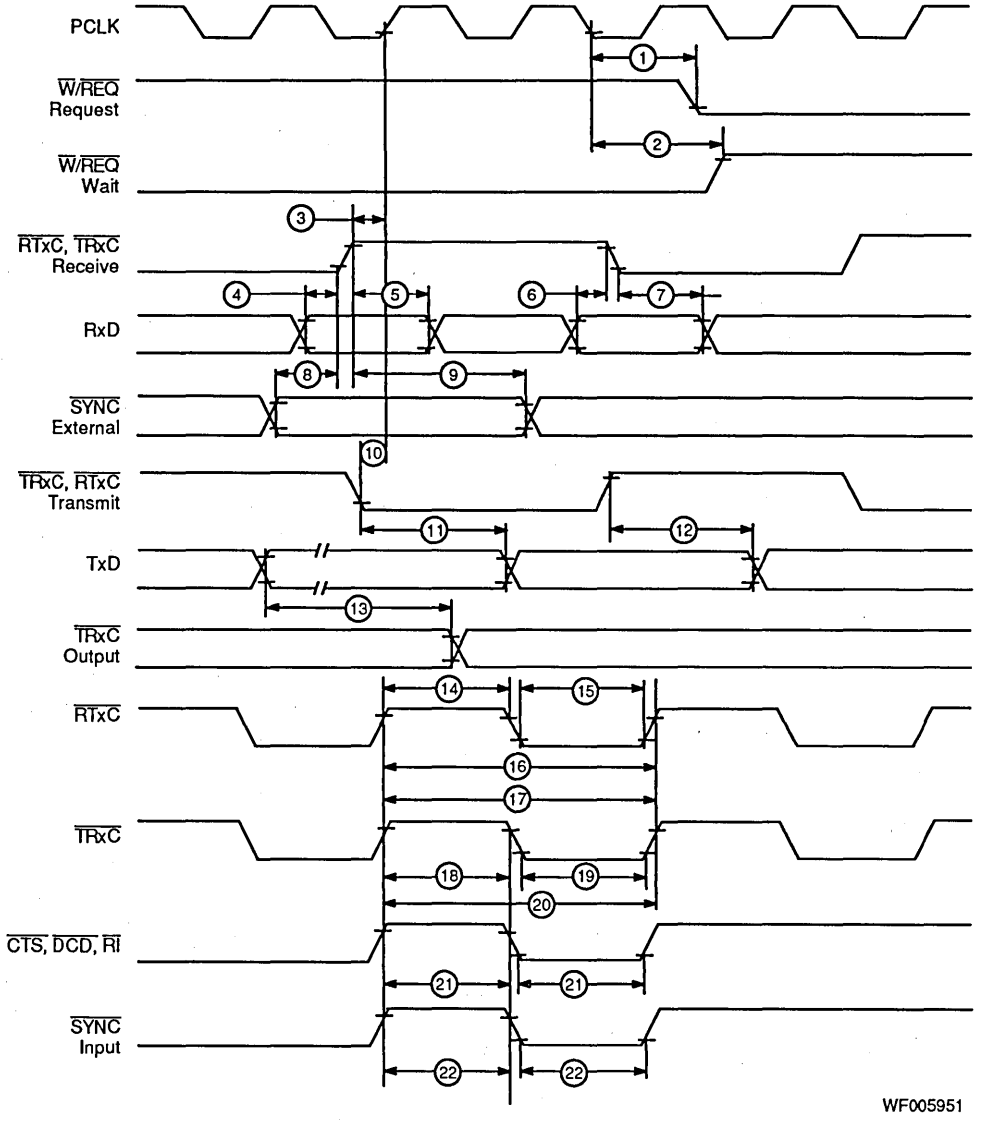


Figure 12. General Timing

SWITCHING CHARACTERISTICS over COMMERCIAL operating range

System Timing

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8.192 MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	8	14	TcPC
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	4	7	TcPC
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	10	16	TcPC
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	5	8	TcPC
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	5	11	TcPC
7	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	4	7	TcPC
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	6	10	TcPC
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	2	6	TcPC
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	2	6	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

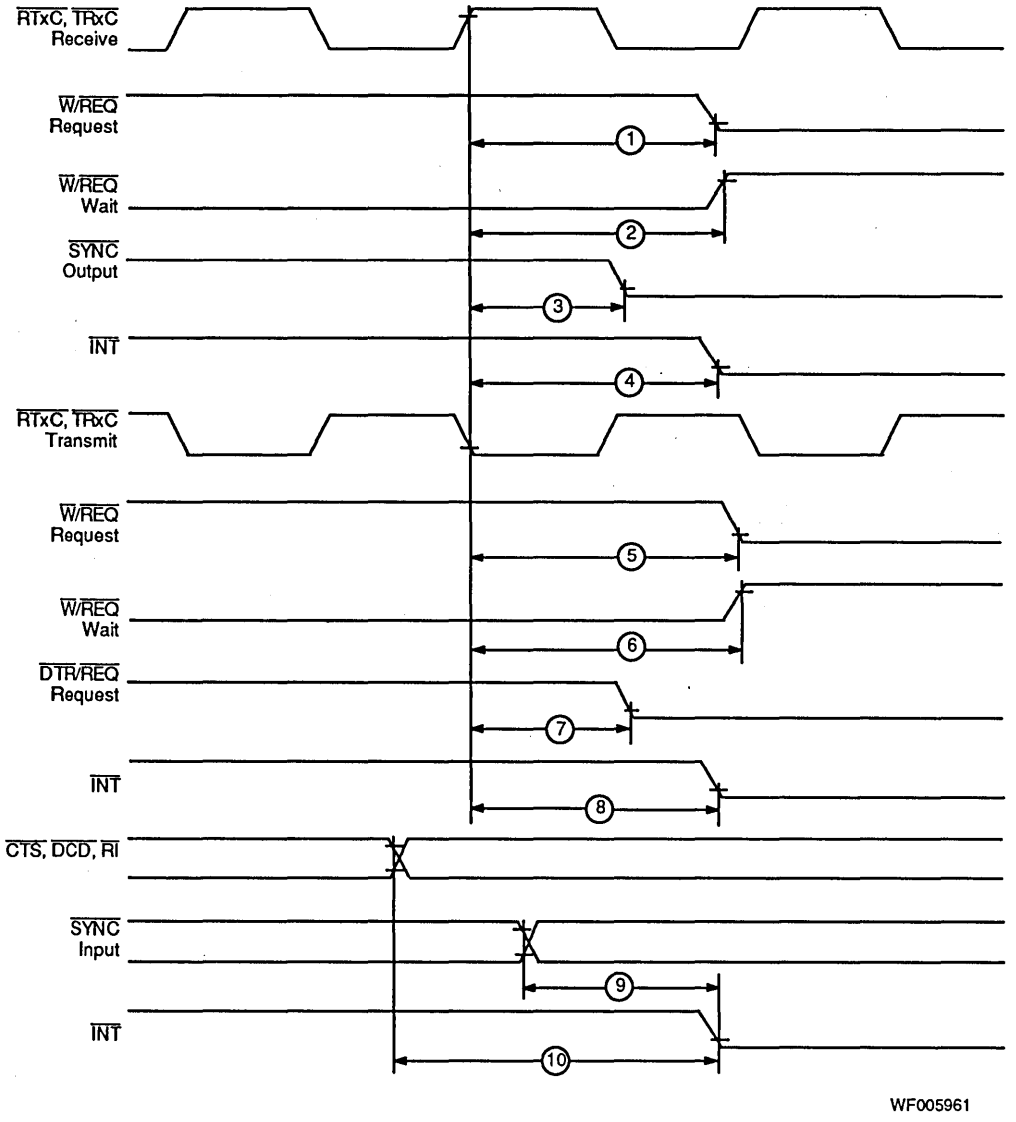


Figure 13. System Timing

SWITCHING CHARACTERISTICS over COMMERCIAL operating range

Read and Write Timing

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8.192 MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	TwPCl	PCLK Low Width	105	2000	70	1000	50	1000	nsec
2	TwPCh	PCLK High Width	105	2000	70	1000	50	1000	nsec
3	TfPC	PCLK Fall Time		20		15		15	nsec
4	TrPC	PCLK Rise Time		20		15		15	nsec
5	TcPC	PCLK Cycle Time	250	4000	165	2000	122	2000	nsec
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		80		70		nsec
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		0		nsec
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		80		70		nsec
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		0		nsec
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	10		10		10		nsec
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	200		160		145		nsec
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		nsec
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note1)	200		160		145		nsec
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		0		nsec
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		100		85		nsec
16	TsCEi(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		0		nsec
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		nsec
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		70		60		nsec
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		0		nsec
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		0		0		nsec
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time (Note1)	100		70		60		nsec
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	240		200		150		nsec
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		0		nsec
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		0		nsec
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		220		180		140	nsec
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		70		45		40	nsec

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a +0.5 V change in the output with a maximum DC load and minimum AC load.

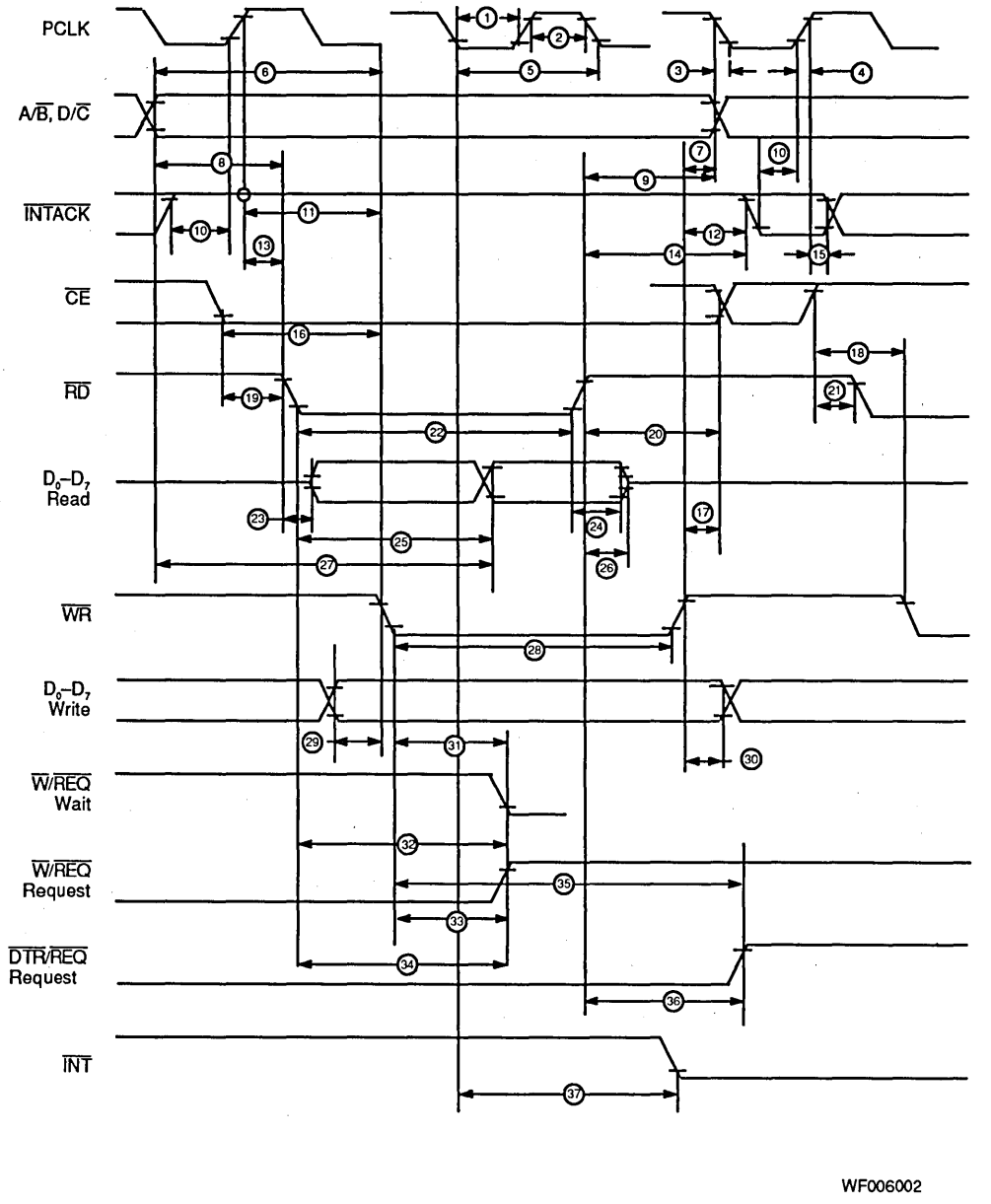


Figure 14. Read and Write Timing

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing

No.	Parameter Symbol	Parameter Description	4 MHz Min.	4 MHz Max.	6 MHz Min.	6 MHz Max.	8.192 MHz Min.	8.192 MHz Max.	Unit
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		300		280		220	nsec
28	TwWRl	$\overline{WR}$ Low Width	240		200		150		nsec
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	10		10		10		nsec
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		0	170	nsec
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 4)		240		200		170	nsec
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 4)		240		200		170	nsec
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200		170	nsec
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200		170	nsec
35	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	nsec
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	nsec
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 4)		500		500		500	nsec
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 5)	250		200		150		nsec
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	250		200		150		nsec
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180		140	nsec
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	120		100		95		nsec
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		0		nsec
43	TdIEI(IEO)	IEI to IEO Delay Time		120		100		95	nsec
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250		250		200	nsec
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 4)		500		500		450	nsec
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	30		15		15		nsec
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	30		30		20		nsec
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	250		200		150		nsec
49	Trc	Valid Access Recovery Time (Note 3)	4TcPC		4TcPC		4TcPC		nsec

Notes: 3. Parameter applies only between transactions involving the SCC.

4. Open-drain output, measured with open-drain test load.

5. Parameter is system dependent. For any SCC in the daisy chain, TdIA(RD) must be greater then the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEIf(IEO) for each deivce separating them in the daisy chain.

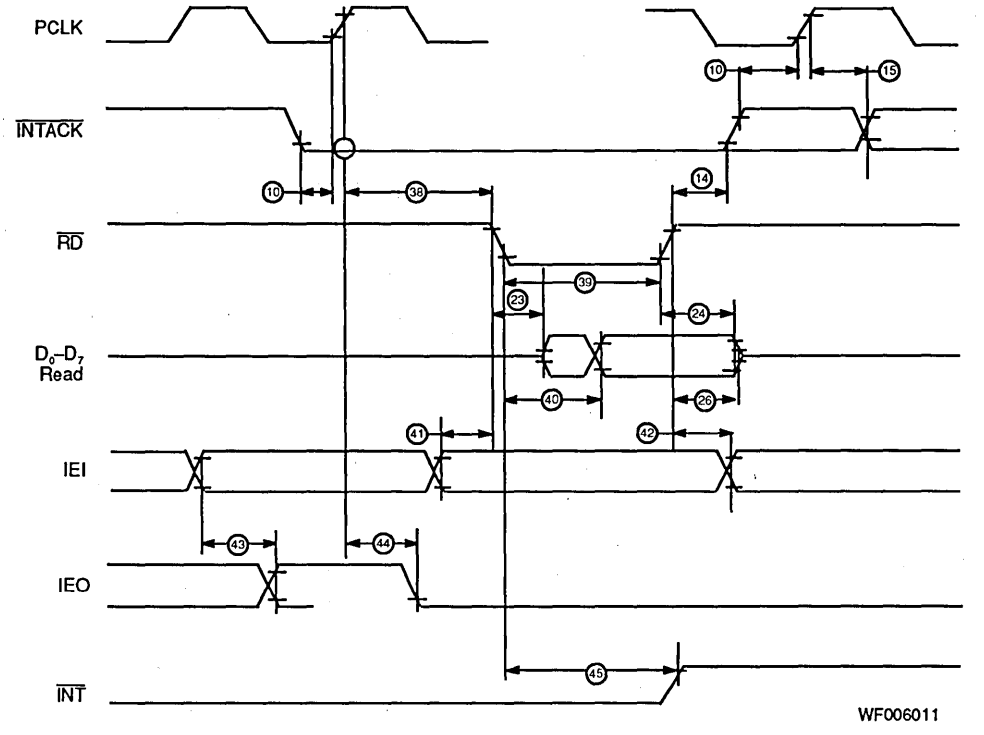


Figure 15. Interrupt Acknowledge Timing

WF006011

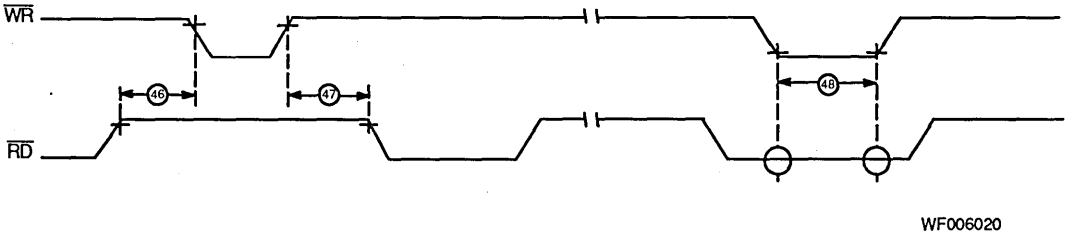


Figure 16. Reset Timing

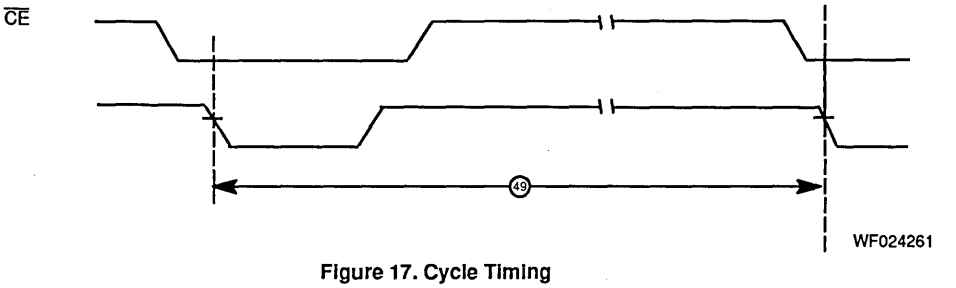


Figure 17. Cycle Timing

*Timings are preliminary and subject to change.

SWITCHING CHARACTERISTICS over MILITARY operating range (for APL Products, Group A, Subgroups 9, 10, and 11 are tested unless otherwise noted)

General Timing

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
1	TdPC(REQ)	PCLK ↓ to W/$\overline{REQ}$ Valid Delay		250	nsec
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350	nsec
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4)	80	T_{WPCL}	nsec
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1)	0		nsec
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1)	150		nsec
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5)	0		nsec
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5)	150		nsec
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1)	–200		nsec
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1)	3TcPC +400		nsec
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4)	0		nsec
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2)		300	nsec
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5)		300	nsec
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		200	nsec
14	TwRTXh	$\overline{RTxC}$ High Width (Note 6)	180		nsec
15	TwRTXl	$\overline{RTxC}$ Low Width (Note 6)	180		nsec
16	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	1000		nsec
17	TcRTXX	Crystal Oscillator Period (Note 8)	250	1000	nsec
18	TwTRXh	$\overline{TRxC}$ High Width (Note 6)	180		nsec
19	TwTRXl	$\overline{TRxC}$ Low Width (Note 6)	180		nsec
20	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	1000		nsec
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	200		nsec
22	TwSY	$\overline{SYNC}$ Pulse Width	200		nsec

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. Not tested; guaranteed by design.

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)

System Timing

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	TcPC
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	TcPC
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	TcPC
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	TcPC
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	TcPC
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	TcPC
7	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	TcPC
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	TcPC
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	TcPC
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)

Read and Write Timing

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
1	TwPCl	PCLK Low Width	105	1000	
2	TwPCh	PCLK High Width	105	1000	nsec
3	TfPC	PCLK Fall Time		20	nsec
4	TrPC	PCLK Rise Time		20	nsec
5	TcPC	PCLK Cycle Time	250	2000	nsec
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		nsec
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		nsec
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		nsec
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		nsec
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	10		nsec
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	200		nsec
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		nsec
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	200		nsec
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		nsec
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		nsec
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		nsec
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		nsec
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		nsec
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time (Note 1)	0		nsec
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		nsec
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time (Note 1)	100		nsec
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	240		nsec
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		nsec
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		nsec
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		220	nsec
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		70	nsec

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for the data bus to be released with a maximum DC load and minimum AC load.

SWITCHING CHARACTERISTICS over MILITARY operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		400	nsec
28	TwWRl	$\overline{WR}$ Low Width	240		nsec
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	10		nsec
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		nsec
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 4)		240	nsec
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 4)		240	nsec
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240	nsec
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240	nsec
35	TdWRr(REQ)	$\overline{WR}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300	nsec
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300	nsec
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 4)		500	nsec
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 5)	250		nsec
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	250		
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		250	nsec
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	120		nsec
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		nsec
43	TdIEI(IEO)	IEI to IEO Delay Time		120	nsec
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250	nsec
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 4)		500	nsec
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	30		nsec
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	30		nsec
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	250		nsec
49	Trc	Valid Access Recovery Time (Note 3)	6TcPC +200		nsec

Notes: 3. Parameter applies only between transactions involving the SCC.

4. Open-drain output, measured with open-drain test load.

5. Parameter is system dependent. For any SCC in the daisy chain, TdIAI(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEf(IEO) for each device separating them in the daisy chain.

Final

Z85C30

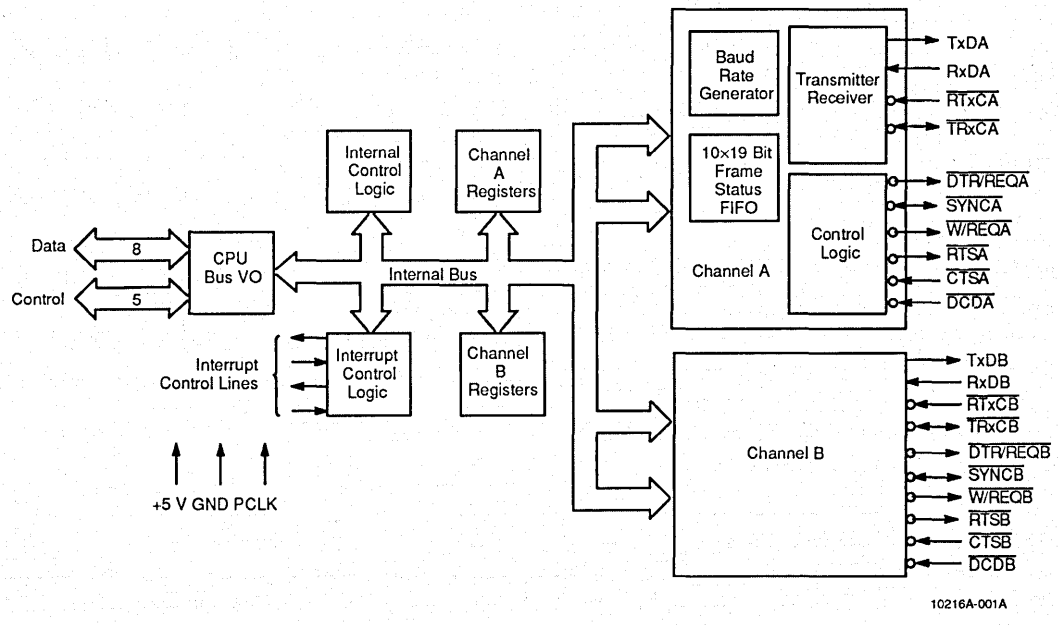
Enhanced Serial Communications Controller

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Fastest data rate of any Z8530**
  - 8.192 MHz / 2.048 Mb/s
  - 10 MHz / 2.5 Mb/s
  - 12.5 MHz / 3 Mb/s
  - 16.384 MHz / 4.096 Mb/s
  - 20 MHz / 5 Mb/s (prelim)
- **Low-power CMOS technology**
- **Pin and function compatible with other NMOS and CMOS Z8530s**
- **Easily interfaced with most CPUs**
  Compatible with non-multiplexed bus
- **Many enhancements over NMOS Z8530H**
  - Allows 85C30 to be used more effectively in high-speed applications
  - Improves interface capabilities
- **Two independent full-duplex serial channels**
- **Asynchronous mode features**
  - Programmable stop bits, clock factor, character length and parity
  - Break detection/generation
  - Error detection for framing, overrun, and parity
- **Synchronous mode features**
  - Supports IBM BISYNC, SDLC, SDLC Loop, HDLC, and ADCCP Protocols
  - Programmable CRC generators and checkers
  - SDLC/HDLC support includes frame control, zero insertion and deletion, abort, and residue handling

BLOCK DIAGRAM



10216A-001A

BD008260

DISTINCTIVE CHARACTERISTICS (continued)

- **Enhanced SCC functions support high-speed frame reception using DMA**
  - — 14-bit byte counter
  - — 10 × 19 SDLC/HDLC Frame Status FIFO
  - — Independent Control on both channels
  - — Enhanced operation does not allow special receive conditions to lock the 3-byte DATA FIFO when the 10 × 19 FIFO is enabled
- **Local Loopback and Auto Echo modes**
- **Internal or external character synchronization**
- **2-Mb/s FM encoding transmit and receive capability using internal DPLL for 16.384-MHz product**
- **Internal synchronization between RxC to PCLK and TxC to PCLK**
  - —This allows the user to eliminate external synchronization hardware required by the NMOS device when transmitting or receiving data at the maximum rate of 1/4 PCLK frequency.

GENERAL DESCRIPTION

AMD's Z85C30 is an enhanced pin-compatible version of the popular Z8530/Z85C30 Serial Communications Controller. The Enhanced Serial Communications Controller (ESCC) is a high-speed, low-power, multi-protocol communications peripheral designed for use with 8- and 16-bit microprocessors. It has two independent, full-duplex channels and functions as a serial-to-parallel, parallel-to-serial converter/controller. AMD's proprietary enhancements make the Z85C30 easier to interface and more effective in high-speed applications due to a reduction in software burden and the elimination of the need for some external glue logic.

The Z85C30 is easy to use due to a variety of sophisticated internal functions, including on-chip baud rate generators, digital phase-locked loops, and crystal oscillators, which dramatically reduce the need for external logic. The device can generate and check CRC codes in any SYNC mode, and can be programmed to check data integrity in various modes. The ESCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

This versatile device supports virtually any serial data transfer application such as networks, modems, cassettes, and tape drivers. The ESCC is designed for non-multiplexed buses and is easily interfaced with most CPUs, such as 80188, 80186, 80286, 8080, Z80, 6800, 68000 and MULTIBUS.

Enhancements that allow the Z85C30 to be used more effectively in high-speed applications include:

- a 10 × 19 bit SDLC/HDLC frame status FIFO array
- a 14-bit SDLC/HDLC frame byte counter
- automatic SDLC/HDLC opening frame flag transmission
- TxD pin forced High in SDLC NRZI mode after closing flag
- automatic SDLC/HDLC Tx underrun/EOM flag reset
- automatic SDLC/HDLC Tx CRC generator reset/preset
- $\overline{RTS}$ synchronization to closing SDLC/HDLC flag
- $\overline{DTR/REQ}$ deactivation delay significantly reduced
- external PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation

Other enhancements to improve the Z85C30 interface capabilities include:

- write data valid setup time to falling edge of $\overline{WR}$ requirement eliminated
- reduced $\overline{INT}$ response time
- reduced access recovery time (t_{RC}) to 3 PCLK best case (3 1/2 PCLK worst case)
- improved $\overline{Wait}$ timing
- Write Registers WR3, WR4, WR5, and WR10 made readable
- lower priority interrupt masking without $\overline{INTACK}$
- complete SDLC/HDLC CRC character reception

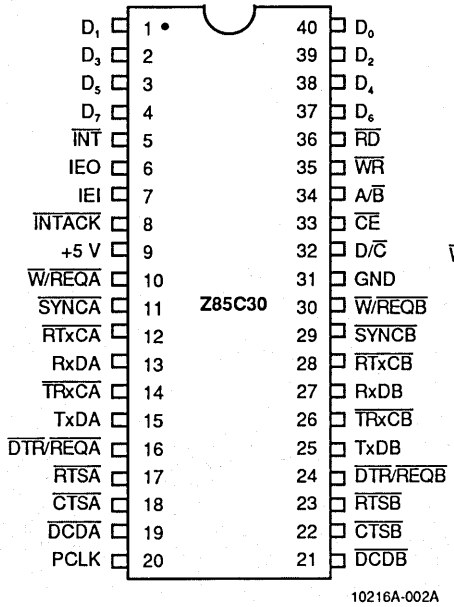
RELATED AMD PRODUCTS

Part No.	Description	Part No.	Description
Am7960	Coded Data Transceiver	Am9517A	DMA Controller
80186	Highly Integrated 16-Bit Microprocessor	5380, 53C80	SCSI Bus Controller
		80188	Highly Integrated 8-Bit Microprocessor
80286, 80C286	High-Performance 16-Bit Microprocessor		

CONNECTION DIAGRAMS

Top View

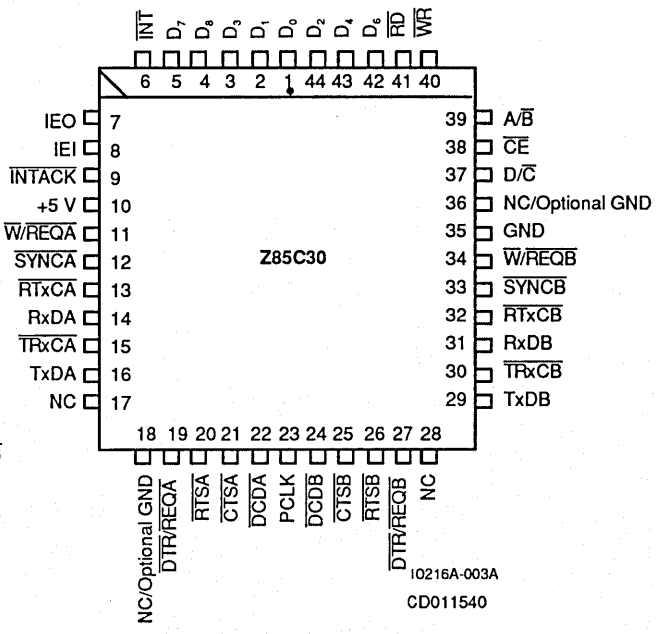
DIPs



10216A-002A

CD011530

PLCC, LCC

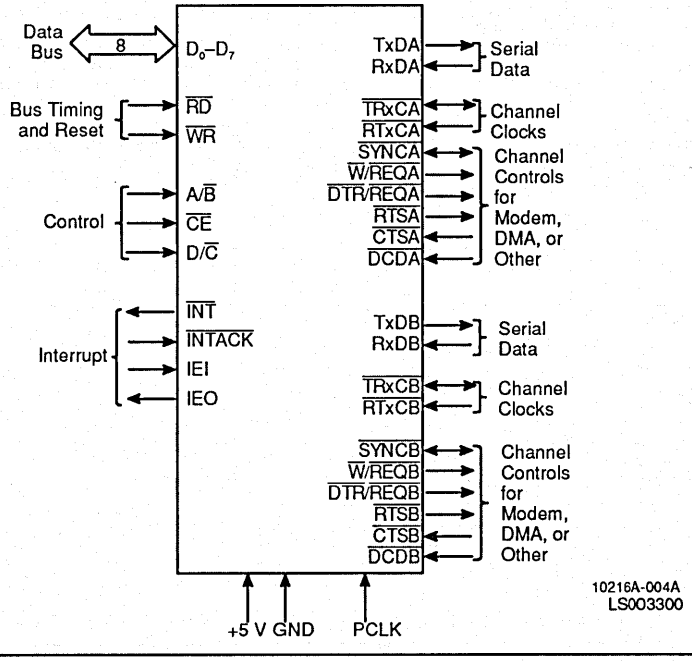


10216A-003A

CD011540

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



10216A-004A

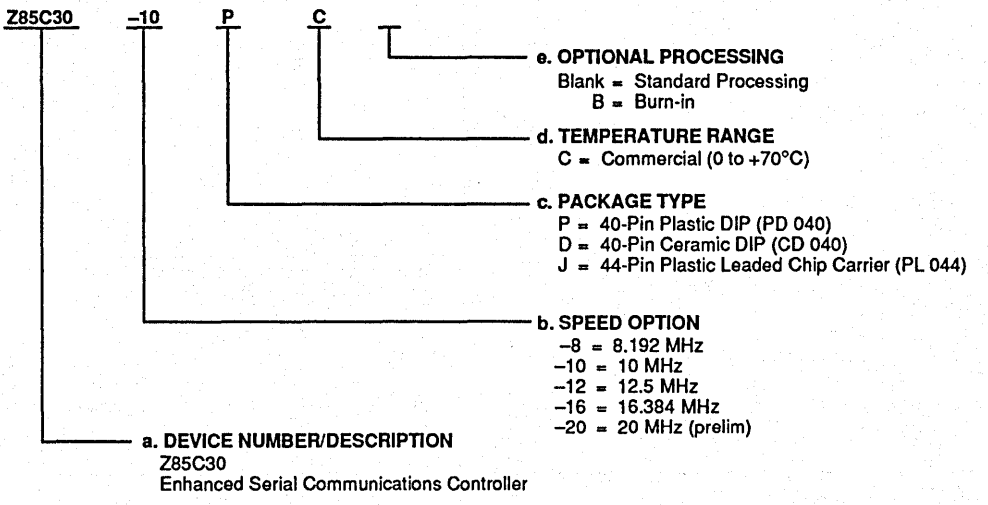
LS003300

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Z85C30-8 Z85C30-10 Z85C30-12 Z85C30-16 Z85C30-20 (prelim)	PC, DC, JC DCB

Valid Combinations

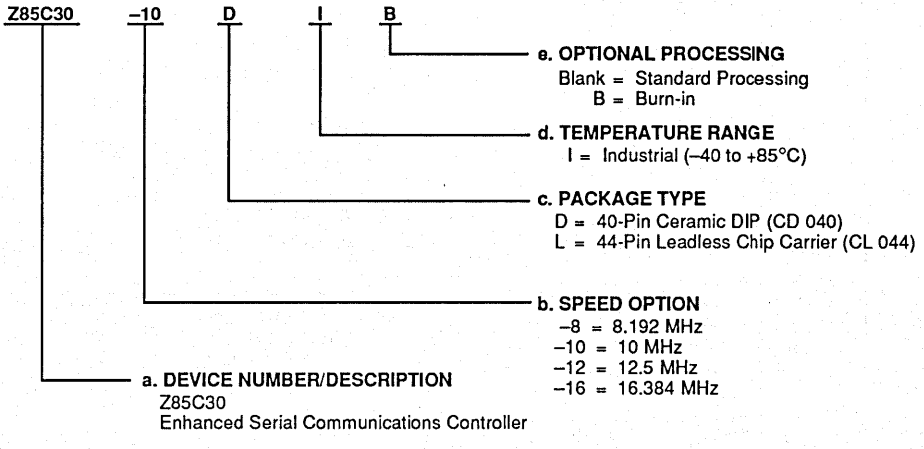
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION

Industrial Products

AMD industrial products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Z85C30-8 Z85C30-10 Z85C30-12 Z85C30-16	DIB, LIB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

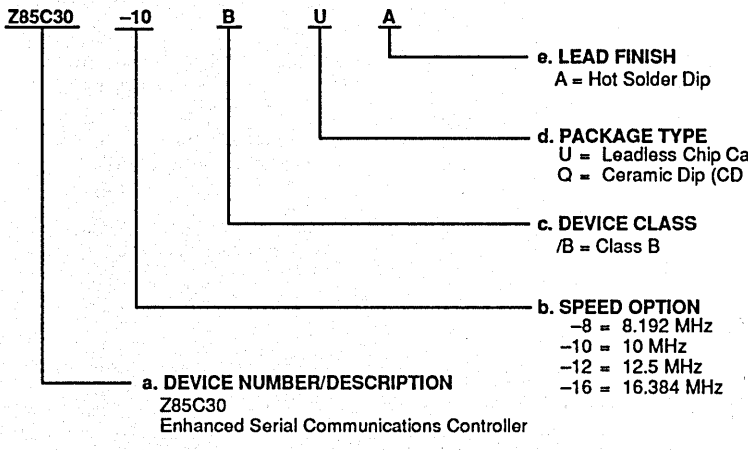
2

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
Z85C30-8 Z85C30-10 Z85C30-12 Z85C30-16	BQA, BUA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Bus Timing and Reset

$\overline{RD}$
Read (Input; Active Low)

This signal indicates a Read operation and, when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

$\overline{WR}$
Write (Input; Active Low)

When the SCC is selected, this signal indicates a Write operation. The coincidence of $\overline{RD}$ and $\overline{WR}$ is interpreted as a reset.

Channel Clocks

$\overline{RTxCA}$, $\overline{RTxCB}$
Receive/Transmit Clocks (Inputs; Active Low)

These pins can be programmed in several different modes of operation. In each channel, $\overline{RTxC}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock of the digital phase-locked loop. These pins can also be programmed for use with the respective $\overline{SYNC}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

$\overline{TRxCA}$, $\overline{TRxCB}$
Transmit/Receive Clocks
(Inputs/Outputs; Active Low)

These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

Channel Controls for Modem, DMA, or Other

$\overline{CTSA}$, $\overline{CTSB}$
Clear to Send (Inputs; Active Low)

If these pins are programmed as Auto Enables, a Low on these inputs enables their respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and may interrupt the CPU on both logic level transitions.

$\overline{DCDA}$, $\overline{DCDB}$
Data Carrier Detect (Inputs; Active Low)

These pins function as receiver enables if they are programmed as Auto Enables; otherwise, they may be used as general-purpose input pins. Both are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and may interrupt the CPU on both logic level transitions.

$\overline{DTR/REQA}$
Data Terminal Ready/Request
(Outputs; Active Low)

These outputs follow the inverted state programmed into the DTR bit in WR5. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.

$\overline{RTSA}$, $\overline{RTSB}$
Request to Send (Outputs; Active Low)

When the Request to Send (RTS) bit in Write Register 5 is set, the $\overline{RTS}$ signal goes Low. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. In SYNC mode, or in asynchronous mode with Auto Enable off, the $\overline{RTS}$ pins strictly follow the inverted state of the RTS bit. Both pins can be used as general-purpose outputs.

In SDLC mode, the AUTO RTS RESET enhancement described later in this document brings $\overline{RTS}$ High after the last 0 of the closing flag leaves the TxD pin.

$\overline{SYNCA}$, $\overline{SYNCB}$
Synchronization (Inputs/Outputs; Active Low)

These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{SYNC}$ must be driven Low two receive clock cycles after the last bit in the SYNC character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{SYNC}$.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which SYNC characters are recognized. The SYNC condition is not latched, so these outputs are active each time a SYNC pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

$\overline{W/REQA}$, $\overline{W/REQB}$
Wait/Request (Outputs; Open drain when programmed for a Wait function, driven High or Low when programmed for a Request function)

These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

Control

A/$\overline{B}$
Channel A/Channel B Select (Input)

This signal selects the channel in which the Read or Write operation occurs.

$\overline{CE}$
Chip Enable (Input; Active Low)

This signal selects the SCC for a Read or Write operation.

D/$\overline{C}$
Data/Control Select (Input)

This signal defines the type of information transferred to or from the SCC. A High means data is transferred; a Low indicates a command is transferred.

Data Bus

D_0–D_7
Data Bus (Input/Output; Three State)

These lines carry data and commands to and from the SCC.

Interrupt

IEI
Interrupt Enable In (Input; Active High)

IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

IEO
Interrupt Enable Out (Output; Active High)

IEO is High only if IEI is High and the CPU is not servicing an SCC interrupt or the SCC is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

$\overline{INT}$
Interrupt Request (Output; Active Low, Open Drain)

This signal is activated when the SCC requests an interrupt.

$\overline{INTACK}$
Interrupt Acknowledge (Input; Active Low)

This signal indicates an active interrupt acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of PCLK.

Serial Data

RxDa, RxDB
Receive Data (Inputs; Active High)

These input signals receive serial data at standard TTL levels.

TxDA, TxDB
Transmit Data (Outputs; Active High)

These output signals transmit serial data at standard TTL levels.

Miscellaneous

GND
Ground

PCLK
Clock (Input)

This is the master SCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock. PCLK is a TTL-level signal. Maximum transmit rate is 1/4 PCLK.

V_{CC}
+ 5 V Power Supply

ARCHITECTURE

The ESCC internal structure includes two full-duplex channels, two 10 × 19 bit SDLC/HDLC frame status FIFOs, two baud rate generators, internal control and interrupt logic, and a bus interface to a non-multiplexed bus. Associated with each channel are a number of Read and Write registers for mode control and status information, as well as logic necessary to interface with modems or other external devices (see Logic Symbol).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (Write) registers, two SYNC character (Write) registers, and four status (Read) registers. In addition, each baud rate generator has two (Read/Write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a Write register for the interrupt vector accessible through either channel, a Write-only Master Interrupt Control register, and three Read registers: one containing the vector with status information (Channel B only), one containing the vector without status (A only), and one containing the interrupt pending bits (A only).

The registers for each channel are designated as follows:

WR0–WR15—Write Registers 0 through 15. An additional Write register, WR7 Prime (WR7'), is available for enabling or disabling additional SDLC/HDLC enhancements if bit D_0 of WR15 is set.

RR0–RR3, RR10, RR12, RR13, RR15—Read Registers 0 through 3, 10, 12, 13, and 15.

If bit D_2 of WR15 is set, then two additional Read registers, RR6 and RR7, are available. These registers are used with the 10 × 19 bit Frame Status FIFO.

Table 1 lists the functions assigned to each Read and Write register. The ESCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

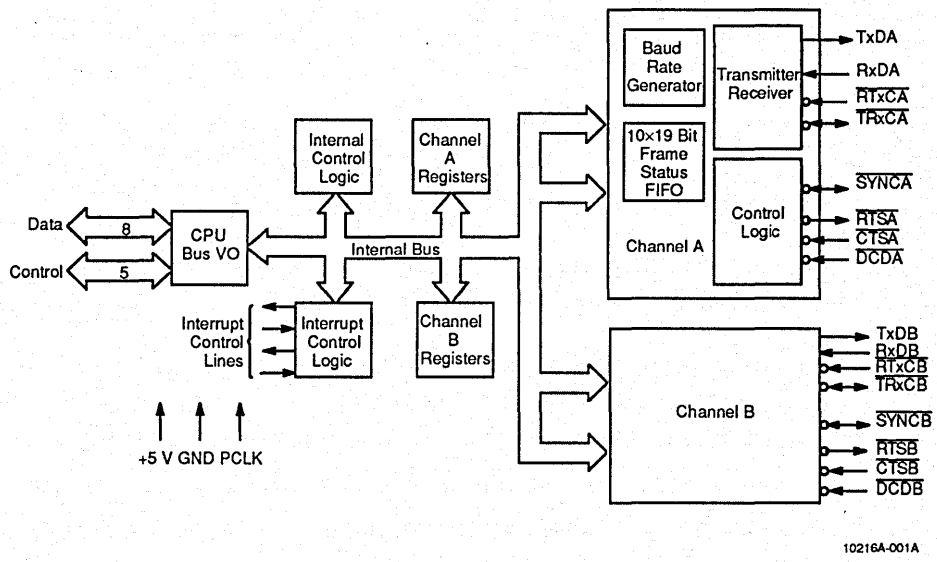


Figure 1. Block Diagram of ESCC Architecture

Data Path

The transmit and receive data path illustrated in Figure 2 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data are routed through one of several paths (data or CRC) depending on the selected mode (the character length in asynchronous modes also determines the data path).

The transmitter has an 8-bit transmit data buffer register loaded from the internal data bus and a 20-bit transmit shift register that can be loaded either from the sync-character registers or from the transmit data register. Depending on the operational mode, outgoing data are routed through one of four main paths before they are transmitted from the Transmit Data output (TxD).

Table 1. Read and Write Register Functions

Read Register Functions		Write Register Functions	
RR0	Transmit/Receive buffer status and External status	**WR0**	Command Register, Register Pointers CRC initialize, initialization commands for the various modes, shift right/shift left command
RR1	Special Receive Condition status (also 10 × 19 bit FIFO Frame Reception Status if WR15 bit D_2 is set)	**WR1**	Interrupt conditions and data transfer mode definition
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)	**WR2**	Interrupt vector (accessed through either channel)
RR3	Interrupt Pending bits (Channel A only)	**WR3**	Receive parameters and control
RR6	LSB Byte Count (14-bit counter) (if WR15 bit D_2 set)	**WR4**	Transmit/Receive miscellaneous parameters and modes
RR7	MSB Byte Count (14-bit counter) and 10 × 19 bit FIFO Status (if WR15 bit D_2 is set)	**WR5**	Transmit parameters and controls
RR8	Receive buffer	**WR6**	Sync character or SDLC address field
RR10	Miscellaneous XMTR, RCVR status	**WR7**	Sync character or SDLC flag
RR12	Lower byte of baud rate generator time constant	**WR7'**	SDLC/HDLC enhancements (if bit D_0 of WR15 is set)
RR13	Upper byte of baud rate generator time constant	**WR8**	Transmit buffer
RR15	External/Status interrupt information	**WR9**	Master interrupt control and reset (accessed through either channel)
		WR10	Miscellaneous transmitter/receiver control bits, data encoding
		WR11	Clock mode control, Rx and Tx clock source
		WR12	Lower byte of baud rate generator time constant
		WR13	Upper byte of baud rate generator time constant
		WR14	Miscellaneous control bits, DPLL control
		WR15	External/Status interrupt control

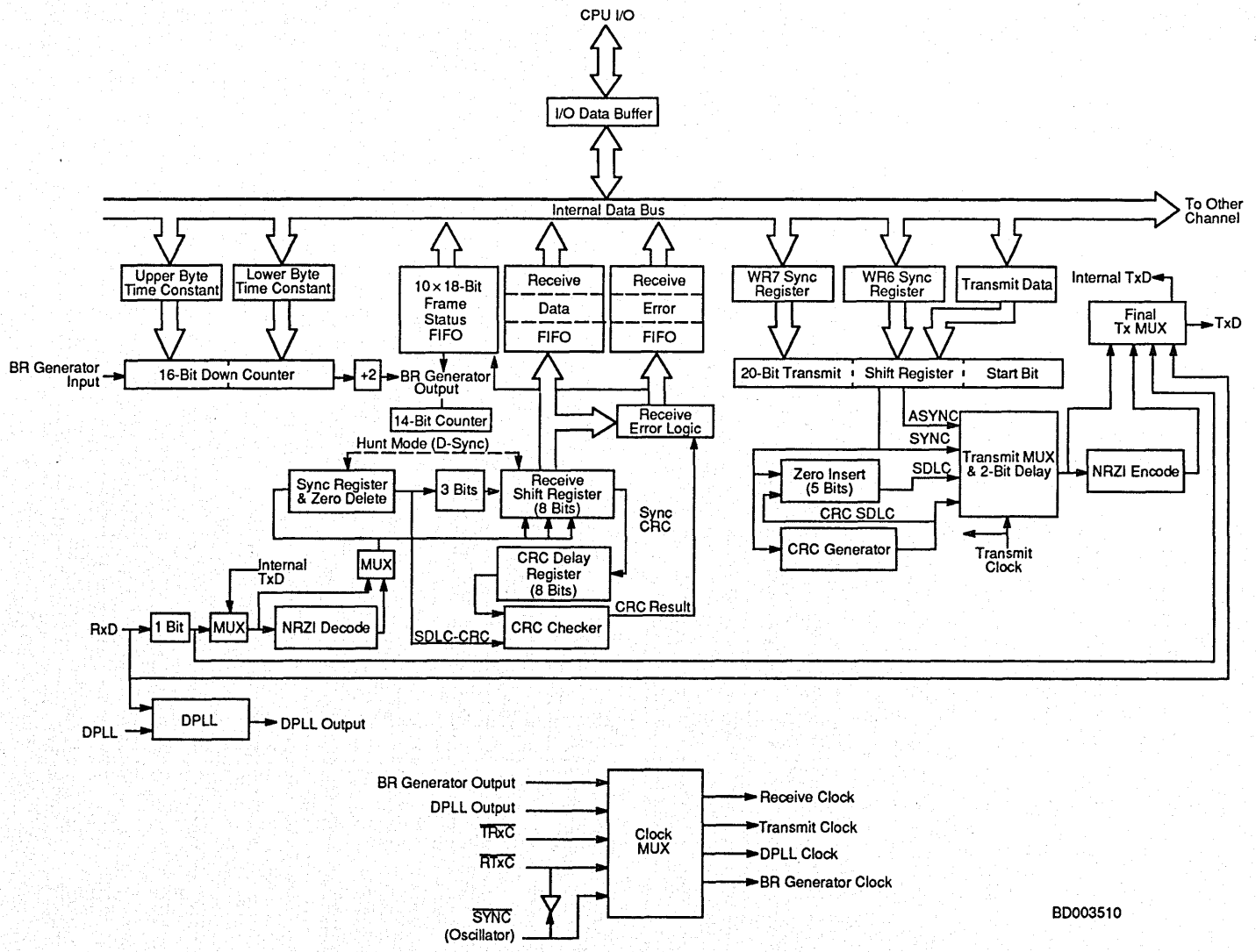


Figure 2. Data Path

DETAILED DESCRIPTION

The functional capabilities of the ESCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, it interacts with the CPU and provides vectored interrupts and handshaking signals.

Data Communications Capabilities

The ESCC provides two independent full-duplex channels programmable for use in any common asynchronous or SYNC data-communication protocol. Figure 3 and the following description briefly detail these protocols.

Asynchronous Modes

Transmission and reception can be accomplished independently on each channel with 5 to 8 bits per character, plus optional even or odd parity. The transmitters can supply 1, 1 1/2, or 2 stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input. If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing of error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of framing error as a new start bit; a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The ESCC does not require symmetric transmit and receive clock signals—a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In asynchronous modes, the $\overline{SYNC}$ pin may be programmed as an input used for functions, such as monitoring a ring indicator.

Synchronous Modes

The ESCC supports both byte-oriented and bit-oriented synchronous communication. SYNC byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit SYNC character (Monosync), any 12-bit or 16-bit SYNC pattern (Bisync), or with an external SYNC signal. Leading SYNC characters can be removed without interrupting the CPU.

Five- or 7-bit SYNC characters are detected with 8- or 16-bit patterns in the ESCC by overlapping the larger pattern across multiple incoming SYNC characters as shown in Figure 4.

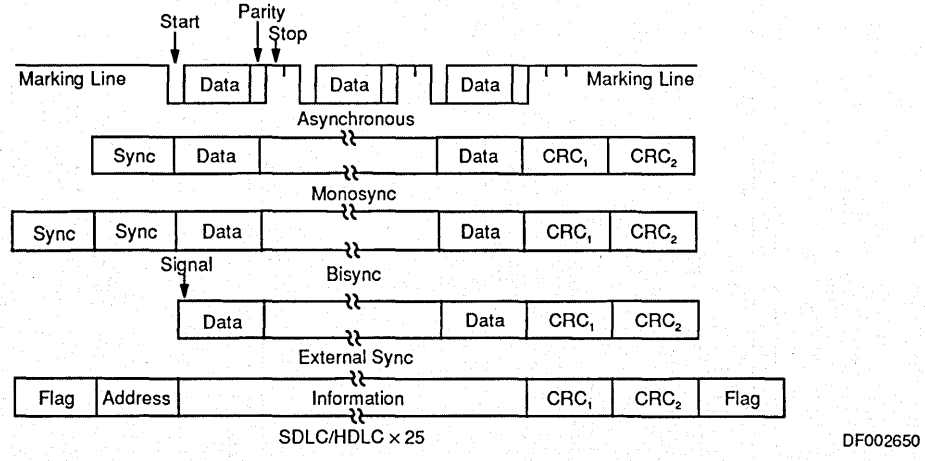


Figure 3. SCC Protocols

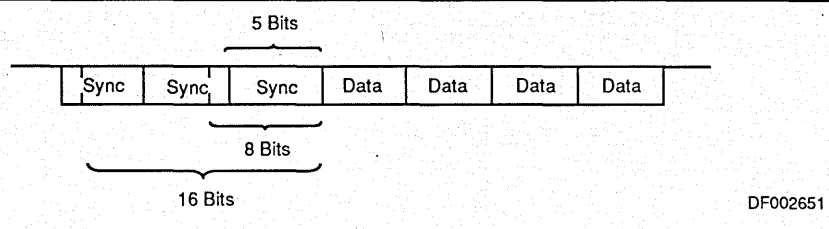


Figure 4. Detecting 5- or 7-Bit Synchronous Characters

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols, such as IBM BISYNC.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error-checking polynomials are supported. Either polynomial may be selected in BISYNC and MONOSYNC modes. Users may preset the CRC generator and checker to all 1s or all 0s. The ESCC also provides a feature that automatically transmits CRC data when no other data are available for transmission. This allows for high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there are no data or CRC to send in SYNC modes, the transmitter inserts 6-, 8-, or 16-bit SYNC characters, regardless of the programmed character length.

The ESCC supports SYNC bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero-bit insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the ESCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The ESCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to 8 bits per character can be sent allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{SYNC}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or 4 bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the ESCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1s or all 0s. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1X mode. The parity options available in asynchronous modes are available in synchronous modes.

The ESCC can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the ESCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The ESCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the ESCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC Loop Mode

The ESCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow and any number of secondary stations. In SDLC Loop mode, the ESCC performs the functions of a secondary station while an ESCC operating in regular SDLC mode can act as a controller (Figure 5).

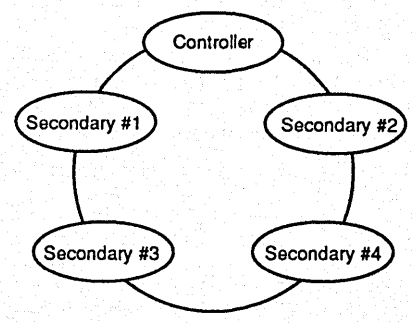


Figure 5. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, must pass these messages to the rest of the loop by retransmitting them with a 1-bit time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations farther down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the ESCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

Baud Rate Generator

Each channel in the ESCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On start-up, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching zero; the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the digital phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate where PCLK or $\overline{RTxC}$ is the baud rate generator input frequency in Hz. The clock mode is X1, X16, X32, or X64 as selected in Write Register 4, bits D_6 and D_7. Synchronous operation modes should select X1 and asynchronous should select X16, X32, or X64.

$$\text{Time Constant} = \left[\frac{\text{PCLK or RTxC Frequency}}{2\,(\text{Baud Rate})(\text{Clock Mode})}\right] - 2$$

The following formula relates the time constant to the baud rate. (The baud rate is in bits/second and the BR clock period is in seconds given by Clock Mode/Clock Frequency.)

$$\text{baud rate} = \frac{1}{2\,(\text{Time Constant} + 2) \times (\text{BR Clock Period})}$$

Time Constant Values for Standard Baud Rates at BR Clock = 3.9936 MHz

Rate (Baud)	Time Constant (decimal/Hex notation)		Error
19200	102	(0066)	0
9600	206	(00CE)	0
7200	275	(0113)	0.12%
4800	414	(019E)	0
3600	553	(0229)	0.06%
2400	830	(033E)	0
2000	996	(03E4)	0.04%
1800	1107	(0453)	0.03%
1200	1662	(067E)	0
600	3326	(0CFE)	0
300	6654	(19FE)	0
150	13310	(33FE)	0
134.5	14844	(39FC)	0.0007%
110	18151	(46E7)	0.0015%
75	26622	(67FE)	0
50	39934	(98FE)	0

Digital Phase-Locked Loop

The ESCC contains a digital phase-locked loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1/0 or 0/1). As long as no transitions are detected, the DPLL output will be free running and its input clock source will be divided by 32, producing an output clock without any phase jitter. Upon detecting a transition the DPLL will adjust its clock output (during the next counting cycle) by adding or subtracting a count of 1, thus producing a terminal count closer to the center of the bit cell. The adding or subtracting of a count of 1 will produce a phase jitter of ±5.63° on the output of the DPLL. Because the SCC's DPLL uses both edges of the incoming signal to compare with its clock source, the mark-space ratio (50%) of the incoming signal should not deviate by more than ±1.5% if proper locking is to occur.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15/16 counting transition.

The 32X clock for the DPLL can be programmed to come from either the $\overline{RTxC}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

Crystal Oscillator

When using a crystal oscillator to supply the receive or transmit clocks to a channel of the SCC, the user should:

1. Select a crystal oscillator that satisfies the following specifications:
   - 30 ppm @ 25°C
   - 50 ppm over temperatures of –20° to 70°C
   - 5 ppm/yr aging
   - 5-MW drive level
2. Place crystal across $\overline{RTxC}$ and $\overline{SYNC}$ pins
3. Place 30-pF capacitors to ground from both $\overline{RTxC}$ and $\overline{SYNC}$ pins
4. Set bit D_7 of WR11 to 1

Data Encoding

The ESCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a 1 is represented by a High level, and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level, and a 0 is represented by a change in level. In FM_1 (more properly, biphase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell, and a 0 is represented by no additional transition at the center of the bit cell. In FM_0 (biphase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the ESCC can be used to decode Manchester (biphase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0/1, the bit is a 0. If the transition is 1/0, the bit is a 1.

Auto Echo and Local Loopback

The ESCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes but works in SYNC and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{CTS}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed, and the programmer is responsible for disabling transmitter interrupts and $\overline{WAIT/REQUEST}$ on transmit.

The ESCC is also capable of Local Loopback. In this mode, TxD is RxD just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data, and RxD is ignored (except to be echoed out via TxD). The $\overline{CTS}$ and $\overline{DCD}$ inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in asynchronous, SYNC, and SDLC modes with NRZ, NRZI, or FM coding of the data stream.

I/O Interface Capabilities

The ESCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

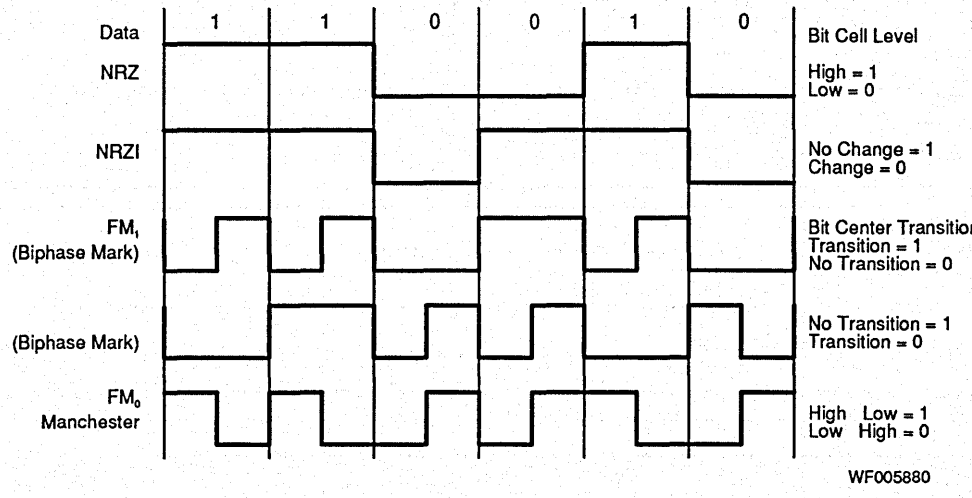


Figure 6. Data Encoding Methods

Polling

All interrupts are disabled. Three status registers in the ESCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts

When an ESCC responds to an Interrupt Acknowledge signal (INTACK) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 8 and 9).

To speed interrupt response time, the ESCC can modify 3 bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the ESCC (Transmit, Receive, and External/Status interrupts in both channels) has 3 bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write-only.

The other 2 bits are related to the Z-Bus interrupt priority chain (Figure 7). As a Z-Bus peripheral, the ESCC may request an interrupt only when no higher priority device is requesting one, for example, when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the A/D bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is set to 1 and the IEI input is High, the $\overline{INT}$ output is pulled Low, requesting an interrupt. In the ESCC, if the IE bit is set for an interrupt, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the ESCC and external to the ESCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the ESCC being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receive, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the Receive can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive condition
- Interrupt on all Receive Characters or Special Receive condition
- Interrupt on Special Receive condition only

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in asynchronous mode, end-of-frame in SDLC mode, and optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary Receive Character Available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first Receive Character Interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{SYNC}$ pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, a zero count in

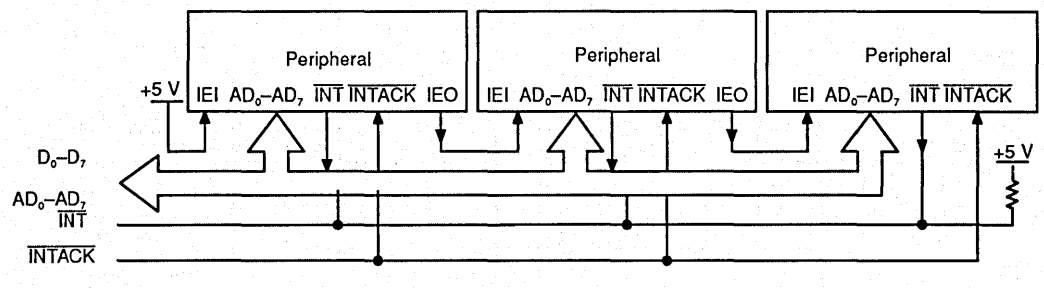


Figure 7. Z-Bus Interrupt Schedule

the baud rate generator, the detection of a Break (asynchronous mode), Abort (SDLC mode), or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the ESCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer

The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{\text{WAIT/REQUEST}}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{\text{WAIT/REQUEST}}$ output can be defined under software control as a $\overline{\text{WAIT}}$ line in the CPU Block Transfer mode or as a $\overline{\text{REQUEST}}$ line in the DMA Block Transfer mode.

To a DMA controller, the ESCC $\overline{\text{REQUEST}}$ output indicates that the ESCC is ready to transfer data to or from memory. To the CPU, the $\overline{\text{WAIT}}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{\text{DTR/REQUEST}}$ can be used as the transmit request line, thus allowing full-duplex operation under DMA control.

PROGRAMMING INFORMATION

Each channel has fifteen Write registers that are individually programmed from the system bus to configure the functional personality of each channel. Each channel also has eight Read registers from which the system can read Status, Baud rate, or Interrupt information.

On the Z85C30, only four data registers (Read and Write for Channels A and B) are directly selected by a High on the D/$\overline{\text{C}}$ input and the appropriate levels on the $\overline{\text{RD}}$, $\overline{\text{WR}}$, and A/$\overline{\text{B}}$ pins. All other registers are addressed indirectly by the content of Write Register 0 in conjunction with a Low on the D/$\overline{\text{C}}$ input and the appropriate levels on the $\overline{\text{RD}}$, $\overline{\text{WR}}$, and A/$\overline{\text{B}}$ pins. If bit D_3 in WR0 is 1 and bits 5 and 6 are 0, then bits 0, 1, and 2 address the higher registers 8 through 15. If bits 4, 5, and 6 contain a different code, bits 0, 1, and 2 address the lower registers 0 through 7 as shown in Table 2.

Writing to or reading from any register except RR0, WR0, and the data registers thus involves two operations:

First, write the appropriate code into WR0, then follow this by a Write or Read operation on the register thus specified. Bits 0 through 4 in WR0 are automatically cleared after this operation, so that WR0 then points to WR0 or RR0 again.

Channel A/Channel B selection is made by the A/$\overline{\text{B}}$ input (High = A, Low = B).

The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set and, finally, receiver or transmitter enable.

Table 2. Register Addressing

D/$\overline{\text{C}}$	"Point High" Code In WR0:	D_2 In WR0:	D_1 In WR0:	D_0 In WR0:	Write Register	Read Register
High	Either Way	X	X	X	Data	Data
Low	Not True	0	0	0	0	0
Low	Not True	0	0	1	1	1
Low	Not True	0	1	0	2	2
Low	Not True	0	1	1	3	3
Low	Not True	1	0	0	4	(0)
Low	Not True	1	0	1	5	(1)
Low	Not True	1	1	0	6	(2)
Low	Not True	1	1	1	7	(3)
Low	True	0	0	0	Data	Data
Low	True	0	0	1	9	–
Low	True	0	1	0	10	10
Low	True	0	1	1	11	(15)
Low	True	1	0	0	12	12
Low	True	1	0	1	13	13
Low	True	1	1	0	14	(10)
Low	True	1	1	1	15	15

Read Registers

The ESCC contains eight Read registers [actually nine, counting the receive buffer (RR8) in each channel]. Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). In addition, if bit D_2 of WR15 is set, RR6 and RR7 are available for providing frame status from the 10 × 19 bit Frame Status FIFO. Figure 8 shows the formats for each Read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring, for example, when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

Write Registers

The ESCC contains 15 Write registers (16 counting WR8, the transmit buffer) in each channel. These Write registers are programmed separately to configure the functional "personality" of the channels. Two registers (WR2 and WR9) are shared by the two channels that can be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. In addition, if bit D_0 of WR15 is set, Write Register 7 prime (WR7') is available for programming additional SDLC/HDLC enhancements. When bit D_0 of WR15 is set, executing a write to WR7 actually writes to WR7' to further enhance the functional "personality" of each channel. Figure 9 shows the format of each Write register.

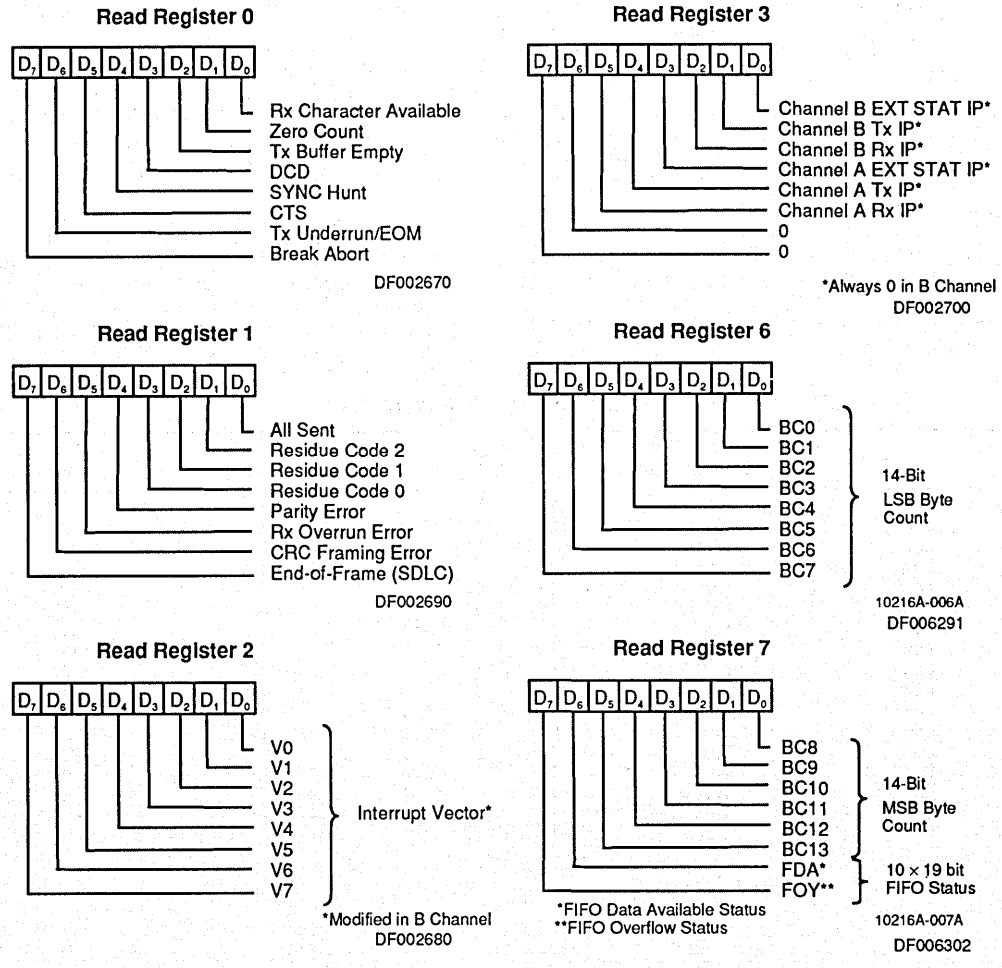


Figure 8. Read Register Bit Functions

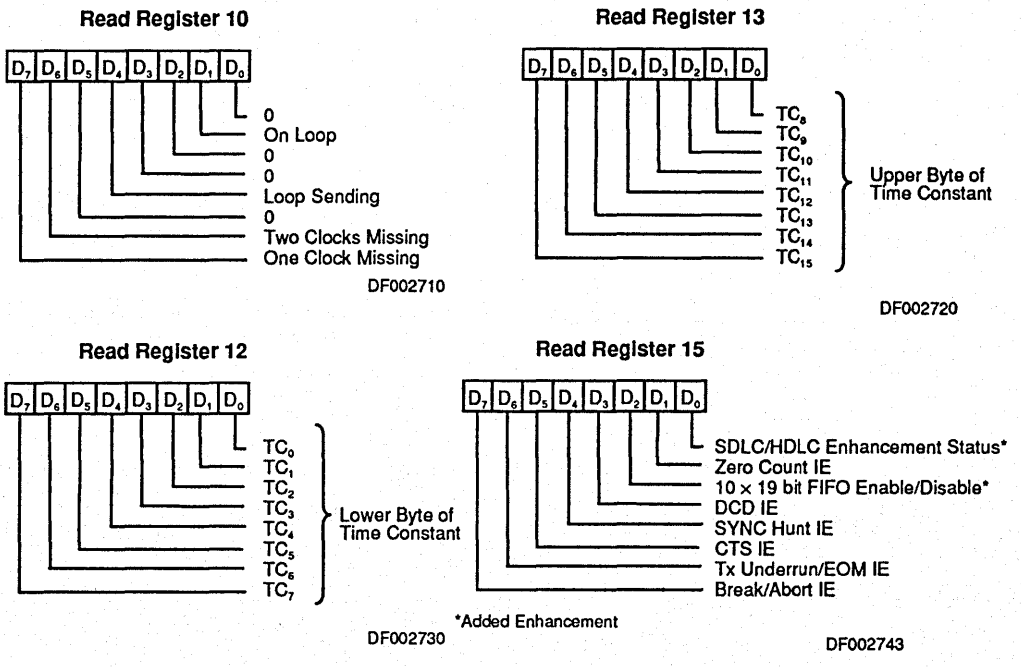


Figure 8. Read Register Bit Functions (continued)

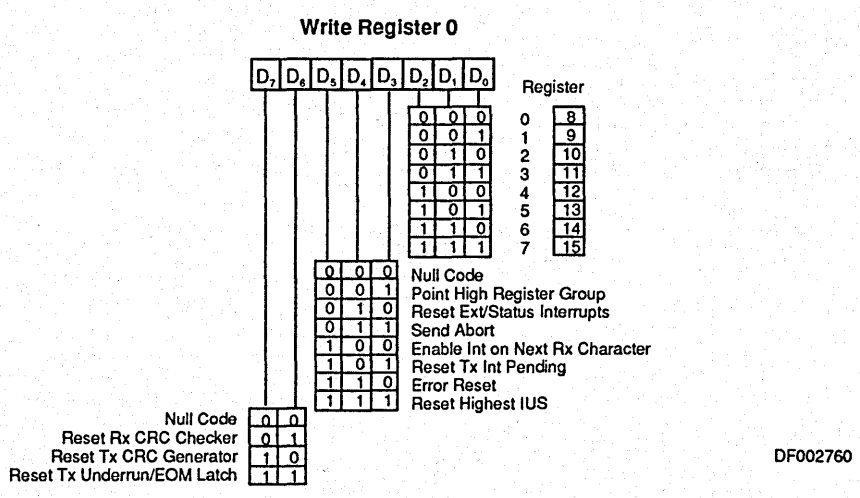


Figure 9. Write Register Bit Functions

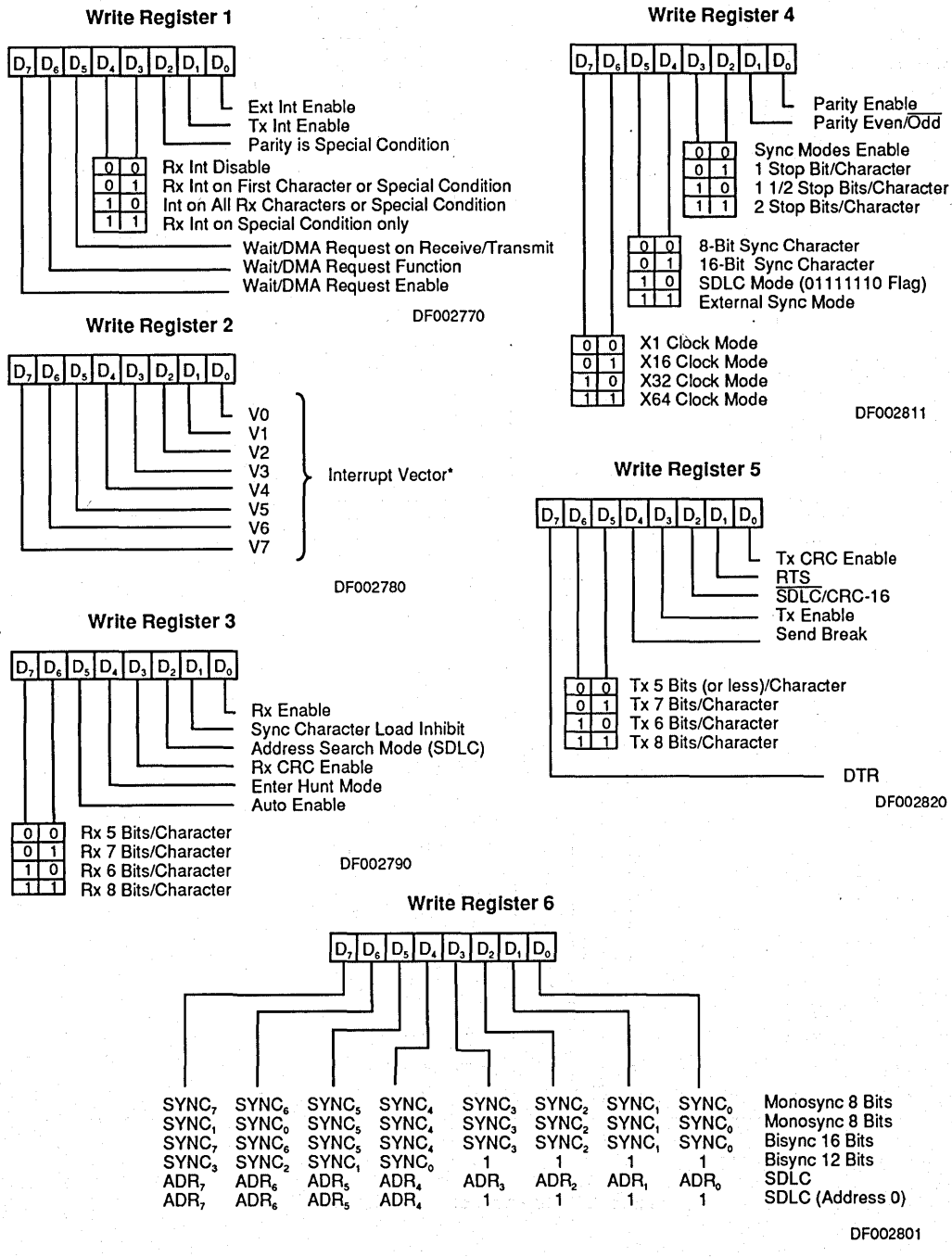
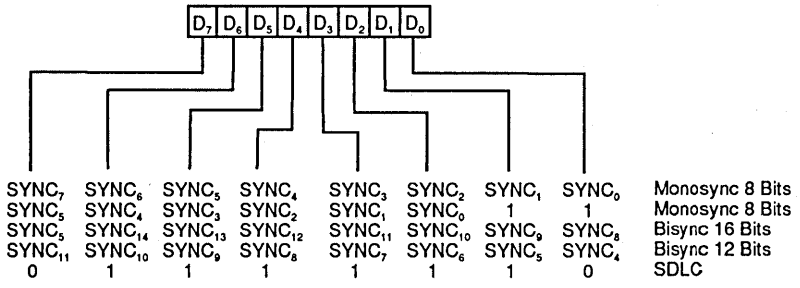


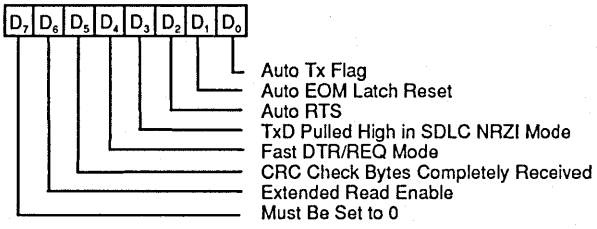
Figure 9. Write Register Bit Functions (continued)

Write Register 7



DF002831

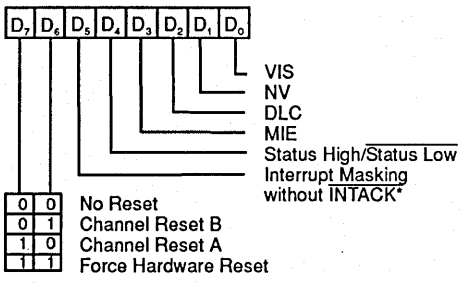
Write Register 7'



10216A-008A
DF006331

2

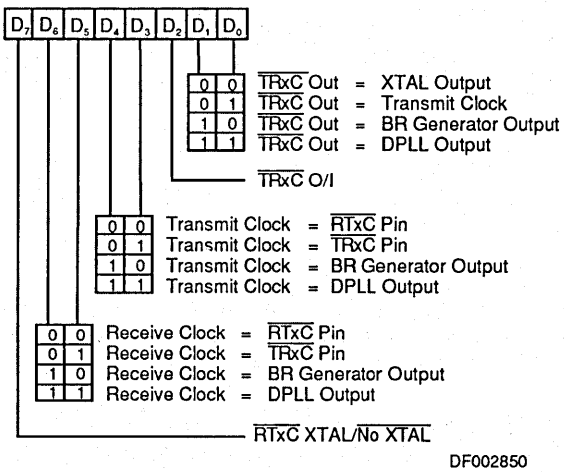
Write Register 9



*Added Enhancement

DF002842

Write Register 11



DF002850

Figure 9. Write Register Bit Functions (continued)

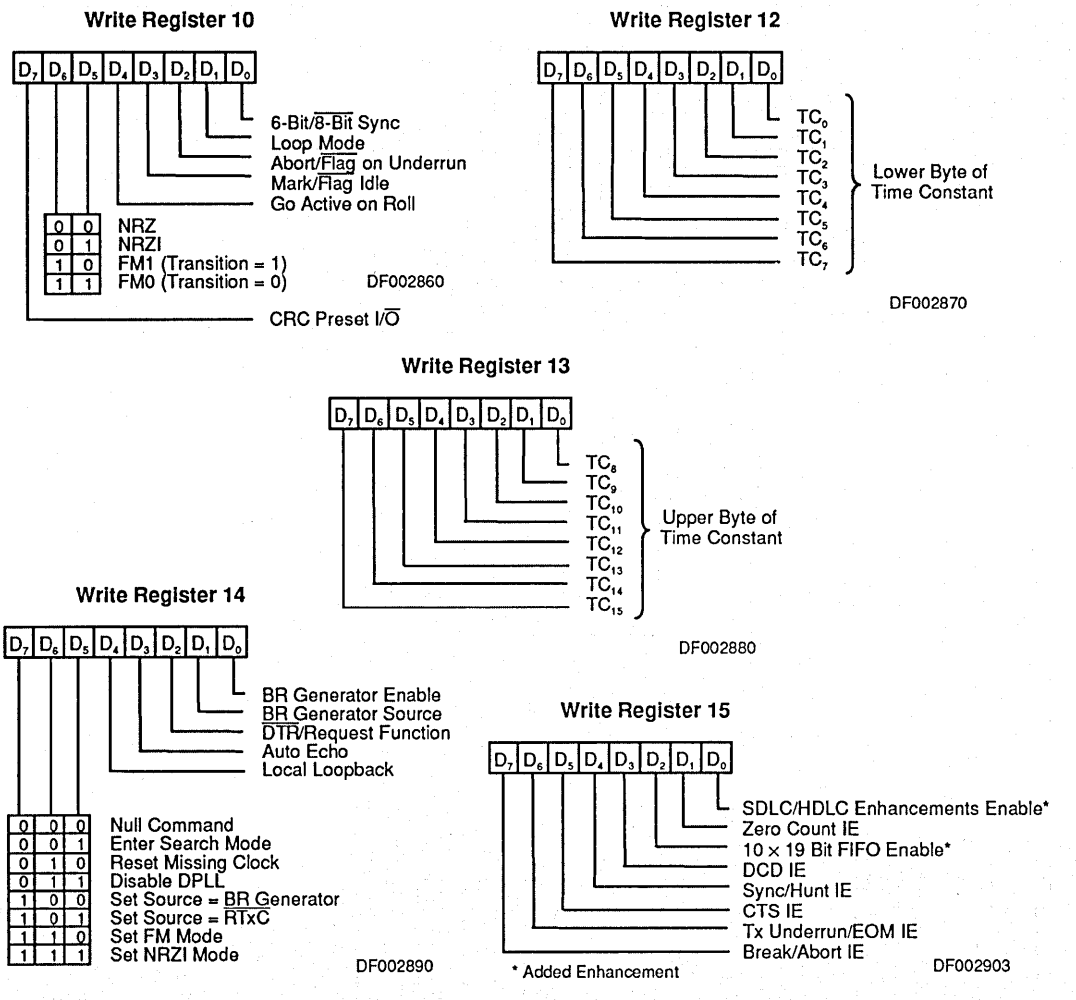


Figure 9. Write Register Bit Functions (continued)

Z85C30 Timing

The ESCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the ESCC. The recovery time required for proper operation is specified from the falling edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the ESCC, to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the ESCC. This time must be at least 3 1/2 PCLK regardless of which register or channel is being accessed.

Read Cycle Timing

Figure 10 illustrates Read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

Write Cycle Timing

Figure 11 illustrates Write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened. Data must be valid before the rising edge of $\overline{WR}$.

Interrupt Acknowledge Cycle Timing

Figure 12 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes Low and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the ESCC and IEI is High when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the ESCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on D_0–D_7 ; it then sets the appropriate Interrupt-Under-Service latch internally.

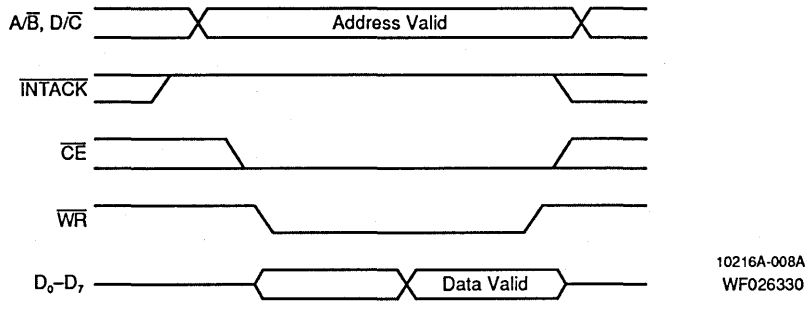


Figure 10. Read Cycle Timing

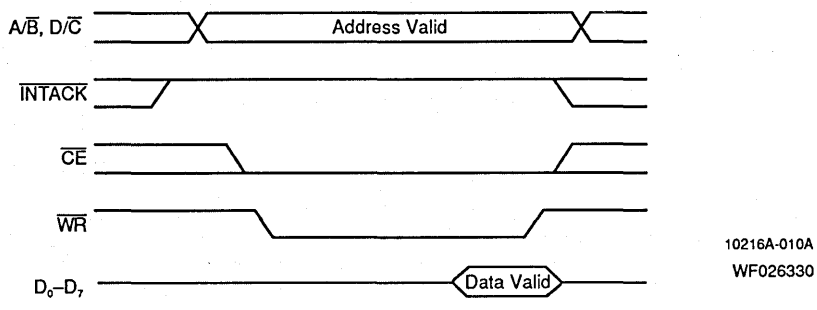


Figure 11. Write Cycle Timing

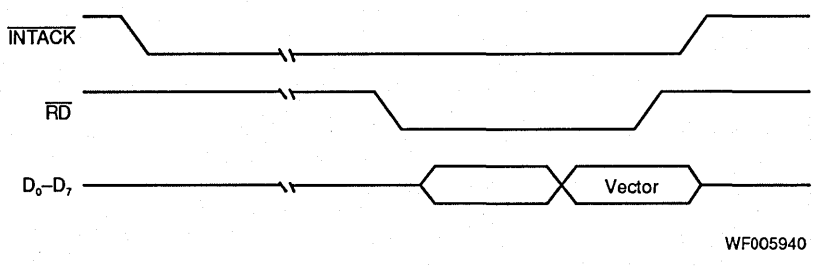


Figure 12. Interrupt Acknowledge Cycle Timing

FIFO

FIFO Enhancements

When used with a DMA controller, the Z85C30 Frame Status FIFO enhancement maximizes the ESCC's ability to receive high-speed back-to-back SDLC messages while minimizing frame overruns due to CPU latencies in responding to interrupts.

Additional logic was added to the industry-standard NMOS SCC consisting of a 10-deep by 19-bit status FIFO, a 14-bit receive byte counter, and control logic as shown in Figure 13. The 10 × 19 bit status FIFO is separate from the existing 3-byte receive data and error FIFOs.

When the enhancement is enabled, the status in Read Register 1 (RR1) and byte count for the SDLC frame will be stored in the 10 × 19 bit status FIFO. This allows the DMA controller to transfer the next frame into memory while the CPU verifies that the message was properly received.

Summarizing the operation, data is received, assembled, and loaded into the 3-byte receive FIFO before being transferred to memory by the DMA controller. When a flag is received at the end of an SDLC frame, the frame byte count from the 14-bit counter and 5 status bits are loaded into the status FIFO for verification by the CPU. The CRC checker is automatically reset in preparation for the next frame, which can begin immediately. Since the byte count and status are saved for each frame, the message integrity can be verified at a later time. Status information for up to 10 frames can be stored before a status FIFO overrun could occur.

If receive interrupts are enabled while the 10 × 19 FIFO is enabled, an SDLC end-of-frame special condition will not lock the 3-byte receive data FIFO. An SDLC end-of-frame still locks the 3-byte receive data FIFO in "Interrupt on first Receive Character or Special Condition" and "Interrupt on Special Condition Only" modes when the 10 × 19 FIFO is disabled. This feature allows

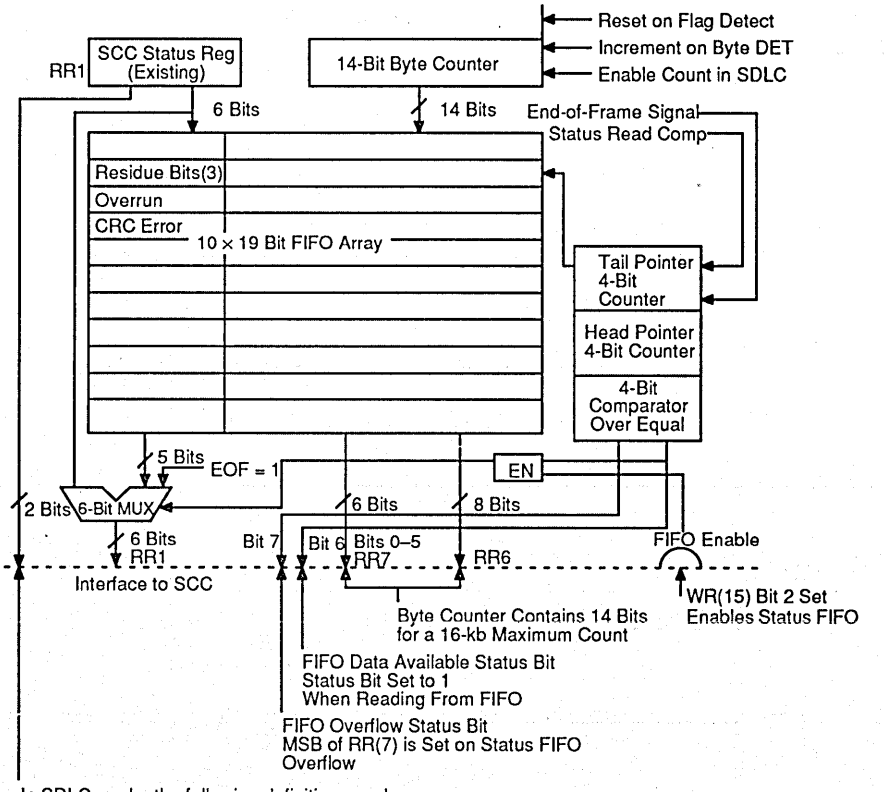


Figure 13. SCC Status Register Modifications

the 10 × 19 SDLC FIFO to accept multiple SDLC frames without CPU intervention at the end of each frame.

FIFO Detail

For a better understanding of details of the FIFO operation, refer to the block diagram contained in Figure 13.

Enable/Disable

This FIFO is implemented so that it is enabled when WR15 bit 2 is set and the ESCC is in the SDLC/HDLC mode, otherwise the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or via a channel or power-on reset). When the FIFO mode is disabled, the ESCC is completely downward-compatible with the NMOS Z8530. The FIFO mode is disabled on power-up (WR15 bit 2 is set to 0 on reset). The effects of backward compatibility on the register set are that RR4 is an image of RR0, RR5 is an image of RR1, RR6 is an image of RR2, and RR7 is an image of RR3. For the details of the added registers, refer to Figure 15. The status of the FIFO Enable signal can be obtained by reading RR15 bit 2. If the FIFO is enabled, the bit will be set to 1; otherwise, it will be reset.

Read Operation

When WR15 bit 2 is set and the FIFO is not empty, the next read to status register RR1 or the additional registers RR7 and RR6 will actually be from the FIFO. Reading status register RR1 causes one location of the FIFO to be emptied, so status should be read after reading the byte count, otherwise the count will be incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched to allow status to be read directly from the status register, and reads from RR7 and RR6 will contain bits that are undefined. Bit 6 of RR7 (FIFO Data Available) can be used to determine if status data is coming from the FIFO or directly from the status register, since it is set to 1 whenever the FIFO is not empty.

Because not all status bits are stored in the FIFO, the All Sent, Parity, and EOF bits will bypass the FIFO. The status bits sent through the FIFO will be Residue Bits (3), Overrun, and CRC Error.

The sequence for proper operation of the byte count and FIFO logic is to read the registers in the following order, RR7, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR7 latches the FIFO empty/full status bit (bit 6) and steers the status multiplexer to read from the SCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic is added to prevent a FIFO underflow condition).

Write Operation

When the end of an SDLC frame (EOF) has been received and the FIFO is enabled, the contents of the status and byte-count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, the MSB of RR7 (FIFO Overflow) is set to indicate the overflow. This bit and the FIFO control logic are reset by disabling and reenabling the FIFO control bit (WR15 bit 2). For details of FIFO control timing during an SDLC frame, refer to Figure 14.

Byte Counter Detail

The 14-bit byte counter allows for packets up to 16K bytes to be received. For a better understanding of its operation, refer to Figures 13 and 14.

Enable

The byte counter is enabled when the SCC is in the SDLC/HDLC mode and WR15 bit 2 is set to 1.

Reset

The byte counter is reset whenever an SDLC flag character is received. The reset is timed so that the contents of the byte counter are successfully written into the FIFO.

Increment

The byte counter is incremented by writes to the data FIFO. The counter represents the number of bytes received by the SCC, rather than the number of bytes transferred from the SCC. (These counts may differ by

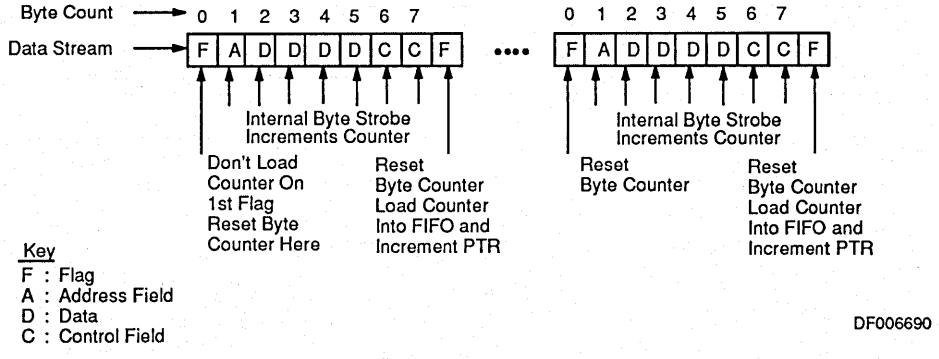


Figure 14. SDLC Byte Counting Detail

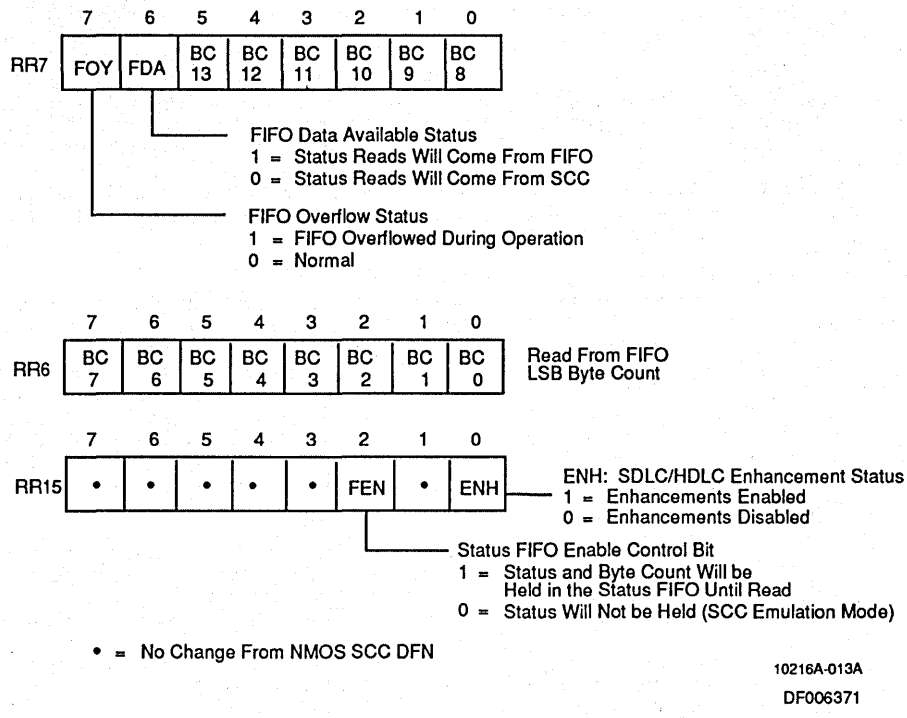


Figure 15. SCC Additional Registers

up to the number of bytes in the receive data FIFO contained in the SCC.)

Z85C30 SDLC/HDLC Enhancement Register Access

SDLC/HDLC enhancements on the Z85C30 are enabled or disabled via bits D_2 or D_0 in WR15. Bit D_2 determines whether or not the 10 × 19 bit SDLC/HDLC frame status FIFO is enabled while bit D_0 determines whether or not other enhancements are enabled via WR7'. Table 3 shows what functions on the Z85C30 are enabled when these bits are set.

When bit D_2 of WR15 is set to 1, two additional registers (RR6 and RR7) per channel specific to the 10 × 19 bit Frame Status FIFO are made available. The Z85C30

Table 3. Enhancement Options

WR15 Bit D_2 10 × 19 Bit FIFO Enabled	WR15 Bit D_0 SDLC/HDLC Enhancement Enabled	WR7' Bit D_6 Extended Read Enabled	Functions Enabled
1	0	x	10 × 19 bit FIFO enhancement enabled only
0	1	0	SDLC/HDLC enhancements enabled only
0	1	1	SDLC/HDLC enhancements enabled with extended read enabled
1	1	0	10 × 19 bit FIFO and SDLC/HDLC enhancements enabled
1	1	1	10 × 19 bit FIFO and SDLC/HDLC enhancements with extended read enabled

register map when this function is enabled is shown in Table 4.

Bit D_0 of WR15 determines whether or not other enhancements pertinent only to SDLC/HDLC mode operation are available for programming via WR7′ as shown below. Write Register 7 prime (WR7′) can be written to when bit D_0 of WR15 is set to 1. When this bit is set, writing to WR7 (flag register) actually writes to WR7′. If bit D_6 of this register is set to 1, previously unreadable registers WR3, WR4, WR5, and WR10 are readable by the processor. In addition, WR7′ is also readable by having this bit set. WR3 is read when a bogus RR9 register is accessed during a read cycle. WR10 is read by accessing RR11, and WR7′ is accessed by executing a read to RR14. The Z85C30 register map with bit D_0 of WR15 and bit D_6 of WR7′ set is shown in Table 5.

If both bits D_0 and D_2 of WR15 are set to 1 and D_6 of WR7′ is set to 1, then the Z85C30 register map is as shown in Table 6.

Auto RTS Reset

On the CMOS ESCC, if bit D_0 of WR15 and bit D_2 of WR7′ are set to 1 and the channel is in SDLC mode, the $\overline{RTS}$ pin may be reset early in the Tx Underrun routine and the $\overline{RTS}$ pin will remain active until the last 0 bit of the closing flag leaves the TxD pin as shown in Figure 16. Note that

Table 4. 10×19 Bit FIFO Enabled

A/$\overline{B}$	PNT_2	PNT_1	PNT_0	Write	Read
0	0	0	0	WR0B	RR0B
0	0	0	1	WR1B	RR1B
0	0	1	0	WR2	RR2B
0	0	1	1	WR3B	RR3B
0	1	0	0	WR4B	(RR0B)
0	1	0	1	WR5B	(RR1B)
0	1	1	0	WR6B	RR6B
0	1	1	1	WR7B	RR7B
1	0	0	0	WR0A	RR0A
1	0	0	1	WR1A	RR1A
1	0	1	0	WR2	RR2A
1	0	1	1	WR3A	RR3A
1	1	0	0	WR4A	(RR0A)
1	1	0	1	WR5A	(RR1A)
1	1	1	0	WR6A	RR6A
1	1	1	1	WR7A	RR7A
		With the Point High command:			
0	0	0	0	WR8B	RR8B
0	0	0	1	WR9	RR13B
0	0	1	0	WR10B	RR10B
0	0	1	1	WR11B	(RR15B)
0	1	0	0	WR12B	RR12B
0	1	0	1	WR13B	RR13B
0	1	1	0	WR14B	(RR10B)
0	1	1	1	WR15B	RR15B
1	0	0	0	WR8A	RR8A
1	0	0	1	WR9	(RR13A)
1	0	1	0	WR10A	RR10A
1	0	1	1	WR11A	(RR15A)
1	1	0	0	WR12A	RR12A
1	1	0	1	WR13A	RR13A
1	1	1	0	WR14A	(RR10A)
1	1	1	1	WR15A	RR15A

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Must Be Set to 0	Ext. Read Enable	Rx comp. CRC	$\overline{DTR}/\overline{REQ}$ Fast Mode	Force TxD High	SDLC/HDLC Auto $\overline{RTS}$ Turnoff	SDLC/HDLC Auto EOM Reset	SDLC/HDLC Auto Tx Flag

WR7′—SDLC/HDLC Programmable Enhancements*

*Note: Options 3, 4, 5, and 6 may be used regardless of whether SDLC/HDLC mode is selected.

Table 5. SDLC/HDLC Enhancements Enabled

A/$\overline{B}$	PNT_2	PNT_1	PNT_0	Write	Read
0	0	0	0	WR0B	RR0B
0	0	0	1	WR1B	RR1B
0	0	1	0	WR2	RR2B
0	0	1	1	WR3B	RR3B
0	1	0	0	WR4B	RR4B (WR4B)
0	1	0	1	WR5B	RR5B (WR5B)
0	1	1	0	WR6B	(RR2B)
0	1	1	1	WR7B	(RR3B)
1	0	0	0	WR0A	RR0A
1	0	0	1	WR1A	RR1A
1	0	1	0	WR2	RR2A
1	0	1	1	WR3A	RR3A
1	1	0	0	WR4A	RR4A (WR4A)
1	1	0	1	WR5A	RR5A (WR5A)
1	1	1	0	WR6A	(RR2A)
1	1	1	1	WR7A	(RR3A)
				With the Point High command:	
0	0	0	0	WR8B	RR8B
0	0	0	1	WR9	RR9 (WR3B)
0	0	1	0	WR10B	RR10B
0	0	1	1	WR11B	RR11B (WR10B)
0	1	0	0	WR12B	RR12B
0	1	0	1	WR13B	RR13B
0	1	1	0	WR14B	RR14B (WR7'B)
0	1	1	1	WR15B	RR15B
1	0	0	0	WR8A	RR8A
1	0	0	1	WR9	RR9A (WR3A)
1	0	1	0	WR10A	RR10A
1	0	1	1	WR11A	RR11A (WR10A)
1	1	0	0	WR12A	RR12A
1	1	0	1	WR13A	RR13A
1	1	1	0	WR14A	RR14A (WR7A)
1	1	1	1	WR15A	RR15A

in order for this to function properly, bits D_3 and D_2 of WR10 must be set to 1 and 0, respectively.

CRC Character Reception

NMOS Z8530H

On the NMOS 8530H, when the end-of-frame flag is detected, the contents of the Receive Shift Register are transferred to the Receive Data FIFO regardless of the number of bits accumulated. Because of the 3-bit delay between the Receive SYNC Register and Receive Shift Register, the last 2 bits of the CRC check character received are never transferred to the Receive Data FIFO. Thus, the received CRC characters are unavailable for use.

CMOS Z85C30

On the Z85C30, the option of being able to receive the complete CRC characters generated by the transmitter is provided when both bit D_0 of WR15 and bit D_5 of WR7' are set to 1. When these 2 bits are set and an end-of-frame flag is detected, the last 2 bits of the CRC will be clocked into the Receive Shift Register before its contents are transferred to the Receive Data FIFO. The data-CRC boundary and CRC character bit formats for each Residue Code provided are shown in Figures 17A through 17D for each character length selected.

Table 6. SDLC/HDLC Enhancements and 10×19 Bit FIFO Enabled

A/$\overline{B}$	PNT_2	PNT_1	PNT_0	Write	Read
0	0	0	0	WR0B	RR0B
0	0	0	1	WR1B	RR1B
0	0	1	0	WR2	RR2B
0	0	1	1	WR3B	RR3B
0	1	0	0	WR4B	RR4B (WR4B)
0	1	0	1	WR5B	RR5B (WR5B)
0	1	1	0	WR6B	RR6B
0	1	1	1	WR7B	RR7B
1	0	0	0	WR0A	RR0A
1	0	0	1	WR1A	RR1A
1	0	1	0	WR2	RR2A
1	0	1	1	WR3A	RR3A
1	1	0	0	WR4A	RR4A (WR4A)
1	1	0	1	WR5A	RR5A (WR5A)
1	1	1	0	WR6A	RR6A
1	1	1	1	WR7A	RR7A
				With the Point High command:	
0	0	0	0	WR8B	RR8B
0	0	0	1	WR9	RR9 (WR3B)
0	0	1	0	WR10B	RR10B
0	0	1	1	WR11B	RR11B (WR10B)
0	1	0	0	WR12B	RR12B
0	1	0	1	WR13B	RR13B
0	1	1	0	WR14B	RR14B (WR7′B)
0	1	1	1	WR15B	RR15B
1	0	0	0	WR8A	RR8A
1	0	0	1	WR9	RR9A (WR3A)
1	0	1	0	WR10A	RR10A
1	0	1	1	WR11A	RR11A (WR10A)
1	1	0	0	WR12A	RR12A
1	1	0	1	WR13A	RR13A
1	1	1	0	WR14A	RR14A (WR7′A)
1	1	1	1	WR15A	RR15A

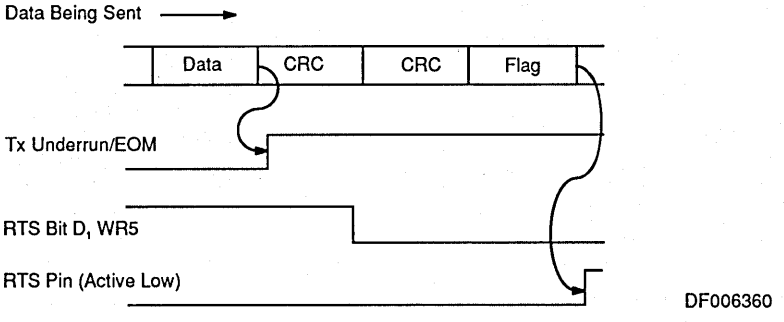


Figure 16. Auto $\overline{RTS}$ Reset Mode

Residue Code 012 001

D	D	D	D	D	C_0	C_1	C_2
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 101

D	D	D	D	D	D	C_0	C_1
D	C_0	C_1	C_2	C_3	C_4	C_5	C_6
C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 100

D	D	D	D	D	D	D	C_0
D	D	C_0	C_1	C_2	C_3	C_4	C_5
C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 010

D	D	D	D	D	D	D	D
D	D	D	C_0	C_1	C_2	C_3	C_4
C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 110

D	D	D	D	D	D	D	D
D	D	D	D	C_0	C_1	C_2	C_3
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

10216A-015A
TB001170

Figure 17A. 5 Bits/Character

Residue Code 012/010

D	D	D	D	D	D	C_0	C_1
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/110

D	D	D	D	D	D	D	C_0
D	C_0	C_1	C_2	C_3	C_4	C_5	C_6
C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/001

D	D	D	D	D	D	D	D
D	D	C_0	C_1	C_2	C_3	C_4	C_5
C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/101

D	D	D	D	D	D	D	D
D	D	D	C_0	C_1	C_2	C_3	C_4
C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/011

D	D	D	D	D	D	D	D
D	D	D	D	C_0	C_1	C_2	C_3
C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/100

D	D	D	D	D	D	D	D
D	D	D	D	D	C_0	C_1	C_2
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

10216A-016A
TB001180

Figure 17B. 6 Bits/Character

Residue Code 012 / 111

D	D	D	D	D	D	D	C_0
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 / 100

D	D	D	D	D	D	D	D
D	C_0	C_1	C_2	C_3	C_4	C_5	C_6
C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 / 010

D	D	D	D	D	D	D	D
D	D	C_0	C_1	C_2	C_3	C_4	C_5
C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 / 110

D	D	D	D	D	D	D	D
D	D	D	C_0	C_1	C_2	C_3	C_4
C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 / 001

D	D	D	D	D	D	D	D
D	D	D	D	C_0	C_1	C_2	C_3
C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 / 101

D	D	D	D	D	D	D	D
D	D	D	D	D	C_0	C_1	C_2
C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 / 011

D	D	D	D	D	D	D	D
D	D	D	D	D	D	C_0	C_1
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

10216A-017A
TB001190

Figure 17C. 7 Bits/Character

Residue Code 012/011

(No Residue)

D	D	D	D	D	D	D	D
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/111

(1 Residue Bit)

D	D	D	D	D	D	D	D
D	C_0	C_1	C_2	C_3	C_4	C_5	C_6
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/000

(2 Residue Bits)

D	D	D	D	D	D	D	D
D	D	C_0	C_1	C_2	C_3	C_4	C_5
C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/100

(3 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	C_0	C_1	C_2	C_3	C_4
C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/010

(4 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	D	C_0	C_1	C_2	C_3
C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/110

(5 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	D	D	C_0	C_1	C_2
C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/001

(6 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	D	D	D	C_0	C_1
C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/101

(7 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	D	D	D	D	C_0
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

10216A-018A
TB001-200

Figure 17D. 8 Bits/Character

Auto Flag Mode

On the NMOS Z8530H, if the transmitter is actively mark idling and a frame of data is ready to be transmitted, the Mark/Flag Idle bit must be set to 0 before data is written to WR8, otherwise the opening flag will not be sent properly. However, care must be exercised in doing this because the mark idle pattern (eight 1 bits) is transmitted 8 bits at a time, and all 8 bits must have transferred out of the Transmit Shift Register before a flag may be loaded and sent. If data is written into the Transmit Buffer (WR8) before the flag is loaded into the Transmit Shift Register, the data character written to WR8 will supersede flag transmission and the opening flag will not be transmitted.

On the CMOS Z85C30, if bit D_0 of WR15 is set to 1 and the ESCC is programmed for SDLC operation, an option is provided via bit D_0 of WR7′ that eliminates this requirement. If bit D_0 of WR7′ is set to 1 and a character is written to the Transmit Buffer while the transmitter is mark idling, the Mark/Flag Idle bit in WR10 need not be reset to 0 in order to have the opening flag sent because the transmitter will automatically send it before commencing to send data.

In addition, as long as bit D_0 of WR15 and bit D_1 of WR7′ are set to 1, the CRC transmit generator will be automatically preset to the initial state programmed by bit D_7 of WR10 (so the Reset Tx CRC Generator command is also not necessary), and the Tx Underrun/EOM latch will be reset automatically on every new frame sent. This ensures that an opening flag and proper CRC generation and transmission will always be sent without processor intervention under varying bus latency conditions.

Auto Transmit CRC Generator Preset

The NMOS Z8530H does not automatically preset the CRC generator prior to frame transmission. This must be done in software, usually during the initialization routine. This is accomplished by issuing the Reset Tx CRC Generator Command via WR0. For proper results, this command must be issued while the transmitter is enabled and idling and before any data are written to the Transmit Buffer.

In addition, if CRC is to be used, the transmit CRC generator must be enabled by setting bit D_0 of WR5 to 1. CRC is normally calculated on all characters between opening and closing flags, so this bit should be set to 1 at initialization and never changed.

On the CMOS Z85C30, setting bit D_0 of WR15 to 1 will cause the transmit CRC generator to be preset automatically every time an opening flag is sent, so the Reset Tx CRC Generator Command is not necessary.

Auto Tx Underrun/EOM Latch Reset

On the ESCC, the transmission of the CRC check characters is controlled by the Transmit CRC Enable bit in WR5 (D_0) and the Tx Underrun/EOM bit in RR0 (D_6). However, if the Transmit Enable bit is set to 0 when a transmit underrun (i.e., both the Transmit Buffer and Transmit Shift Register become empty) occurs, the CRC check characters will not be sent regardless of the state of the Tx Underrun/EOM bit.

If the Transmit Enable bit is set to 1 when an underrun occurs, then the state of the Tx Underrun/EOM bit and the Abort/Flag on Underrun bit in WR10 (D_2) determine the action taken by the transmitter. The Abort/Flag on Underrun bit may be set or reset by the processor, whereas the Tx Underrun/EOM bit is set by the transmitter and can only be reset by the processor via the Reset Tx Underrun/EOM Command in WR0.

If the Tx Underrun/EOM bit is set to 1 when an underrun occurs, the transmitter will close the frame by sending a flag; however, if this bit is set to 0, the frame data will be appended with either the accumulated CRC characters followed by a flag or an abort pattern followed by a flag, depending on the state of the Abort/Flag on Underrun bit in the WR10 (D_2). In either case, after the closing flag is sent, the transmitter will idle the transmission line as specified by the Mark/Flag Idle bit D_3 in WR10.

Hence, if the CRC check characters are to be properly appended to a frame, the Abort/Flag on Underrun bit must be set to 0, and the Reset Tx Underrun/EOM Command must be issued after the first but before the last character is written to the Transmit Buffer. This will ensure that either an abort or the CRC will be transmitted if an underrun occurs. Normally, the Abort/Flag on Underrun bit in WR10 should be set to 1 around the same time that the Tx Underrun/EOM bit is reset so that an abort will be sent if the transmitter accidentally underruns, and then set to 0 near the end of the frame to allow the correct transmission of CRC.

On the Z85C30, if bit D_0 of WR15 is set to 1, the option of having the Tx Underrun/EOM bit reset automatically at the start of every frame is provided via bit D_1 of WR7′. This helps alleviate the software burden of having to respond within one character time when high-speed data are being sent.

SDLC/HDLC NRZI Transmitter Disabling

On the NMOS Z8530H, if NRZI encoding is being used and the transmitter is disabled, the state of the TxD pin will depend on the last bit sent. That is, the TxD pin may either idle in a Low or High state as shown in Figure 18.

On the CMOS Z85C30, an option is provided that allows setting the TxD pin High when operating in SDLC mode with NRZI encoding enabled. If bit D_0 of WR15 is set to 1, then bit D_3 of WR7′ can be used to set the TxD pin High. Note that the operation of this bit is independent of the Tx Enable bit in WR5. The Tx Enable bit in WR5 is used to disable and enable the transmitter, whereas bit D_3 of WR7′ acts as a pseudo transmitter disable and enable by just forcing the TxD pin High when set even though the transmitter may actually be mark or flag idling. Care must be used when setting this bit because any character being transmitted at the time this bit is set will be "chopped off," and data written to the Transmit Buffer while this bit is set will be lost.

When the transmit underrun occurs and the CRC and closing flag have been sent, bit D_3 can be set to pull TxD High. When ready to start sending data again this bit must be reset to 0 before the first character is written to the Transmit Buffer. Note that resetting this bit causes the TxD pin to take whatever state the NRZI encoder is in at the time, so synchronization at the receiver may take longer because the first transition seen on the TxD pin may not coincide with a bit boundary. Note that in order

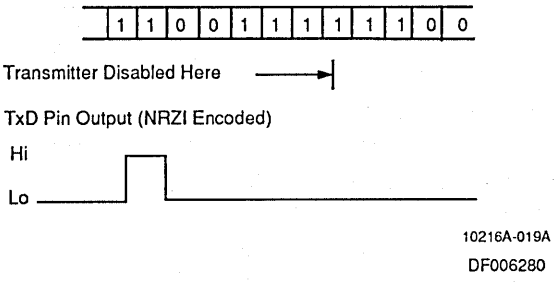


Figure 18. Transmitter Disabling with NRZI Encoding

for this to function properly, bits D_3 and D_2 of WR10 must be set to 1 and 0, respectively.

Interrupt Masking Without $\overline{\text{INTACK}}$

The NMOS Z8530H's ability to mask lower priority interrupts is done via the IUS bit. This bit is internal to the SCC and is not observable by the processor. Being able to automatically mask lower priority interrupts allows a modular approach to coding interrupt routines. However, using the masking capabilities of the NMOS SCC requires that the $\overline{\text{INTACK}}$ cycle be generated. In standalone applications, having to generate $\overline{\text{INTACK}}$ through external hardware in order to use this capability is an unnecessary expense.

On the CMOS Z85C30, if bit D_5 in WR9 is set to 1, the $\overline{\text{INTACK}}$ cycle does not need to be generated in order to have the IUS bit set. This allows the user to respond to ESCC interrupt requests with a software acknowledgment through RR2. When bit D_5 in WR9 is set and an interrupt occurs, a read to RR2 emulates a hardware Interrupt Acknowledge cycle as it functions in Vectored mode. In this case the CPU must first read RR2 to determine the internal interrupt source and then jump to the appropriate interrupt routine. Reading RR2 sets the IUS bit for the highest priority IP. After the interrupting condition is cleared, the routine can then read RR3 to determine if any other IPs are set and clear them. At the end of the interrupt routine, a Reset IUS command must be issued to unlock the internal daisy chain.

Since the CPU can acknowledge the ESCC of highest priority with a read of its RR2 interrupt vector, there is no need for an external daisy chain. IEI for all ESCC devices should be tied active High. When acknowledging an ESCC interrupt request, the CPU must issue one read to RR2 per interrupt request. The modified interrupt vector can be read from Channel B, or the original vector stored in WR2 can be read from Channel A. Either action will produce the same internal actions on the IUS logic. Note that the No Vector and Vector Includes Status bits in WR9 are ignored when bit D_5 in WR9 is set to 1.

2-Mb/s FM Data Transmission and Reception

The 16-MHz version of the CMOS Z85C30 (Z85C30-16) is capable of transmitting and receiving FM-encoded data at the rate of 2 Mb/s. This is accomplished by applying a 32-MHz clock to the $\overline{\text{RTxC}}$ pin and assigning this waveform to drive the Internal Digital Phase-Locked Loop (DPLL) clock. This feature allows the user to send both clock and data information over the same line at 2 Mb/s and can eliminate external DPLLs required for high-speed NRZ data clock generation.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	−65 to +150°C
Voltage at any Pin Relative to V_{SS}	−0.5 to +7.0 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices	
Ambient Temperature (T_A)	0 to +70°C
Supply Voltage (V_{CC})	+5 V ± 10%
Industrial (I) Devices	
Ambient Temperature (T_A)	−40 to +85°C
Supply Voltage (V_{CC})	5 V ±10%
Military (M) Devices	
Case Temperature (T_C)	−55° to 125°
Supply Voltage (V_{CC})	5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IH}	Input High Voltage	Commercial	2.2	V_{CC} +0.3*	V
V_{IL}	Input Low Voltage		−0.3*	0.8	V
V_{OH1}	Output High Voltage	I_{OH} = −1.6 mA	2.4		V
V_{OH2}	Output High Voltage	I_{OH} = −250 µA	V_{CC} −0.8		V
V_{OL}	Output Low Voltage	I_{OL} = +2.0 mA		0.4	V
I_{IL}	Input Leakage	0.4 V ≤ V_{IN} ≤ 2.4 V		±10.0	µA
I_{OL}	Output Leakage	0.4 V ≤ V_{OUT} ≤ 2.4 V		±10.0	µA
I_{CC1}	V_{CC} Supply Current	8.192 MHz		18	mA
		10 MHz Inputs at		18	mA
		12 MHz voltage rails,		22	mA
		16.384 MHz output unloaded		22	mA
		20 MHz		30	mA
C_{IN}	Input Capacitance	Unmeasured pins returned		10	pF
C_{OUT}	Output Capacitance	to ground ! = 1 MHz over		15	pF
C_{MO}	Bidirectional Capacitance	specified temperature range		20	pF

* V_{IH} Max. and V_{IL} Min. not tested. Guaranteed by design.

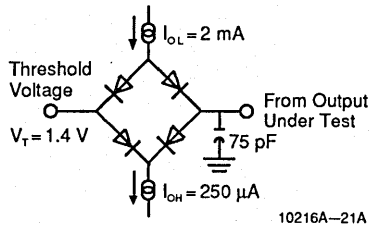
Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

+4.5 V ≤ V_{CC} ≤ +5.5 V
GND = 0 V
0°C ≤ T_A ≤ 70°C

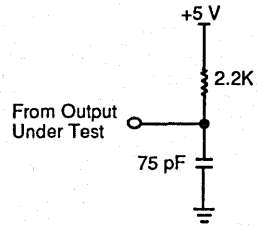
SWITCHING TEST CIRCUITS

Standard Test Dynamic Load Circuit



10216A–21A
TC004800

Open-Drain Test Load



TC001833

SWITCHING CHARACTERISTICS over COMMERCIAL operating range

General Timing (see Figure 19)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	12.5 MHz Min.	12.5 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to $\overline{W/REQ}$ Valid Delay		250		150		120	ns
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		250		220	ns
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8)	NA	NA	NA	NA	NA	NA	
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1)	0		0		0		ns
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1)	150		125		100		ns
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		0		ns
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5)	150		125		100		ns
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1)	−200		−150		−125		ns
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1)	5TcPC		5TcPC		5TcPC		ns
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8)	NA		NA		NA		
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2)		200		150		130	ns
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5)		200		150		130	ns
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		200		140		120	ns
14a	TwRTXh	$\overline{RTxC}$ High Width (Note 6)	150		120		100		ns
14b	TwRTxh(E)	$\overline{RTxC}$ High Width (Note 9)	50		40		34		ns
15a	TwRTXl	$\overline{RTxC}$ Low Width (Note 6)	150		120		100		ns
15b	TwRTXl(E)	$\overline{RTxC}$ Low Width (Note 9)	50		40		34		ns
16a	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	488		400		320		ns
16b	TcRTx(E)	$\overline{RTxC}$ Cycle Time (Note 9)	125		100		80		ns
17	TcRTXX	Crystal Oscillator Period (Note 3)	125	1000	100	1000	80	1000	ns
18	TwTRXh	$\overline{TRxC}$ High Width (Note 6)	150		120		100		ns
19	TwTRXl	$\overline{TRxC}$ Low Width (Note 6)	150		120		100		ns
20	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	488		400		320		ns
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	200		120		100		ns
22	TwSY	$\overline{SYNC}$ Pulse Width	200		120		100		ns

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 ns. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is the worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

General Timing (see Figure 19)

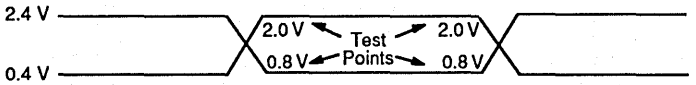
No.	Parameter Symbol	Parameter Description	16.384 MHz Min.	16.384 MHz Max.	20 MHz Min.	20 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to $\overline{W/REQ}$ Valid Delay		80		70	ns
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		180		170	ns
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8)	NA	NA	NA	NA	
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1)	0		0		ns
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1)	50		45		ns
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		ns
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5)	50		45		ns
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1)	−100		−90		ns
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1)	5TcPc		5TcPc		ns
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8)	NA		NA		
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2)		80		70	ns
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5)		80		70	ns
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		80		70	ns
14a	TwRTXh	$\overline{RTxC}$ High Width (Note 6)	80		70		ns
14b	TwRTxh(E)	$\overline{RTxC}$ High Width (Note 9)	15.6		15.6		ns
15a	TwRTXl	$\overline{RTxC}$ Low Width (Note 6)	80		70		ns
15b	TwRTXl(E)	$\overline{RTxC}$ Low Width (Note 9)	15.6		15.6		ns
16a	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	244		200		ns
16b	TcRTx(E)	$\overline{RTxC}$ Cycle Time (Note 9)	31.25		31.25		ns
17	TcRTXX	Crystal Oscillator Period (Note 3)	62	1000	50	1000	ns
18	TwTRXh	$\overline{TRxC}$ High Width (Note 6)	80		70		ns
19	TwTRXl	$\overline{TRxC}$ Low Width (Note 6)	80		70		ns
20	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	244		200		ns
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	70		65		ns
22	TwSY	$\overline{SYNC}$ Pulse Width	70		65		ns

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

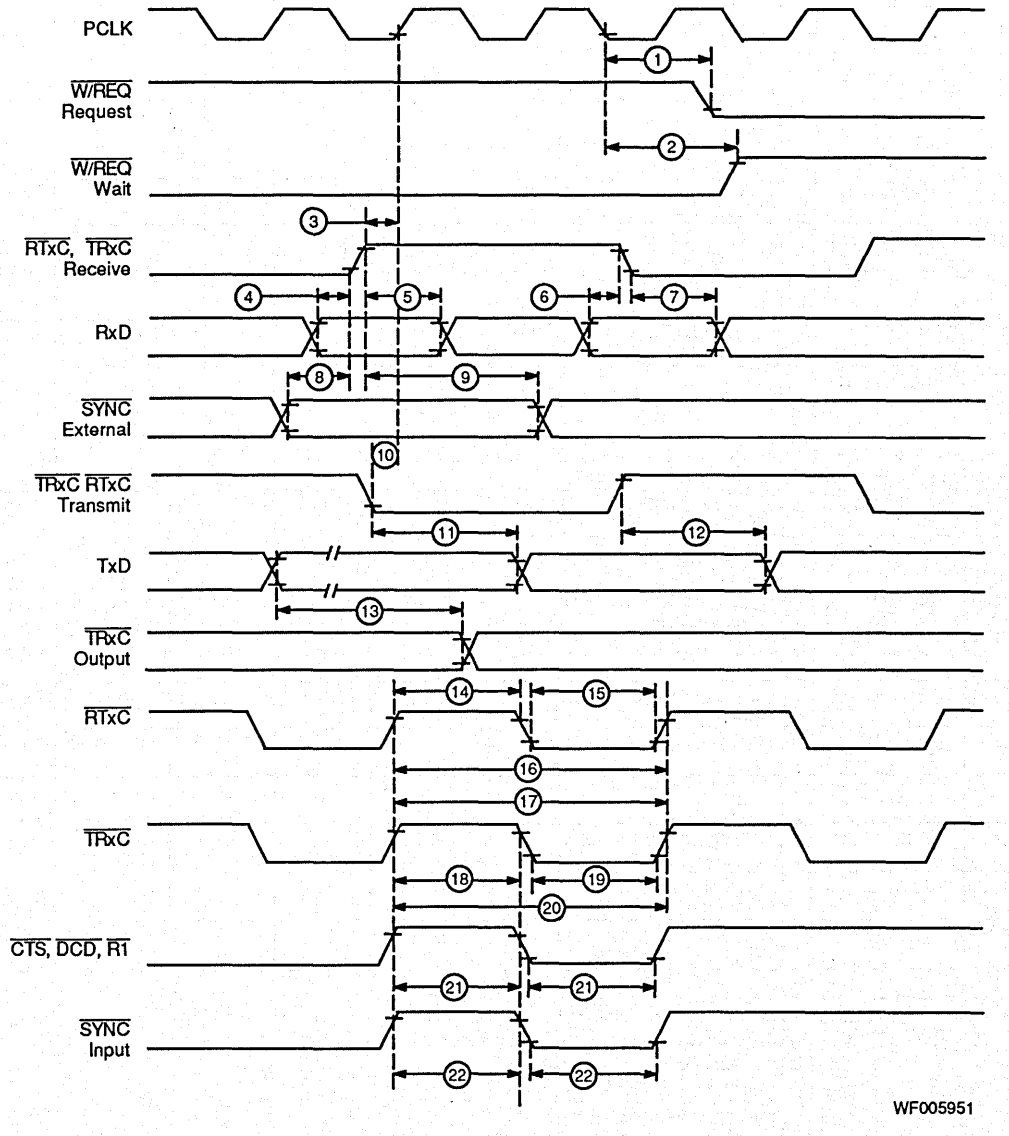
   Tx and Rx input clock slow rates should be kept to a maximum of 30 ns. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is the worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WR006354

AC testing: Inputs are driven at 2.4 V for a logic 1 and 0.4 V for a logic 0.
Timing measurements are made at 2.0 V for a logic 1 and 0.8 V for logic 0.



WF005951

Figure 19. General Timing

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

System Timing (see Figure 20)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	12.5 MHz Min.	12.5 MHz Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	8	12	TcPc
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	8	14	TcPc
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	4	7	TcPc
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	10	16	TcPc
5	TdTXC(REQ)	$\overline{TxC}$ ↑ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	5	8	TcPc
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	5	11	TcPc
7a	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	4	7	TcPc
7b	TdTXC(EDRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4)	5	8	5	8	5	8	TcPc
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	6	10	TcPc
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	2	6	TcPc
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	2	6	TcPc

No.	Parameter Symbol	Parameter Description	16.384 MHz Min.	16.384 MHz Max.	20 MHz Min.	20 MHz Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPc
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPc
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPc
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPc
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPc
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPc
7a	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPc
7b	TdTXC(EDRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4)	5	8	5	8	TcPc
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPc
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
4. Parameter applies to Enhanced Request mode only.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Read and Write Timing (see Figure 21)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	12.5 MHz Min.	12.5 MHz Max.	Unit
1	TwPCl	PCLK Low Width	50	2000	40	2000	34	2000	ns
2	TwPCh	PCLK High Width	50	2000	40	2000	34	2000	ns
3	TfPC	PCLK Fall Time		15		12		10	ns
4	TrPC	PCLK Rise Time		15		12		10	ns
5	TcPC	PCLK Cycle Time	122	4000	100	4000	80	4000	ns
6	TsA(WR)	Address to $\overline{WR}\downarrow$ Setup Time	70		50		45		ns
7	ThA(WR)	Address to $\overline{WR}\uparrow$ Hold Time	0		0		0		ns
8	TsA(RD)	Address to $\overline{RD}\downarrow$ Setup Time	70		50		45		ns
9	ThA(RD)	Address to $\overline{RD}\uparrow$ Hold Time	0		0		0		ns
10	TsIA(PC)	$\overline{INTACK}$ to PCLK $\uparrow$ Setup Time	20		20		15		ns
11	TsIA(WR)	$\overline{INTACK}$ to $\overline{WR}\downarrow$ Setup Time (Note 1)	145		120		95		ns
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}\uparrow$ Hold Time	0		0		0		ns
13	TsIA(RD)	$\overline{INTACK}$ to $\overline{RD}\downarrow$ Setup Time (Note 1)	145		120		95		ns
14	ThIAi(RD)	$\overline{INTACK}$ to $\overline{RD}\uparrow$ Hold Time	0		0		0		ns
15	ThIA(PC)	$\overline{INTACK}$ to PCLK $\uparrow$ Hold Time	40		30		20		ns
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}\downarrow$ Setup Time	0		0		0		ns
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}\uparrow$ Hold Time	0		0		0		ns
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}\downarrow$ Setup Time	60		50		40		ns
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}\downarrow$ Setup Time (Note 1)	0		0		0		ns
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}\uparrow$ Hold Time (Note1)	0		0		0		ns
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}\downarrow$ Setup Time (Note 1)	60		50		40		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	150		125		90		ns
23	TdRD(DRA)	$\overline{RD}\downarrow$ to Read Data Active Delay	0		0		0		ns
24	TdRDr(DR)	$\overline{RD}\uparrow$ to Read Data Not Valid Delay	0		0		0		ns
25	TdRDf(DR)	$\overline{RD}\downarrow$ to Read Data Valid Delay		140		120		85	ns
26	TdRD(DRz)	$\overline{RD}\uparrow$ to Read Data Float Delay (Note 2)		40		35		25	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Read and Write Timing (see Figure 21)

No.	Parameter Symbol	Parameter Description	16.384 MHz		20 MHz		Unit
			Min.	Max.	Min.	Max.	
1	TwPCl	PCLK Low Width	26	2000	22	1000	ns
2	TwPCh	PCLK High Width	26	2000	22	1000	ns
3	TfPC	PCLK Fall Time		8		5	ns
4	TrPC	PCLK Rise Time		8		5	ns
5	TcPC	PCLK Cycle Time	61	4000	50	2000	ns
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	35		30		ns
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		ns
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	35		30		ns
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		ns
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	15		15		ns
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	70		65		ns
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	70		65		ns
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		ns
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	15		15		ns
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		ns
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	30		25		ns
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		ns
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note1)	0		0		ns
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time (Note 1)	30		25		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	75		65		ns
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		ns
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		ns
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		70		60	ns
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		20		20	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.

2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

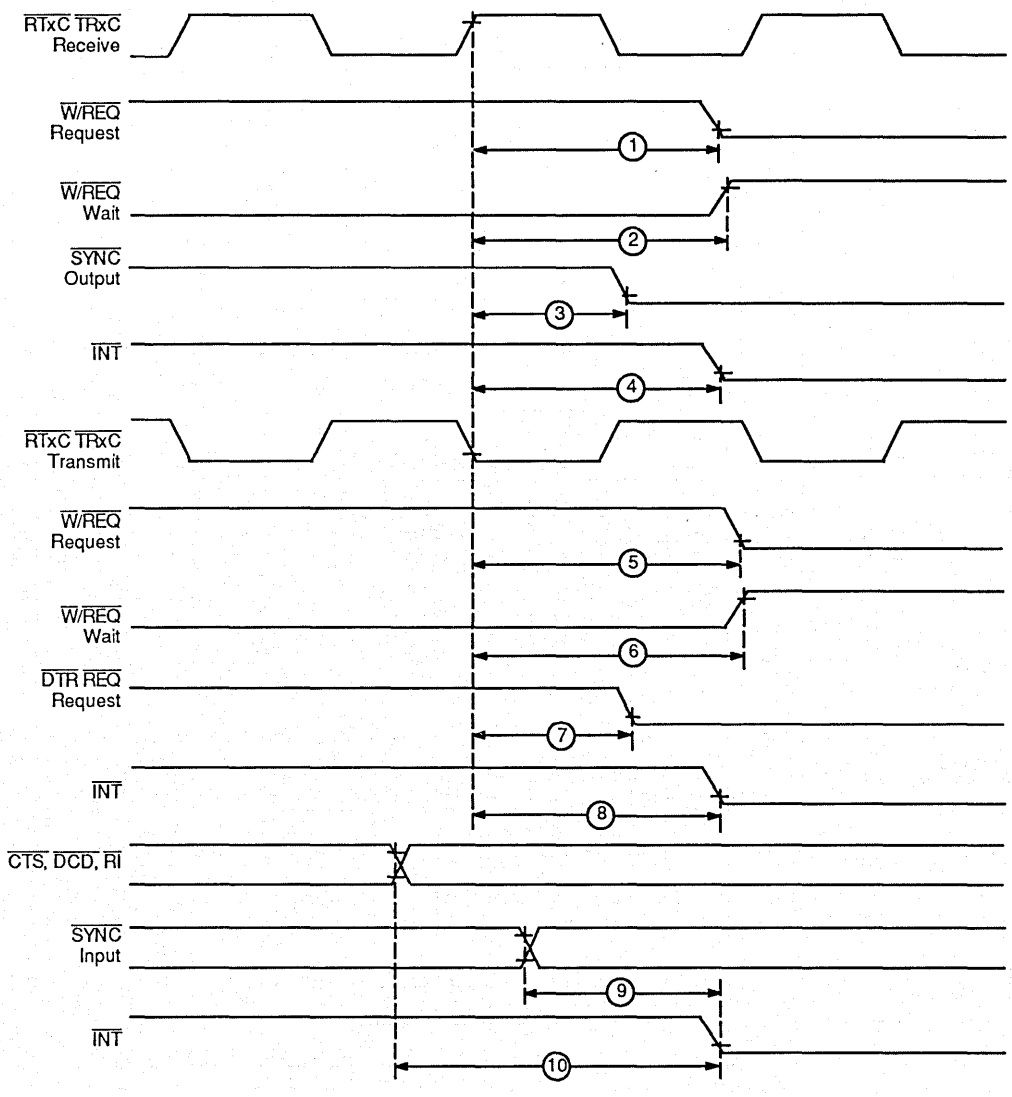


Figure 20. System Timing

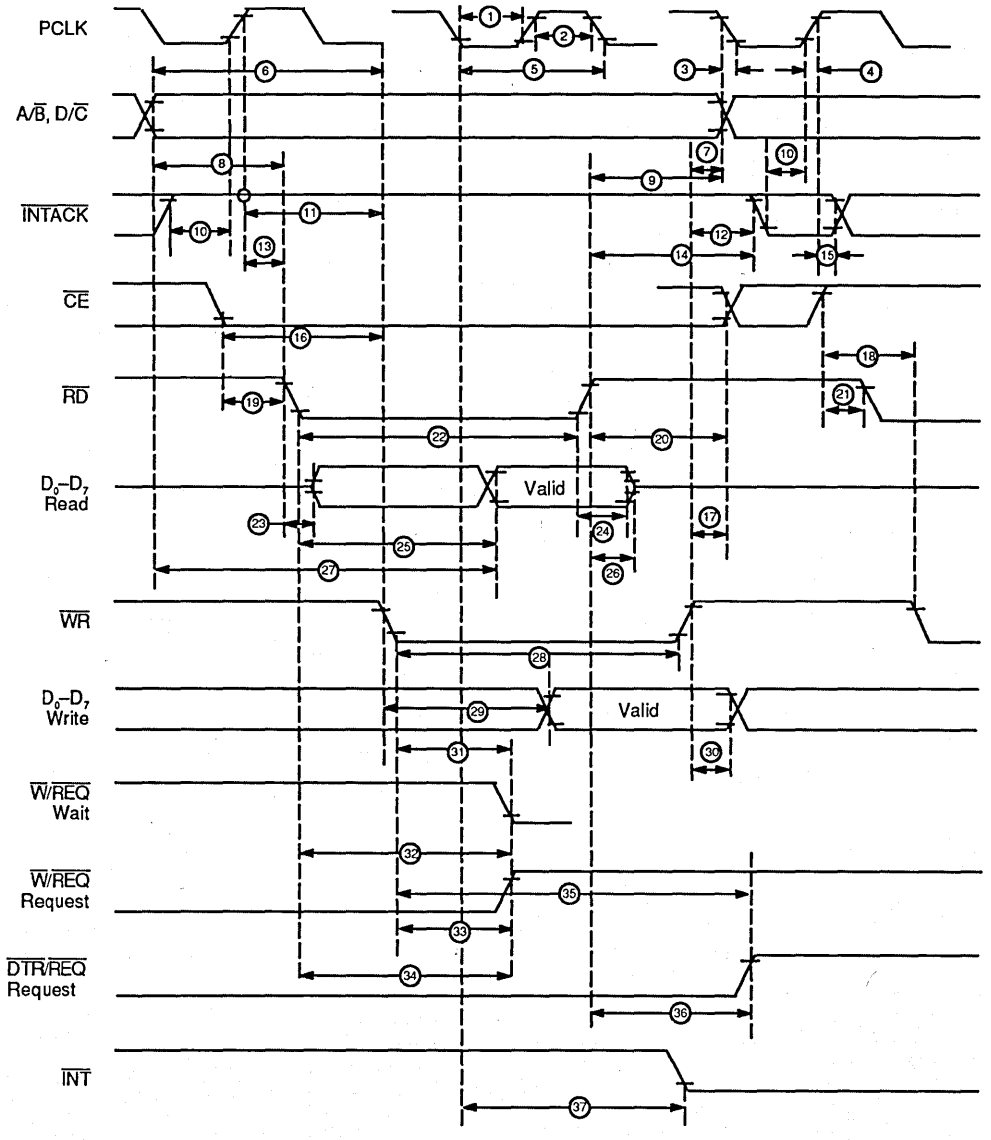


Figure 21. Read and Write Timing

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)

No.	Parameter Symbol	Parameter Description	8.192 MHz		10 MHz		12.5 MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		220		160		120	ns
28	TwWRI	$\overline{WR}$ Low Width	150		125		90		ns
29	TdWRf(DW)	$\overline{WR}$ ↓ to Write Data Valid		35		35		25	ns
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		0		ns
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 2)		170		100		70	ns
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 2)		170		100		70	ns
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		170		120		100	ns
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		170		120		100	ns
35a	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4.0TcPc		4.0TcPc		4.0TcPc	ns
35b	TdWRr(EREQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		120		120		100	ns
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		NA		NA		NA	ns
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2)		500		400		350	ns
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3)	150		125		95		ns
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	150		125		95		ns
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		140		120		90	ns
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	95		80		65		ns
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		0		ns
43	TdIEI(IEO)	IEI to IEO Delay Time		95		80		65	ns
44	TdPC(IEO)	PCLK ↑ to IEO Delay		200		175		130	ns
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2)		450		320		260	ns
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	15		15		10		ns
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	15		15		10		ns
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	150		100		85		ns
49	Trc	Valid Access Recovery Time (Note 1)	3.5		3.5		3.5		TcPc

Notes:
1. Parameter applies only between transactions involving the ESCC, if WR/RD falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of DdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)

No.	Parameter Symbol	Parameter Description	16.384 MHz Min.	16.384 MHz Max.	20 MHz Min.	20 MHz Max.	Unit
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		100		90	ns
28	TwWRI	$\overline{WR}$ Low Width	75		65		ns
29	TdWRf(DW)	$\overline{WR}$ ↓ to Write Data Valid		20		20	ns
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		ns
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 2)		50		45	ns
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 2)		50		45	ns
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		70		65	ns
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		70		65	ns
35a	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4.0TcPc		4.0TcPc	ns
35b	TdWRr(EREQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4)		70		65	ns
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		NA		NA	ns
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2)		175		160	ns
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3)	50		45		ns
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	75		65		ns
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		70		60	ns
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	50		45		ns
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		ns
43	TdIEI(IEO)	IEI to IEO Delay Time		45		40	ns
44	TdPC(IEO)	PCLK ↑ to IEO Delay		80		70	ns
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2)		200		180	ns
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	10		10		ns
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	10		10		ns
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	75		65		ns
49	Trc	Valid Access Recovery Time (Note 1)	3.5		3.5		TcPc

Notes:
1. Parameter applies only between transactions involving the ESCC. If $\overline{WR/RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

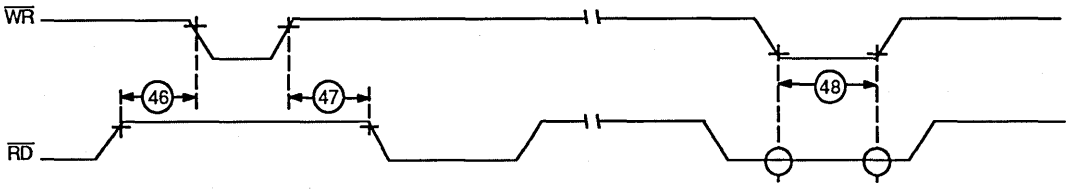


Figure 22. Reset Timing

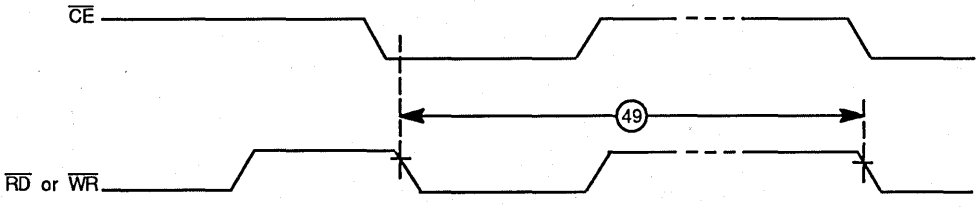


Figure 23. Cycle Timing

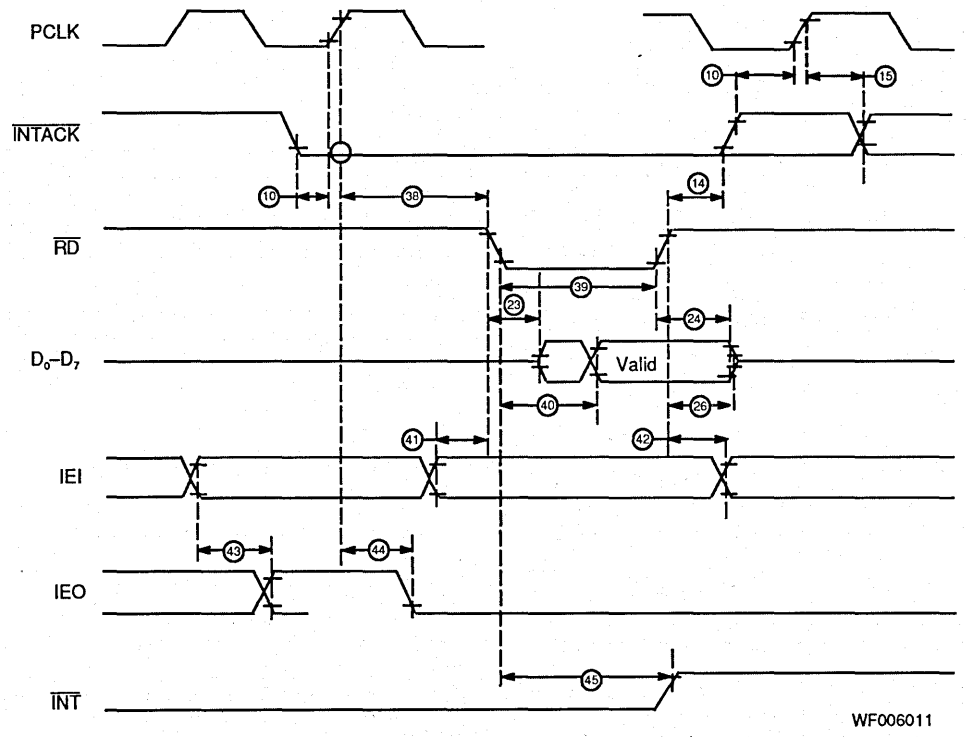


Figure 24. Interrupt Acknowledge Timing

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range

General Timing (see Figure 19)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to W/REQ Valid Delay		250		150	ns
2	TdPC(W)	PLCK ↓ to Wait Inactive Delay		350		250	ns
3	TsRXC(PC)	RxC ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8)	NA	NA	NA	NA	
4	TsRXD(RXCr)	RxD to RxC ↑ Setup Time (XI Mode) (Note 1)	0		0		ns
5	ThRXD(RXCr)	RxD to RxC ↑ Hold Time (XI Mode) (Note 1)	150		125		ns
6	TsRXD(RXCf)	RxD to RxC ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		ns
7	ThRXD(RXCf)	RxD to RxC ↓ Hold Time (XI Mode) (Notes 1, 5)	150		125		ns
8	TsSY(RXC)	SYNC to RxC ↑ Setup Time (Note 1)	–200		–150		ns
9	ThSY(RXC)	SYNC to RxC ↑ Hold Time (Note 1)	5TcPc		5TcPc		ns
10	TsTXC(PC)	TxC ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8)	NA		NA		
11	TdTXCf(TXD)	TxC ↓ to TxD Delay (XI Mode) (Note 2)		200		150	ns
12	TdTXCr(TXD)	TxC ↑ to TxD Delay (XI Mode) (Notes 2, 5)		200		150	ns
13	TdTXD(TRX)	TxD to TRxC Delay (Send Clock Echo)		200		140	ns
14a	TwRTXh	RTxC High Width (Note 6)	150		120		ns
14b	TwRTXh(E)	RTxC High Width (Note 9)	50		40		ns
15a	TwRTXl	RTxC Low Width (Note 6)	150		120		ns
15b	TwRTXl(E)	RTxC Low Width (Note 9)	50		40		ns
16a	TcRTX	RTxC Cycle Time (Notes 6, 7)	488		400		ns
16b	TcRTx(E)	RTxC Cycle Time (Note 9)	125		100		ns
17	TXRTXX	Crystal Oscillator Period (Note 3)	125	1000	100	1000	ns
18	TwTRXh	TRxC High Width (Note 6)	150		120		ns
19	TwTRXl	TRxC Low Width (Note 6)	150		120		ns
20	TcTRX	TRxC Cycle Time (Notes 6, 7)	488		400		ns
21	TwEXT	DCD or CTS Pulse Width	200		120		ns
22	TwSY	SYNC Pulse Width	200		120		ns

Notes:
1. RxC is RTxC or TRxC, whichever is supplying the receive clock.
2. TxC is TRxC or RTxC, whichever is supplying the transmit clock.
3. Both RTxC and SYNC have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between RxC and PCLK or TxC and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to RxC or TxC synchronization requirement eliminated for PCLK divide-by-four operation.

   TRxC and RTxC rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 ns. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is the worst case.
9. ENHANCED FEATURE— RTxC used as input to internal DPLL only.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

General Timing (see Figure 19)

No.	Parameter Symbol	Parameter Description	12.5 MHz Min.	12.5 MHz Max.	16.384 MHz Min.	16.384 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to W/$\overline{REQ}$ Valid Delay		120		80	ns
2	TdPC(W)	PLCK ↓ to Wait Inactive Delay		220		180	ns
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8)	NA	NA	NA	NA	
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1)	0		0		ns
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1)	100		50		ns
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		ns
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5)	100		50		ns
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1)	–125		–100		ns
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1)	5TcPc		5TcPc		ns
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8)	NA		NA		
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2)		130		80	ns
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5)		130		80	ns
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		120		80	ns
14a	TwRTXh	$\overline{RTxC}$ High Width (Note 6)	100		80		ns
14b	TwRTXh(E)	$\overline{RTxC}$ High Width (Note 9)	34		15.6		ns
15a	TwRTXl	$\overline{RTxC}$ Low Width (Note 6)	100		80		ns
15b	TwRTXl(E)	$\overline{RTxC}$ Low Width (Note 9)	34		15.6		ns
16a	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	320		244		ns
16b	TcRTx(E)	$\overline{RTxC}$ Cycle Time (Note 9)	80		31.25		ns
17	TXRTXX	Crystal Oscillator Period (Note 3)	80	1000	62	1000	ns
18	TwTRxh	$\overline{TRxC}$ High Width (Note 6)	100		80		ns
19	TwTRXl	$\overline{TRxC}$ Low Width (Note 6)	100		80		ns
20	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	320		244		ns
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	100		70		ns
22	TwSY	$\overline{SYNC}$ Pulse Width	100		70		ns

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 ns. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is the worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

System Timing (see Figure 20)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPc
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPc
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPc
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPc
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPc
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPc
7a	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPc
7b	TdTXC(EDRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4)	5	8	5	8	TcPc
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPc
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc

No.	Parameter Symbol	Parameter Description	12.5 MHz Min.	12.5 MHz Max.	16.384 MHz Min.	16.384 MHz Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPc
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPc
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPc
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPc
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPc
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPc
7a	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPc
7b	TdTXC(EDRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4)	5	8	5	8	TcPc
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPc
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
4. Parameter applies to Enhanced Request mode only.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

Read and Write Timing (see Figure 21)

No.	Parameter Symbol	Parameter Description	8.192 MHz		10 MHz		Unit
			Min.	Max.	Min.	Max.	
1	TwPCl	PCLK Low Width	50	1000	40	1000	ns
2	TwPCh	PCLK High Width	50	1000	40	1000	ns
3	TfPC	PCLK Fall Time		15		12	ns
4	TrPC	PCLK Rise Time		15		12	ns
5	TcPC	PCLK Cycle Time	122	2000	100	2000	ns
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	70		50		ns
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		ns
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	70		50		ns
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		ns
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	20		20		ns
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	145		120		ns
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	145		120		ns
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		ns
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	40		30		ns
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		ns
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	60		50		ns
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		ns
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		0		ns
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time (Note 1)	60		50		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	150		125		ns
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		ns
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		ns
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		140		125	ns
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		40		35	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

Read and Write Timing (see Figure 21)

No.	Parameter Symbol	Parameter Description	12.5 MHz		16.384 MHz		Unit
			Min.	Max.	Min.	Max.	
1	TwPCl	PCLK Low Width	34	1000	26	1000	ns
2	TwPCh	PCLK High Width	34	1000	26	1000	ns
3	TfPC	PCLK Fall Time		10		8	ns
4	TrPC	PCLK Rise Time		10		8	ns
5	TcPC	PCLK Cycle Time	80	2000	61	2000	ns
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	45		35		ns
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		ns
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	45		35		ns
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		ns
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time.	15		15		ns
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	95		70		ns
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time	95		70		ns
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		ns
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	20		15		ns
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		ns
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	40		30		ns
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		ns
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		0		ns
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time	40		30		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	90		75		ns
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		ns
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		ns
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		90		70	ns
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		25		20	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.

2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	Unit
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		220		160	ns
28	TwWRI	$\overline{WR}$ Low Width	150		125		ns
29	TdWRf(DW)	$\overline{WR}$ ↓ to Write Data Valid		35		35	ns
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		ns
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 2)		170		100	ns
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 2)		170		100	ns
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		170		120	ns
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		170		120	ns
35a	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4.0TcPc		4.0TcPc	ns
35b	TdWRr(EREQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4)		120		120	ns
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		NA		NA	ns
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2)		500		400	ns
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3)	150		125		ns
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	150		125		ns
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		140		120	ns
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	95		80		ns
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		ns
43	TdIEI(IEO)	IEI to IEO Delay Time		95		80	ns
44	TdPC(IEO)	PCLK ↑ to IEO Delay		200		175	ns
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2)		450		320	ns
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	15		15		ns
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	15		15		ns
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	150		100		ns
49	Trc	Valid Access Recovery Time (Note 1)	3.5		3.5		ns

Notes:
1. Parameter applies only between transactions involving the ESCC. If $\overline{WR/RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)

No.	Parameter Symbol	Parameter Description	12.5 MHz		16.384 MHz		Unit
			Min.	Max.	Min.	Max.	
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		120		100	ns
28	TwWRI	$\overline{WR}$ Low Width	90		75		ns
29	TdWRf(DW)	$\overline{WR}$ ↓ to Write Data Valid		25		20	ns
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		ns
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 2)		70		50	ns
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 2)		70		50	ns
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		100		70	ns
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		100		70	ns
35a	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4.0TcPc		4.0TcPc	ns
35b	TdWRr(EREQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4)		100		70	ns
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		NA		NA	ns
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2)		350		175	ns
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3)	95		50		ns
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	95		75		ns
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		90		70	ns
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	65		50		ns
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		ns
43	TdIEI(IEO)	IEI to IEO Delay Time		65		45	ns
44	TdPC(IEO)	PCLK ↑ to IEO Delay		130		80	ns
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2)		260		200	ns
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	10		10		ns
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	10		10		ns
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	85		75		ns
49	Trc	Valid Access Recovery Time (Note 1)	3.5		3.5		TcPc

Notes:
1. Parameter applies only between transactions involving the ESCC. If $\overline{WR/RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

Interfacing the Z8530H and Z85C30 Serial Communications Controllers to the 80186 Microprocessor

by John Langlois

INTRODUCTION

A simple interface between the 80186 Microprocessor and Z8530 Serial Communications Controller (SCC) can now be realized, due to the enhancements achieved with the Z8530H and Z85C30 Serial Communications Controllers. Previous application notes have required eight 74LS devices to meet the Z8530 timing requirements. This document describes solutions that can be implemented in a single chip. Depending on system constraints, the interface can be as simple as a single inverter (Z85C30 in enhanced mode), but is never more than a PAL16L8 (Z8530H) device.

The timing requirements of each version of the Z8530 are examined in the next section. The original Z8530 is discussed only to provide a reference point. Solutions for each interface problem are then presented. If a particular feature is not required, that circuit obviously need not be implemented. Timing analysis for the complete interfaces follow the solutions. PAL® equations are also included.

For the purposes of this document, it is assumed that the reader is familiar with the Z8530 SCC and 80186 microprocessor. The following data sheets are applicable:

AMD 80186 High Integration 16-Bit Microprocessor (Order #03551)
AMD Z85C30 Enhanced Serial Communications Controller (Order #10216)
AMD Z8530H Serial Communications Controller (Order #00970)
AMD Z85C30 Technical Manual (Order #07513)

TIMING REQUIREMENTS

Z8530 SCC

The original Z8530/80186 interface is complicated by several factors, the most difficult being the access recovery time (T_{rc}). The minimum time from the trailing edge of one command to the leading edge of the next is six clock cycles + 130 ns for the 6-MHz part. This necessitates the use of extensive wait state generation circuitry to hold off back-to-back accesses. It is possible to meet the requirement in software by inserting NOPs. This practice is generally frowned upon, since the processor clock speed would affect the software. Also, DMA transfers at high rates have no way of inserting NOPs. The solution to this has already been addressed and is not part of this discussion (EDN, April 4, 1985, pp. 274–275).

Interrupt acknowledge cycles with the Z8530 don't match the 80186. While the 80186 generates two pulses on the INTA line, reading the vector on the second pulse, the Z8530 expects to see one long INTA pulse, with a read strobe near the end to read the interrupt vector. Note that it is possible to read the interrupt vector from Read Register 2 of the Z8530. This will not, however, set the IUS bit and mask off lower priority interrupts; a hardware cycle must occur to accomplish that.

Data is expected to be valid at the leading edge of write during a write cycle. Consequently, write must be delayed to the Z8530 until the next rising edge of the processor clock. This may or may not be a problem, depending on how the wait state circuitry is implemented.

The read strobe must also be delayed by one-half clock to meet the address setup time requirements.

Direct Memory Access (DMA) transfers are supported by the Z8530. The $\overline{W/REQ}$ pin may be programmed as a DMA request for either transmit or receive. $\overline{DTR/REQ}$ can be used as a DMA request for transmit only. It is possible to support single-channel full-duplex or dual-channel half-duplex DMA with the DMA controller resident on the 80186. While the $\overline{W/REQ}$ timing does not present a problem, it is always possible to generate erroneous requests on the $\overline{DTR/REQ}$ interface. This is due to $\overline{DTR/REQ}$ going to its inactive state timed from the trailing edge of the command, which is too late to prevent an additional DMA request from being recognized by the 80186. $\overline{DTR/REQ}$ must therefore be negated at the beginning of the cycle servicing the request.

A reset of the Z8530 is accomplished by asserting read and write simultaneously. A software reset command can also be issued by reading Read Register 0 and writing the reset command to Write Register 9. This is a choice left to the designer.

Z8530H SCC

A major improvement was made in the access recovery time with the Z8530H. T_{rc} is now measured from the leading edge of one command to the leading edge of the next. It is also reduced from six clocks + 130 ns to four clocks. The increased clock speed (8.192 MHz) allows it to be clocked from the 80186 CLKOUT in many systems. This improvement eliminates the wait state generation circuitry, allowing the 80186's internal wait state generator to be used.

The other Z8530 timing requirements are still valid.

Publication # 12482 Rev. A Amendment /0
Issue Date: August 1989

Z85C30 SCC

The Z85C30 provides many additional improvements. With this device, it is possible to interface to the 80186 with three inverters and still have interrupt and DMA support.

Access recovery time is improved to three clock cycles if the commands are synchronized to PCLK. This is the case here, since CLKOUT from the 80186 is driving PCLK. As in the Z8530H, it is measured from leading edge to leading edge of the command.

With the Z85C30, it is not necessary to use hardware interrupt acknowledge sequences. An additional feature has been added that will emulate a hardware cycle when the vector is read from Read Register 2. As with the hardware cycle, the IUS bit is set and all lower priority interrupts within the device are masked. Bit D5 of Write Register 9 controls this enhancement.

Data does not need to be valid prior to the leading edge of write. The timing has been relaxed so that data can become valid a short time after write. To take advantage of this, the Z85C30 data pins must be connected directly to the 80186, as the additional buffer delay would violate this parameter. This is not a problem in most systems. If it is, write must be delayed.

The read strobe need not be delayed, due to improved address setup time parameters.

The last enhancement of the Z85C30 that affects the 80186 interface is $\overline{DTR/REQ}$ timing. The extended Write Register 7, or WR7', is enabled by setting bit D_0 of Write Register 15. Bit D_4 of WR7' then controls whether $\overline{DTR/REQ}$ is de-asserted at the leading edge, like $\overline{W/REQ}$, or the trailing edge, as in the Z8530.

SOLUTIONS

Delayed Write

Write to the Z8530 must be delayed in the case of the Z8530H, and may need to be delayed with the Z85C30. Two signals enter into the delayed write equation: WRITE from the 80186 and CLKOUT. The PAL equation is:

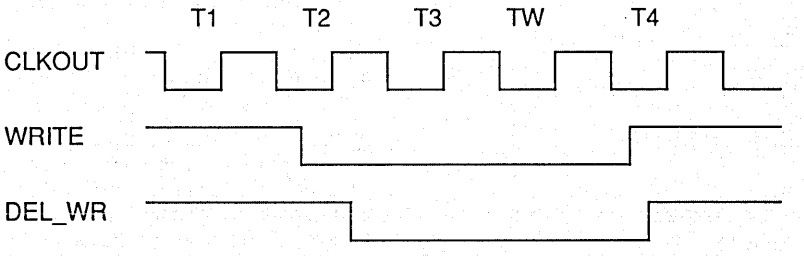
DEL_WR = WRITE * DEL_WR + CLKOUT * WRITE

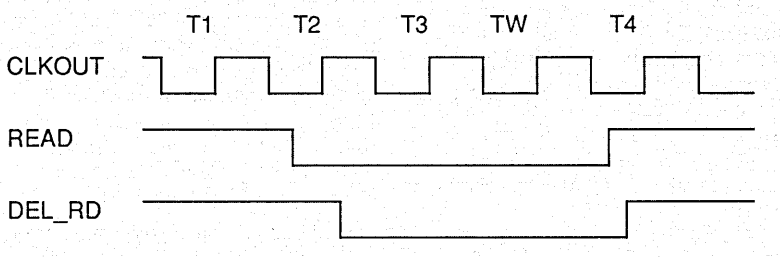
This equation is read as: DEL_WR will be active (low) if WRITE is active (low) and DEL_WR is active (low), or CLKOUT is active (high) and WRITE is active (low).

Delayed Read

The read strobe may need to be delayed to meet address setup requirements. The same equation used for delayed write applies.

DEL_RD = READ * DEL_RD + CLKOUT * READ





Interrupt Acknowledge Generation

This circuit generates the required interrupt acknowledge pulse and read strobe to the Z8530 during an interrupt acknowledge sequence. Inputs are CLKOUT and INTA0, both generated by the 80186. ZINTACK and ZRD are outputs driving INTA and READ of the Z8530, respectively. ZINTACK is simply INTA0 divided by 2, set to its inactive state on reset. This is easily done with a positive edge-triggered flip-flop, but that's not being used. To perform the same function in a combinatorial PAL device, two macrocells are used.

ZRD is merely the logical AND of INTA0 and ZINTACK. ZS0, ZINTACK and ZRD are defined as active low signals. RESET is active high and initializes the two cells.

ZS0 = RESET + (ZS0 * /INTA0 + ZS0 * ZINTACK + INTA0 * ZINTACK)

ZINTACK = /RESET * (/ZS0 * /INTA0 + /ZS0 * ZINTACK + INTA0 * ZINTACK)

ZRD = ZINTACK * INTA0

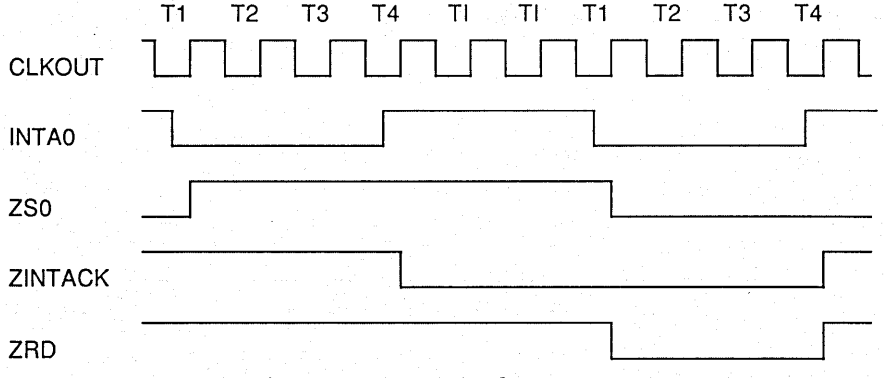
Hardware Reset

A hardware reset is generated by asserting a read and write to the Z8530 simultaneously. RESET is an active high output from the 80186.

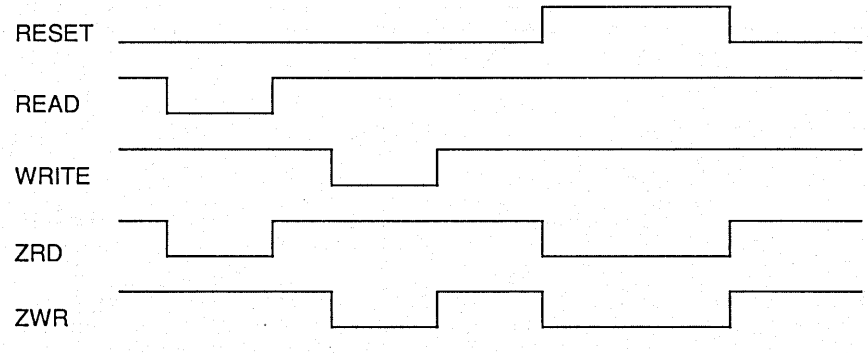
ZRD = READ + RESET
ZWR = WRITE + RESET

$\overline{\text{DTR}}/\overline{\text{REQ}}$ Removal

There are two solutions to the $\overline{\text{DTR}}/\overline{\text{REQ}}$ removal problem. One solution uses a separate Peripheral Chip Select for transfers associated with the request. The other solution requires delayed write and S6 (A19) of the 80186, which indicates whether the current bus cycle is DMA or processor. Write must be delayed, since S6 is not guaranteed valid at its leading edge. Write will be combined with one of these inputs to clear the request. $\overline{\text{DTR}}/\overline{\text{REQ}}$ only supports transmit data, consequently, read is irrelevant.



Interrupt Acknowledge Sequence



Z8530 Hardware Reset

It is not necessary to decode A/$\overline{B}$ to determine which channel of the Z8530 is being accessed. If $\overline{DTR/REQ}$ is used, then that channel is full-duplex, expending both 80186 DMA request inputs; the other channel cannot be supported. If the channel is half-duplex, then $\overline{W/REQ}$ would be utilized because it goes both ways.

The following solution uses delayed write and S6. With a separate chip select, merely substitute it for S6 and use the normal write (not that the delayed write is abnormal).

This circuit could also be implemented using a standard flip-flop, but it can easily be realized with two cells in the PAL device.

Define CLEAR as the logical AND of the Z8530's chip select, delayed write, and S6. Assume PCS0 from the processor is the chip select. DRQ1 is the DMA request to the 80186. DS0 is the output of the second cell. All signals are active low, with the exception of RESET. RESET initializes the two cells to their correct state.

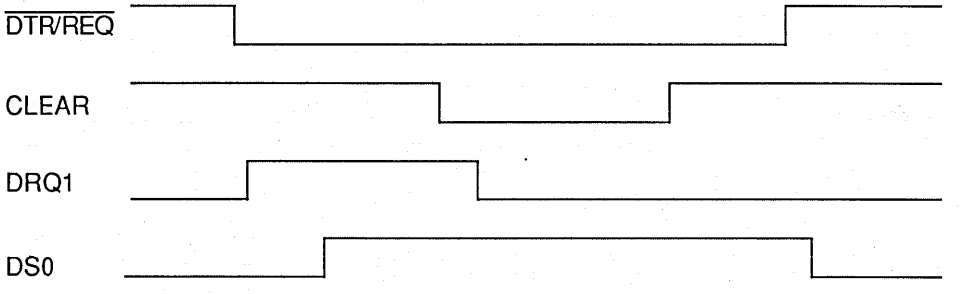
DS0 = DS0 * DRQ1 + /DTR_REQ * DRQ1 + RESET

DRQ1 = /DS0 * DRQ1 + CLEAR * /DS0 + /DS0 * /DTR_REQ + DRQ1 * /DTR_REQ + RESET

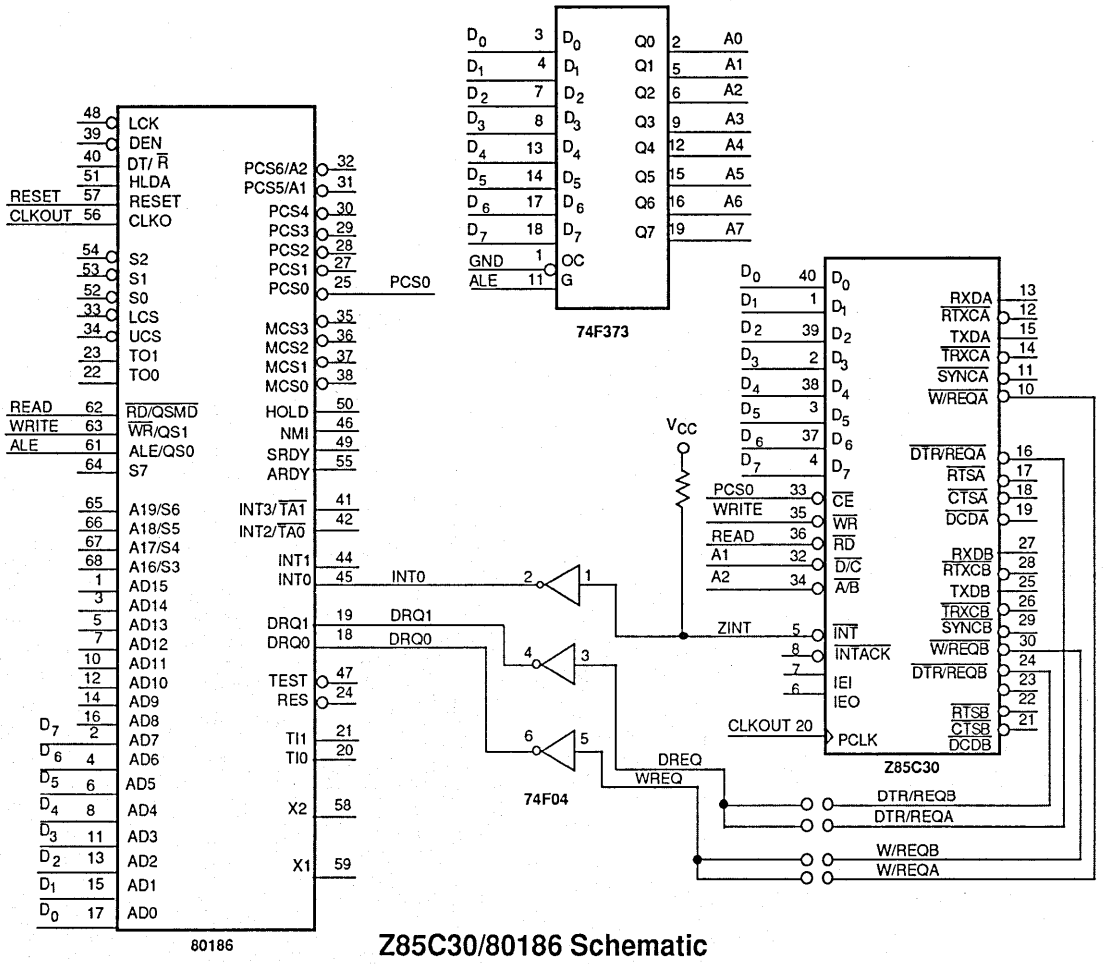
COMPLETE SOLUTIONS

Z85C30 Enhanced Mode

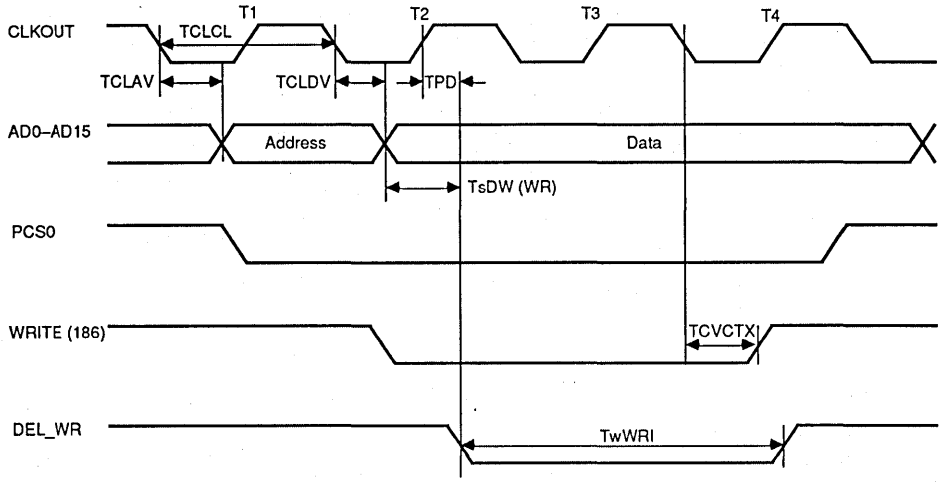
If using the Z85C30 in enhanced mode without hardware interrupt vector transfer, the 80186 interface is trivial. The interrupt and DMA request lines must be inverted, for which 1/2 of a 74F04 is adequate. It is assumed that a data buffer is not placed between the Z85C30 and the 80186, otherwise, WRITE must be delayed. RESET is performed in software and the clock rate is 10 MHz.



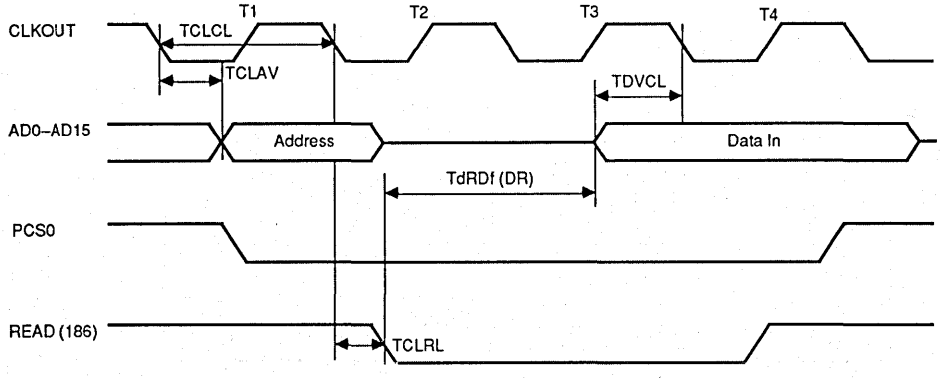
$\overline{DTR/REQ}$ Removal



Z85C30/80186 Schematic



Z8530H Write Cycle Timing Diagram



Z85C30 Read Cycle Timing Diagram

Write Cycle Timing

The Z85C30 required address setup time is 50 ns.

$$T_{sA(WR)} = T_{CLCL} - T_{CLAV} - T_{F373} + T_{CVCTV}$$
$$= 100 - 44 - 8 + 5$$
$$= 53 \text{ ns}$$

Data to the Z85C30 must be valid within 35 ns of WRITE being active.

$$T_{dWRf(DW)} = T_{CLDV(max)} - T_{CVCTV(min)}$$
$$= 40 - 5$$
$$= 35 \text{ ns}$$

WRITE is to be active for a minimum of 125 ns.

$$T_{wWRI} = 2\,T_{CLCL} - T_{CVCTV} + T_{CVCTX}$$
$$= 200 - 56 + 5$$
$$= 149 \text{ ns}$$

The access recovery time of three clocks is easily satisfied.

Read Cycle Timing

Data must be valid at the 80186 15 ns prior to the beginning of state T4. Data vali8d from READ active for the Z85C30 is 120 ns.The Z85C30 required address setup time is 50 ns.

$$T_{DVCL} = T_{CLCL} - T_{CLRL} - T_{dRDf(DR)}$$
$$= 200 - 40 - 120$$
$$= 40 \text{ ns}$$

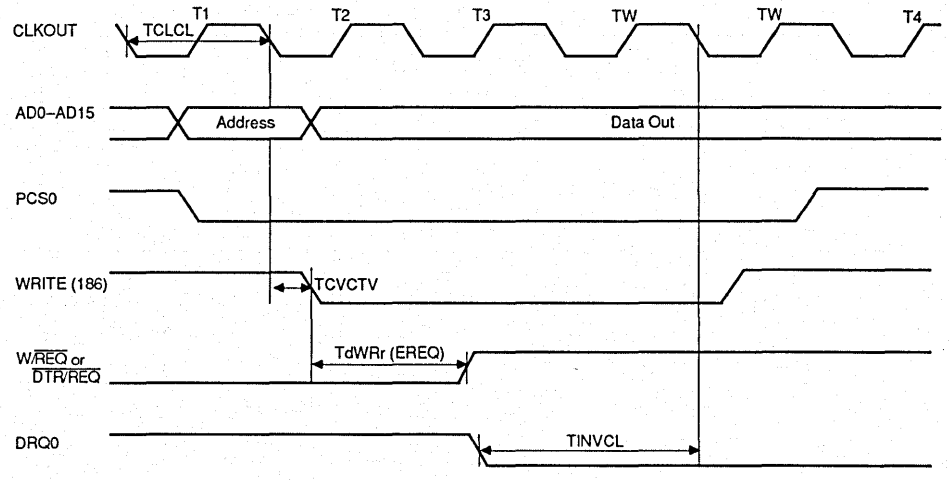
Address setup for READ is the same as WRITE, 50 ns.

$$T_{sA(RD)} = T_{CLCL} - T_{CLAV} - T_{F373} + T_{CLRL}$$
$$= 100 - 44 - 8 + 10$$
$$= 58 \text{ ns}$$

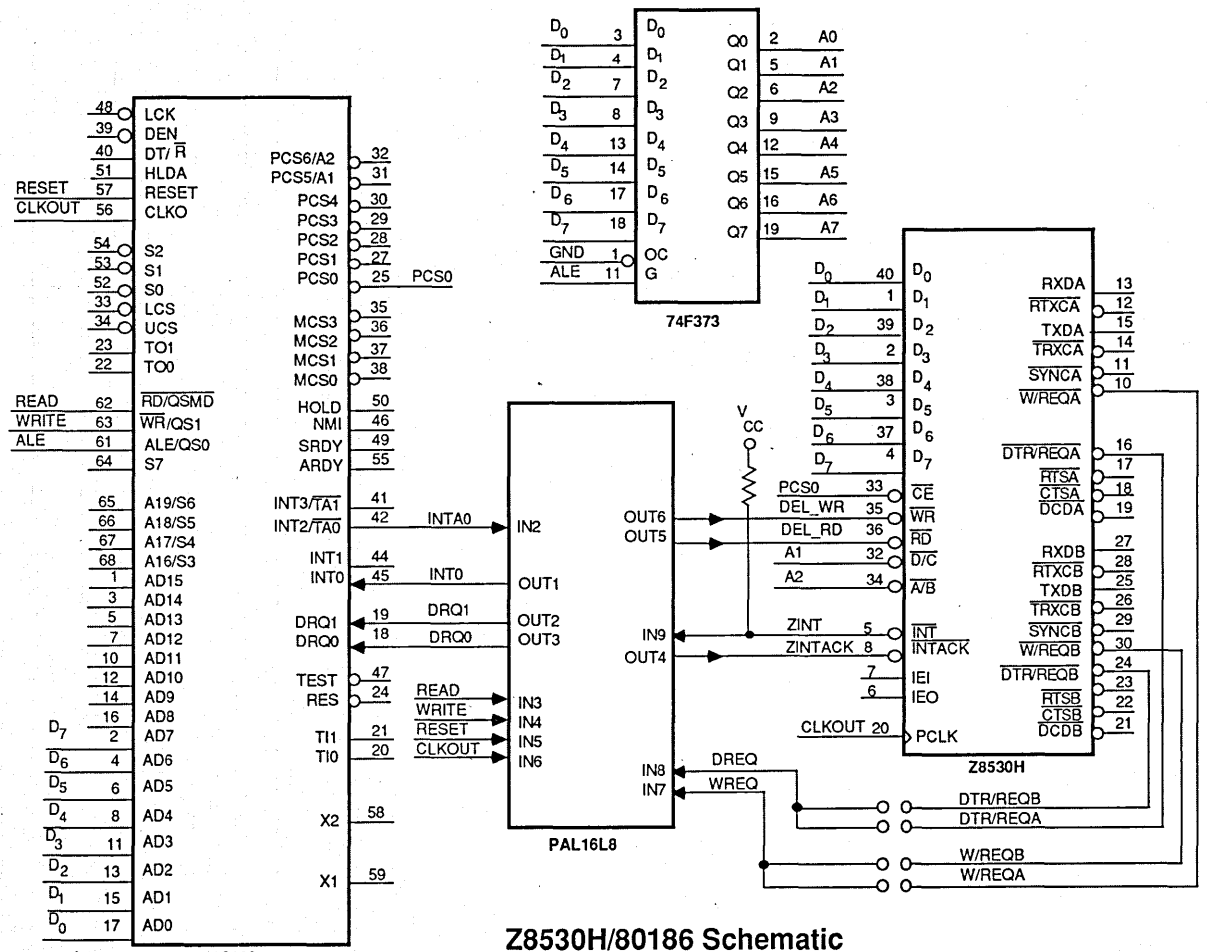
DMA Timing

To prevent a false DMA cycle, the request must be de-asserted two clock periods prior to the end of the current bus cycle. The required setup time for DRQ into the 80186 is 20 ns. With one wait state, the specification is missed by 2 ns, therefore, two wait states are required.

$$T_{INVCL} = 3\,T_{CLCL} - T_{CVCTV} - T_{dWRr(EREQ)} - T_{F04}$$
$$= 300 - 56 - 120 - 6$$
$$= 118 \text{ ns}$$



Z85C30 DMA Cycle Timing Diagram



Z8530H/80186 Schematic

Z8530H

This analysis applies to the 8.192-MHz Z8530H, but the design works with the Z85C30 as well. All of the solutions in the "SOLUTIONS" section are utilized to provide a complete interface. Included are hardware interrupt acknowledge cycles, full DMA support with removal of DTR/REQ, hardware reset, and delayed reads and writes.

Write Cycle Timing

The required address setup time for the 8.192-MHz Z8530H is 70 ns.

$$T_{sA(WR)} = T_{CLCL} + T_{CLCH} - T_{F373} - T_{CLAV}$$
$$= 125 + 56.5 - 8 - 55$$
$$= 118.5 \text{ ns}$$

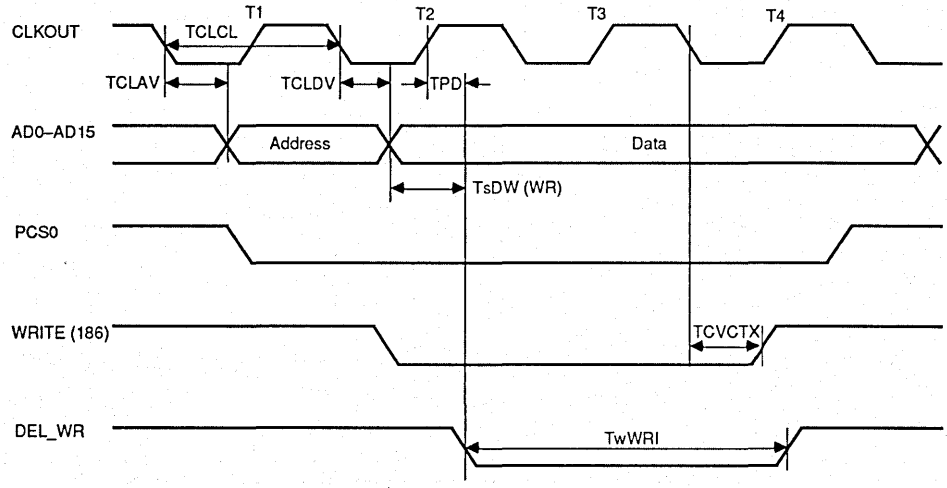
Data has to be valid 10 ns before the falling edge of WRITE to the Z8530H.

$$T_{sDW(W} = T_{CLCH} - T_{CLDV}$$
$$= 56.5 - 44$$
$$= 12.5 \text{ ns}$$

WRITE must be low for 150 ns.

$$T_{wWR} = T_{CLCL} + T_{CLCH} - T_{PD} + T_{CVCTX}$$
$$= 181.5 - 15 + 5$$
$$= 171.5 \text{ ns}$$

Four clock edges occur between leading command edges. This satisfies the access recovery time.



Z8530H Write Cycle Timing Diagram

Read Cycle Timing

Data must be valid at the 80186 20 ns prior to the beginning of state T4. Data valid from READ active for the Z8530H is 140 ns.

$$T_{DVCL} = T_{CLCL} + T_{CLCH} - T_{PD} - T_{dRDf(DR)}$$
$$= 181.5 - 15 - 140$$
$$= 26.5 \text{ ns}$$

The required address setup time for the 8.192-MHz Z8530H is 70 ns.

$$T_{sA(RD} = T_{CLCL} + T_{CLCH} - T_{F373} - T_{CLAV}$$
$$= 181.5 - 8 - 55$$
$$= 118.5 \text{ ns}$$

DMA Timing

To prevent a false DMA cycle, the request must be de-asserted two clock periods prior to the end of the current bus cycle. The required setup time for DRQ into the 80186 is 25 ns.

$\overline{W/REQ}$

$$T_{INVCL} = 2\,T_{CLCL} + T_{CLCH} - T_{PD} - T_{dWR(REQ)} - T_{PD}$$
$$= 306.5 - 15 - 170 - 15$$
$$= 106.5 \text{ ns}$$

$\overline{DTR/REQ}$

$$T_{INVCL} = 2\,T_{CLCL} + T_{CLCH} - 2T_{PD}$$
$$= 306.5 - 30$$
$$= 276.5 \text{ ns}$$

Thus, two wait states are necessary to meet the $\overline{W/REQ}$ timing.

Interrupt Acknowledge Cycle Timing

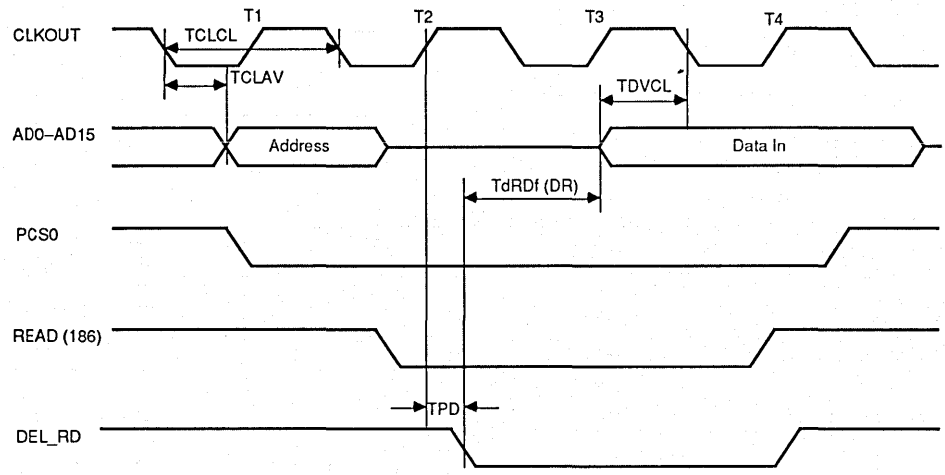
Interrupt acknowledge to the Z8530H is to be valid 150 ns prior to READ becoming active. ZINTACK is guaranteed to be recognized by the Z8530H at the middle of the first idle state between the interrupt acknowledge pulses.

$$T_{dIAi(RD)} = T_{CLCL} + T_{CLCH} + T_{CVCTV}$$
$$= 181.5 + 10$$
$$= 191.5 \text{ ns}$$

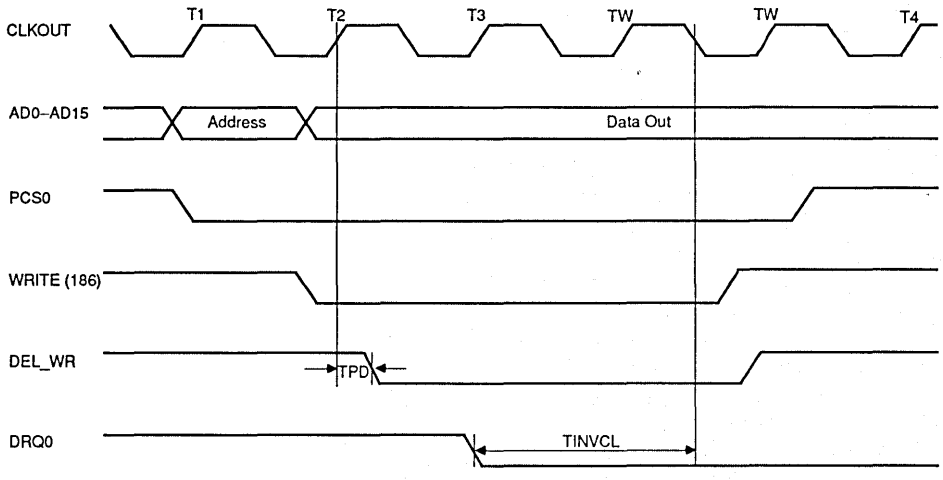
There is not enough delay between ZINTACK and DEL_RD to allow daisy- chained interrupts. Equations are given in the following "Z8530H INTERFACE PAL EQUATIONS" section to assert ZINTACK at the beginning of the first INTA0 cycle.

The interrupt vector must be valid 25 ns prior to T4.

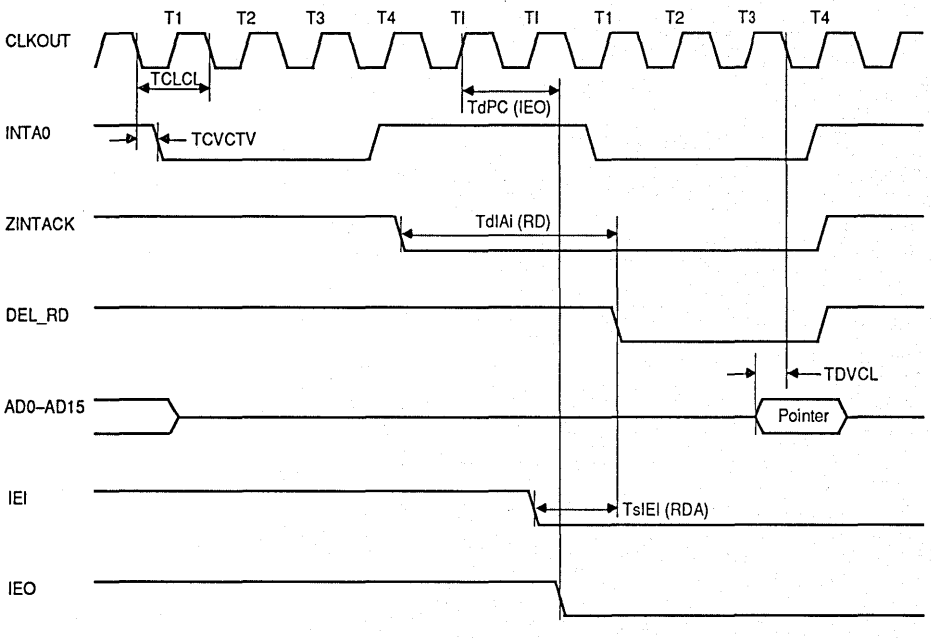
$$T_{DVCL} = 3\,T_{CLCL} - T_{CVCTV} - T_{PD} - T_{dRDA(DR)}$$
$$= 375 - 70 - 15 - 140$$
$$= 150 \text{ ns}$$



Z8530H Read Cycle Timing Diagram



Z8530H DMA Cycle Timing Diagram



Z8530H Interrupt Acknowledge Timing Diagram

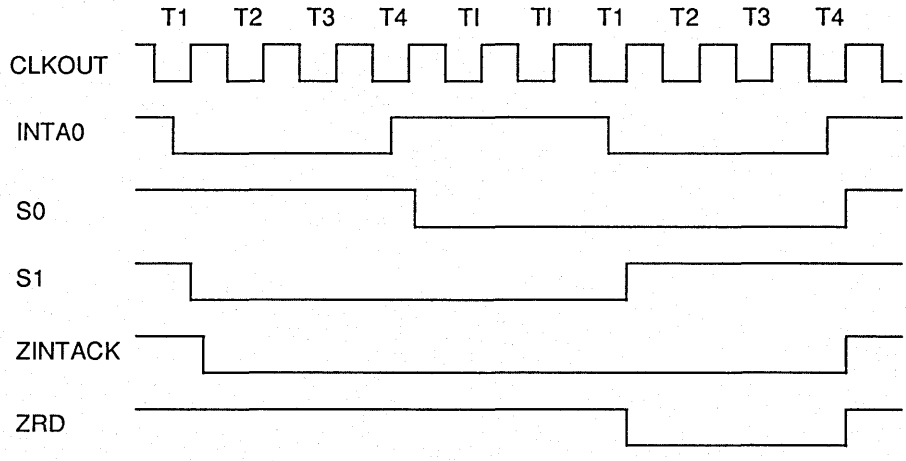
Z8530H INTERFACE PAL EQUATIONS

NOTE: Do not minimize these equations.

```
Device Z8530H (P16L8)
PIN
    RESET       = 1     (INPUT COMBINATORIAL)
    /READ       = 2     (INPUT COMBINATORIAL)
    /WRITE      = 3     (INPUT COMBINATORIAL)
    CLKOUT      = 4     (INPUT COMBINATORIAL)
    /INTA0      = 5     (INPUT COMBINATORIAL)
    /DTR_REQ    = 6     (INPUT COMBINATORIAL)
    /WREQ       = 7     (INPUT COMBINATORIAL)
    /ZINT       = 8     (INPUT COMBINATORIAL)
    A19         = 9     (INPUT COMBINATORIAL)

    /INT0       = 12    (OUTPUT ACTIVE_LOW COMBINATORIAL)
    /DS0        = 13    (OUTPUT ACTIVE_LOW COMBINATORIAL)
    /ZS0        = 14    (OUTPUT ACTIVE_LOW COMBINATORIAL)
    /ZINTACK    = 15    (OUTPUT ACTIVE_LOW COMBINATORIAL)
    /DEL_RD     = 16    (OUTPUT ACTIVE_LOW COMBINATORIAL)
    /DEL_WR     = 17    (OUTPUT ACTIVE_LOW COMBINATORIAL)
    /DRQ1       = 18    (OUTPUT ACTIVE_LOW COMBINATORIAL)
    /DRQ0       = 19    (OUTPUT ACTIVE_LOW COMBINATORIAL);

BEGIN
ENABLE (INT0, ZS0, ZINTACK, DEL_RD, DEL_WR, DRQ1, DRQ0);

ZS0 = RESET + (ZS0 * /INTA0 + ZS0 * ZINTACK + INTA0 * ZINTACK);
ZINTACK = /RESET * (/ZS0 * /INTA0
                  + /ZS0 * ZINTACK
                  + INTA0 * ZINTACK);

DRQ1 = /DS0 * DRQ1
    + A19 * DEL_WR * /DS0
    + /DS0 * /DTR_REQ + DRQ1 * /DTR_REQ
    + RESET;
DS0 = DS0 * DRQ1 + /DTR_REQ * DRQ1 + RESET;

DRQ0 = /WREQ;
INT0 = /ZINT;

DEL_WR = WRITE * DEL_WR
    + CLKOUT * WRITE
    + RESET;
DEL_RD = READ * DEL_RD
    + CLKOUT * READ
    + ZINTACK * INTA0
    + RESET;

END.
```

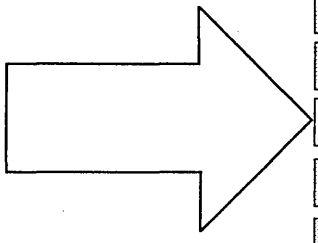
To support a daisy-chained interrupt structure, an extra output is required. This precludes using a 16L8 to implement the complete interface. A 22V10 could easily be utilized. The following equations would generate ZINTACK beginning at the start of the first INTA0 pulse. All signals are combinatorial active low.

```
S0 = (S0*S1 + /INTA0*S1 + INTA0*S0)*/RESET;
S1 = (/S0*S1 + /INTA0*S1 + INTA0*/S0)*/RESET;
ZINTACK = S0 + S1;
DEL_RD = READ * DEL_RD
    + CLKOUT * READ
    + S0 * INTA0
    + RESET;
```



Daisy-Chained Interrupt Acknowledge Sequence

Advanced
Micro
Devices



CHAPTER 3
GRAPHICS PRODUCTS

Am8052

Alphanumeric CRT Controller (CRTC)

FINAL

DISTINCTIVE CHARACTERISTICS

- On-chip DMA capability, operating via linked-list data structures
- Three on-chip row buffers, each 132 characters by 20 bits support split-screen smooth-scrolling
- General-purpose microprocessor interface. Compatible with 8086, Z8000*, and 68000 CPUs.
- Smooth-scrolling capability, with minimal CPU overhead
- Multiple vertical and horizontal screen divisions, with optional smooth-scrolling within a window
- Character attributes (12 bits) can be invoked on a character-by-character basis
- Flexible vertical and horizontal sync control
- Flexible blanking for control of front and back-porch positions
- Non-interlace, repeat field interlace, and video interlace options
- High resolution 5-bit character generator row addressing
- 16M-byte system memory addressing capability
- Programmable blink options for cursors and characters

GENERAL DESCRIPTION

The Am8052 CRT Controller (CRTC) is a general-purpose interface device for raster scan CRT displays. The CRTC provides efficient manipulation of complex character formats and screen structures to allow sophisticated text display without undue CPU overhead.

The CRTC is a register-oriented product that is fully user programmable. The timing definition and operating modes are initialized by the host CPU. Display formats are real-time programmable on a row-by-row basis. Character attributes are specified on a character or field basis, and are interpreted and acted upon during active display of a character row.

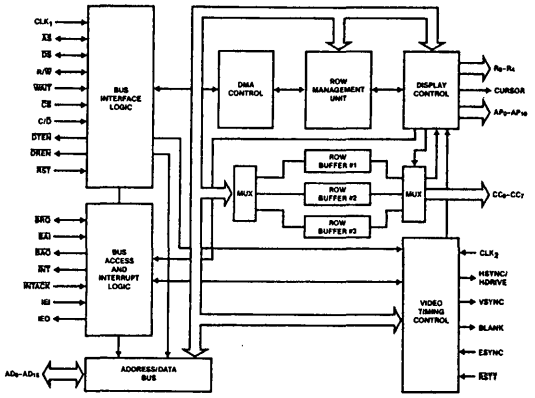
Internal DMA capability assures efficient transfer of display information to the three on-chip line buffers. These three line buffers prevent screen flashing in split-screen smooth-scrolling operations. The DMA loads the line buffers via linked list data blocks which facilitate easier editing and text composition.

The Am8052, in conjunction with the Am8152A bipolar Video System Controller (VSC), allows for the flexible assignment of visual attributes. The twelve attribute bits stored in the Am8052 include superscript, subscript, blink, highlight, reverse, underline, strike through and cursor. Both character and cursor can be made to blink at three different rates, and the blink duty cycle is programmable. Further flexibility is achieved by the Am8152A, which allows the video stream to be manipulated by selection of background and foreground as well as background/foreground reversal.

The Am8052 and Am8152A combination also supports proportional spacing, text justification, and double-width characters.

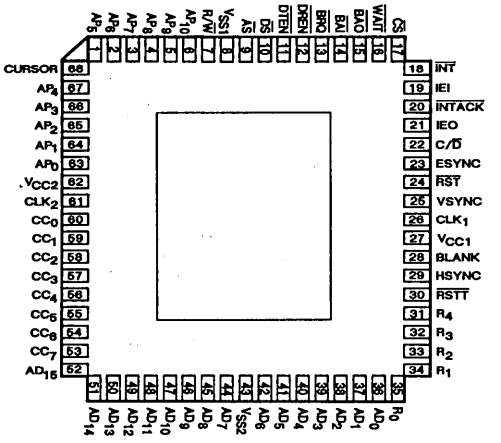
The Am8052 CRTC is assembled in 68-pin plastic leaded chip carrier and ceramic leadless chip carrier packages, while the bipolar Am8152A VSC is assembled in a 48-pin DIP and 68-pin Plastic Leaded Chip Carrier. These interface circuits are available as a chip-set for high performance CRT applications.

BLOCK DIAGRAM



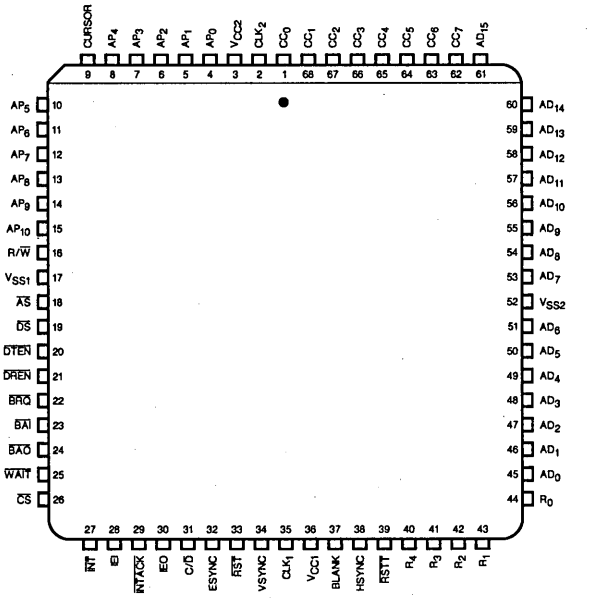
BD003361

CONNECTION DIAGRAMS
Top View

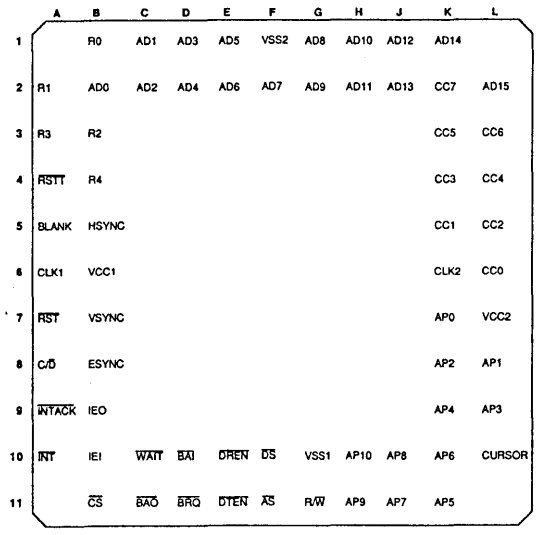


CD005191

68-Pin PLCC
Generic Outline

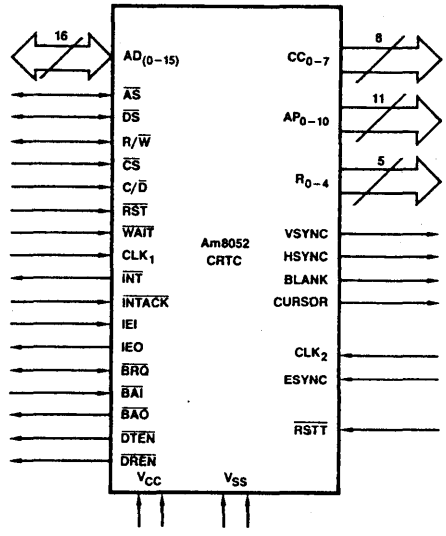


CD010410



CD010420

LOGIC SYMBOL



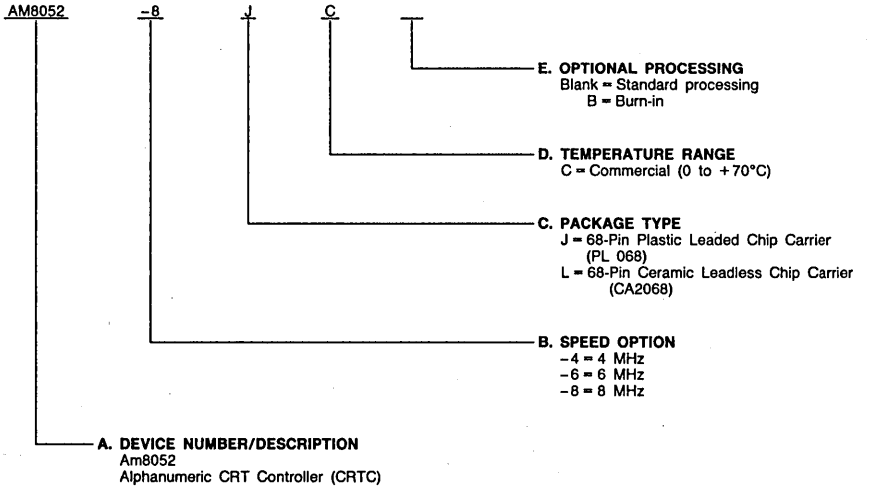
LS001211

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

A. Device Number
B. Speed Option (if applicable)
C. Package Type
D. Temperature Range
E. Optional Processing



Valid Combinations	
AM8052-4	JC, LC
AM8052-6	
AM8052-8	

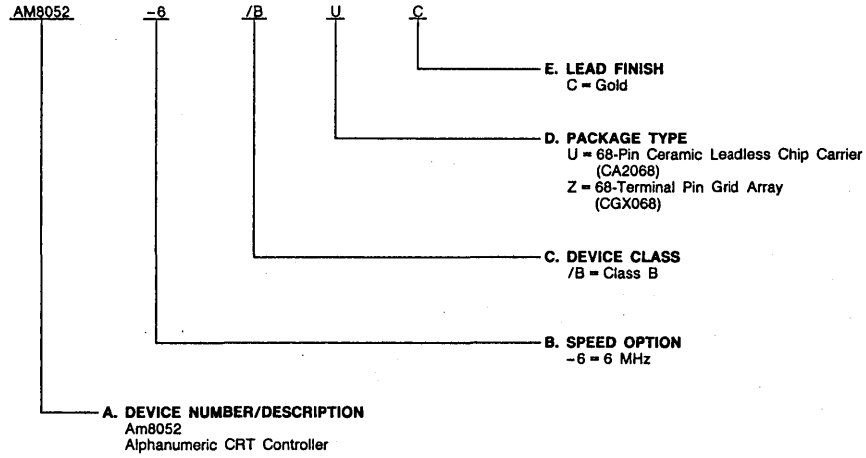
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION (Cont'd.)

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. CPL (Controlled Products List) products are processed in accordance with MIL-STD-883C, but are inherently non-compliant because of package, solderability, or surface treatment exceptions to those specifications. The order number (Valid Combination) for APL products is formed by a combination of: **A. Device Number**
B. Speed Option (if applicable)
C. Device Class
D. Package Type
E. Lead Finish



Valid Combinations	
AM8052-6	/BUC, /BZC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

V_{SS1}, V_{SS2} Ground

V_{CC1}, V_{CC2} +5-V Power Supply

CLK_1 Timing Clock
The Clock 1 signal controls and times the DMA and peripheral portion of the CRTC. In proportional spacing applications, where CLK_2 is variable, CLK_1 must be used to time the horizontal and vertical sync rates. CLK_1 is non-TTL compatible, and is normally driven by the Am8152A VSC.

CLK_2 Display Clock
The Clock 2 signal is used to time character accesses from the CRTC line buffers. In applications which do not use proportional spacing, CLK_2 is fixed in frequency and can be used to time horizontal and vertical sync rates, allowing CLK_1, the system clock, to be unrelated and asynchronous to the display timing. CLK_2 is non-TTL compatible and should be driven by the VSC.

AD_0–AD_{15} Address/Data Bus (Input/Output, Three-State)
The Address/Data Bus is a multiplexed, bidirectional, high-true, three-state bus. The presence of addresses is defined by the $\overline{AS}$ signal, and the presence of data is defined by the $\overline{DS}$ signal. When the CRTC is in control of the system via its internal DMA capability, it controls the AD Bus; when the CRTC is idle, the CPU or other external devices control the AD Bus and may use it to access the internal registers of the CRTC. The high-order 8-bit memory address is output on the $AD_0 - AD_7$ lines. Interrupt vector information is also output in the $AD_0 - AD_7$ lines.

$\overline{AS}$ Address Strobe (Input/Output; Three State, Active LOW)
Address Strobe is a bidirectional, active-LOW, three-state signal. When the CRTC is in the slave mode and the bus master is accessing the CRTC's internal registers, $\overline{AS}$ can be used to optionally latch $\overline{CS}$ and C/$\overline{D}$ information during the first part of the transaction. During a DMA operation when the CRTC is in control of the system, $\overline{AS}$ is an output generated by the CRTC to indicate a valid address on the bus. In the slave mode, the $\overline{AS}$ signal may be asynchronous to CLK_1.

$\overline{DS}$ Data Strobe (Input/Output; Three State, Active LOW)
Data Strobe is a bidirectional, active-LOW, three-state signal. When the CRTC is in the slave mode and the external system is transferring information to or from it, $\overline{DS}$ is a timing input used by the CRTC to move data to or from the AD Bus. In the slave mode, the $\overline{DS}$ signal may be asynchronous to CLK_1. During a DMA operation when the CRTC is in control of the system, $\overline{DS}$ is an output generated by the CRTC and used by the system to move data onto the AD Bus.

$\overline{CS}$ Chip Select (Input, Active LOW)
The $\overline{CS}$ input is an active-LOW signal used by the host processor to select the CRTC for a slave transfer.

$\overline{WAIT}$ Wait (Input, Active LOW)
The $\overline{WAIT}$ input is an active-LOW signal used to stretch the $\overline{DS}$ strobe whenever the CRTC has access to the host's bus for data transfer. The status of the $\overline{WAIT}$ signal is sampled on the falling edge CLK_1 during t_2 or t_W.

R/$\overline{W}$ Read/Write (Input/Output, Three State)
Read/Write is a bidirectional, three-state signal indicating the data direction for the bus transaction under way, and remains stable for the length of the bus cycle. When $\overline{CS}$ input is active, Read (HIGH) indicates that the system is requesting data from the CRTC and Write (LOW) indicates that the system is presenting data to the CRTC. On the other hand, during a DMA operation when the CRTC is in control of the system, R/$\overline{W}$ is an output generated by the CRTC, with Read indicating that data is being requested by the CRTC from the addressed memory location and Write indicating that the CRTC is driving a high-order address to an external latch.

$\overline{BRQ}$ Bus Request (Input/Output, Three State)
When the CRTC requires use of the bus for DMA activity, the $\overline{BRQ}$ line is driven LOW. It remains LOW until it has ceased using the bus.

$\overline{BAI}$ Bus Acknowledge In (Input, Active LOW)
Bus Acknowledge In is an active-LOW input. When the CRTC requires host bus access and has successfully pulled its $\overline{BRQ}$ pin LOW, a $\overline{BAI}$-LOW input signifies that the CRTC has obtained bus mastership after having internally synchronized its $\overline{BAI}$ active-LOW input for two clock periods of CLK_1. The synchronization is required to alleviate metastable problems. When the CRTC does not require host bus access, the $\overline{BAI}$ input ripples to the $\overline{BAO}$. Forcing $\overline{BAI}$ HIGH will cause the Am8052 to relinquish the bus.

CURSOR Cursor (Output)
This pin is the cursor output indicator.

ESYNC External Sync (Input)
This pin is the external synchronization input line. If the ES bit in the mode register is set, the vertical frame scan will commence after the rising edge of ESYNC.

HSYNC Horizontal Sync (Output, Active HIGH)
HYSNC is an active-HIGH output used to cause horizontal retrace of the CRT's electron beam. The output is held active LOW while the CRTC is reset to prevent unknown synchronization to the CRT which may cause damage to high bandwidth tubes. Note that this pin can also be initialized as Horizontal Drive.

VSYNC Vertical Sync (Output, Active HIGH)
VSYNC is an active-HIGH output used to cause vertical retrace of the CRT's electron beam. VSYNC can be optionally synchronized by the ESYNC input. VSYNC is held LOW while the CRTC is reset to prevent damage to the CRT.

BLANK Blank Video (Output, Active HIGH)
BLANK is an active-HIGH output. It serves to blank out inactive display areas of the CRT. The output is held active while the CRTC is reset.

R_0–R_4 Row Control (Output, Active HIGH)
R_0–R_4 outputs are active HIGH. These outputs represent the binary count of the active scan line being displayed. These outputs address the least signficant address portion of an external character generator. The outputs are all held HIGH for those scan lines that do not carry active video during normal character or superscript/subscript display.

CC_0–CC_7 Character Code (Output, Active HIGH)
CC_0–CC_7 outputs are active HIGH. The 8-bit character port, $CC_0 - CC_7$, outputs eight bits of data stored in the character code section of the line buffer currently being displayed.

$\overline{INT}$ Interrupt Request (Output; Open Drain, Active LOW)
This line is used to indicate an interrupt request to the host processor. It is driven LOW by the CRTC until an interrupt acknowledge is received on the $\overline{INTACK}$ pin or the relevant IP or IE bits in Mode Register 2 are reset.

$\overline{\text{INTACK}}$ Interrupt Acknowledge (Input, Active LOW)
When $\overline{\text{INTACK}}$ is driven LOW, the CRTC examines its IEI line to determine whether it has been granted an acknowledge by the CPU. It also starts priority resolution of the Daisy Chain. When $\overline{\text{DS}}$ is active, the vector is placed on the bus if enabled.

IEI Interrupt Enable In (Input)
A HIGH on IEI during an Interrupt Acknowledge cycle is regarded as an interrupt acknowledge to the CRTC. A LOW on IEI during Interrupt Acknowledge signifies that a higher priority interrupt on the dasiy chain is being acknowledged.

IEO Interrupt Enable Out (Output)
IEO follows IEI during Interrupt Acknowledge if the CRTC has not made an interrupt request. IEO LOW disables lower priority devices from making interrrupt requests.

$\overline{\text{DTEN}}$, $\overline{\text{DREN}}$ $\overline{\text{Data Transmit Enable}}$, $\overline{\text{Data Receive Enable}}$ (Outputs; Open-Drain, Active LOW)
Data Transmit Enable and Data Receive Enable are used to control bus transceivers external to the CRTC should they be required. When $\overline{\text{DTEN}}$ is LOW, the transceiver should transmit from the CRTC onto the bus. When $\overline{\text{DREN}}$ is LOW, the transceiver should receive data from the bus. $\overline{\text{DTEN}}$ and $\overline{\text{DREN}}$ are never LOW simultaneously.

C/$\overline{\text{D}}$ Command/$\overline{\text{Data}}$ (Input)
C/$\overline{\text{D}}$ is used by the CRTC when in the slave mode to determine if an I/O transaction with the host CPU is transferring a command or data. When the CRTC is not involved in an I/O transaction with the host, C/$\overline{\text{D}}$ is disregarded.

AP_0–AP_{10} Attribute Port (Output)
These 11 lines are used to display character attribute information synchronous with each character and CLK_2. During HSYNC, the row attribute information contained in the Row Redefinition Block is output on AP_0–AP_{10}.

$\overline{\text{BAO}}$ $\overline{\text{Bus Acknowledge Out}}$ (Output, Active LOW)
$\overline{\text{BAO}}$ output is forced active HIGH when the CRTC requests bus mastership; otherwise, the $\overline{\text{BAI}}$ input ripples out of the CRTC via the $\overline{\text{BAO}}$ output.

$\overline{\text{RST}}$ $\overline{\text{Reset}}$ (Input, Active LOW)
A LOW on this input for at least 5 clock cycles is interpreted as a reset signal. The effect of reset is to drive all CRTC bus signals into the high-impedance state, to clear all mode bits except bits 9 through 15 in MR2, and to force the CRTC into the slave mode.

$\overline{\text{RSTT}}$ $\overline{\text{Test Reset}}$ (Input)
For test use only. This pin is a "No Connect."

TABLE 1. CHARACTER ATTRIBUTE DESCRIPTION

Attribute	Effect
Reverse	– Causes the designated character to be displayed in reverse video.
Highlight	– Highlights the applicable character.
Blink	– Blinks the designated character at one of four programmed blink rates.
Underline	– Underlines the designated character at a programmable scan line.
Subscript	– Causes the character to be displayed as a subscript.
Superscript	– Causes the character to be displayed as a superscript.
Shifted Underline	– A second underline.
Cursor	– Causes the attribute or X-Y cursor to be displayed at the designated character position.
Latched	– Indicates that the attribute should be latched for all successive characters until changed.
Ignore	– Causes the CRTC to skip over the designated characters. Useful for embedded control characters and protected fields that do not get displayed.
User Definable	– Four attribute bits reserved for user definition.

FUNCTIONAL DESCRIPTION

The block diagram of the Am8052 CRTC is shown on front cover. Communication with the external host system takes place over the 16-bit Address/Data Bus, AD_0–AD_{15}. Transfers over the AD Bus are controlled by the $\overline{CS}$, C/$\overline{D}$, $\overline{AS}$, $\overline{DS}$, and R/$\overline{W}$ lines. When the CRTC is in the slave mode, these four bus control lines are inputs. When the CRTC is in the DMA mode, $\overline{AS}$, R/$\overline{W}$ and $\overline{DS}$ are outputs and control the external bus.

Following reset, the host system initializes the CRTC's timing and control registers, as well as one address pointer to the start of the display data location in the host memory. Following initialization and upon command from the host, the CRTC takes over bus control from the host and transfers display row control data, character code, and character attribute data. The CRTC requests the host bus by sampling the $\overline{BRQ}$ line for activity; if the $\overline{BRQ}$ line is HIGH, the CRTC drives it LOW, and also drives $\overline{BAO}$ HIGH, to obtain priority over lower priority bus requestors. The on-chip DMA Controller circuit controls the data transfer and performs character data loading into the on-board line buffers.

The CRTC is real-time programmable on a character row-by-row basis through a row control data block fetched either from the host memory or from a dedicated display memory. The Row Control Block (RCB) contains address links to the next row's RCB, a character and attribute data address for the current row and other pertinent control functions for the row. Data from the RCB is transferred into the appropriate set of registers for active control of display and data fetch operations during the subsequent display of character row data. A Top of Page register contains the address of the Main Definition Block for the screen. The Main Definition Block, in turn, points to the first RCB. The character row data, comprised of character code and attribute (if the latter is specified), is fetched starting at the address and for the character length obtained from the RCB. The character code and its attribute consist of a 20-bit wide word which is stored, FIFO style, into one of the three on-board 132-character by 20-bit line buffers. Character attributes are on a character-by-character basis and are interpreted and acted upon by the CRTC during the active display period of the contents of a line buffer. Output lines CC_0–CC_7 form the transfer path for character code data to an external matrix type character generator, while the character attribute, after selective masking, is interpreted and combined with the resulting video.

Output lines R_0–R_4 exhibit the scan line number for the specific character being displayed, while the character row control logic allows alteration of the scan line number output at the R_0–R_4 lines to enable the display of normal superscript or subscript characters.

The HSYNC, VSYNC and BLANK output lines provide the CRT synchronization signals. The horizontal and vertical control logic blocks contain counters and host programmable registers for deriving the timing signals from either the CLK_1 or the CLK_2 input as well as an ESYNC input line for frame synchronization to an external source, such as the power line frequency. CLK_2 runs at the display character rate. It is a submultiple of the dot clock, whose frequency is determined by the Am8152A oscillator. CLK_2 controls the CRT synchronization lines HSYNC and VSYNC, as well as BLANK, and the rate of character output from the CRTC. CLK_1, which may be asynchronous to CLK_2, controls all DMA and related bus activity, associated with the CRTC. In proportional spacing applications, CLK_1 may be also used to time the synchronization signals.

Character Attributes

Character attributes affect various CRTC output signals and other operations on a character-by-character basis. Each attribute word occupies a 16-bit word in memory. Each character, however, need not invoke a new attribute.

Character attributes are stored in parallel with the corresponding character code in each line buffer.

The character attribute information which makes up the character attribute word is shown below:

AW_{15}	Latched/$\overline{\text{Unlatched}}$	AW_7	User definable
AW_{14}	Cursor	AW_6	Highlight
AW_{13}	Ignore	AW_5	Reverse
AW_{12}	Reserved	AW_4	Superscript
AW_{11}	Reserved	AW_3	Subscript
AW_{10}	User definable	AW_2	Shifted Underline/ Strike Through
AW_9	User definable	AW_1	Underline
AW_8	User definable	AW_0	Blink

Latched/$\overline{\text{Unlatched}}$

When this bit is set to 1 ("latched"), the attribute information applies to all characters following the character that invoked the attribute word. Only the presence of a further latched attribute word cancels the effect of a previous latched attribute word. If the Latched/$\overline{\text{Unlatched}}$ bit is set to 0 ("unlatched"), then the attribute information only applies to the character that invoked the attribute word. All successive characters are modified by the latched attribute information that was valid prior to the unlatched attribute word. The Latched/$\overline{\text{Unlatched}}$ bit is not output to the Attribute Port. The initial state of the latched attribute value is undefined. At the start of any horizontal line, the latched attribute information is the same as at the end of the previous line, unless changed by a further latched attribute.

Cursor

If this bit is set, then a cursor is displayed at the affected character position(s), dependent upon the mode of the cursor display logic. See the section on cursor display for further details.

Ignore

When the Ignore is set, it inhibits the loading of the associated character into the CRTC line buffer. Such character(s) may be used as control character or software tags, and are not displayed. Whenever the Ignore encoding is detected, both the attribute word and its associated character code are not written into the line buffer, unless the DH (Display Hidden) bit in Mode Register 1 is set. Note that the Ignore bit is not brought out to the Attribute Port.

User Definable

The AW_7–AW_{10} attribute bits provide 4 bits of user definable attribute information. These bits are directly output on pins AP_7–AP_{10} of the Attribute Port. (In addition to these four user-definable attribute bits, the Cursor bit can also be user-definable under certain conditions.)

Highlight

When this bit is set and AP_6 is connected to the Foreground Shift (FS) input of the Am8152A, the character is displayed highlighted. The AP_6 pin of the Attribute Port goes active for each scan line of the relevant character(s).

Reverse

When this bit is set and AP_5 is connected to the REV input of the Am8152A, the character is displayed reversed. The AP_5 pin of the Attribute Port goes active for each scan line of the relevant character(s).

Superscript

When this bit is set to 1, the affected character is displayed as a superscript. Its position on the character row ($R_0 - R_4$) is determined by the superscript control field in the Row Redefinition Block for that particular row.

Subscript

When this bit is set, the affected character is displayed as a subscript. Its position on the character row (R_0–R_4) is determined by the subscript control fields in the Row Redefinition Block.

Underline/Shifted Underline

Attribute bits AW_1 and AW_2 provide underline and shifted underline display. The underline/shifted underline display information is output on the AP_1 and AP_2 Attribute Port pins, during applicable scan lines of the character. (The applicable scan lines have been programmed within the Row Redefinition Blocks.)

Blink

When this attribute is invoked, the Attribute Port pin AP_0 is gated with the character blink rate generator, during the time that the relevant character is output on CC_0–CC_7.

The character blink rate and character blink duty cycle are derived from the blink field of the Main Definition Block.

Attribute Fetches

Attributes can be fetched in three different ways to suit most design philosophies (see Figure 2). In Option 1, one attribute is fetched per character. This option, although straightforward, imposes heavy bus overhead since the DMA has to access the attribute list from memory for every character displayed on the screen. Bus overhead can be reduced considerably by fetching attributes on a demand basis. Options 2 and 3 accomplish this in two different ways. In Option 2 one character bit is set to 1 when an attribute is required. When this bit is set to 0, the attribute will not be fetched. This option allows 7 bits of character code or a 128-character set for display with no overhead for attribute incorporation.

Option 3 makes use of an 8-bit flag which precedes the character invoking the attribute. This option allows for a 255-character set with an 8-bit overhead (the flag) per attribute.

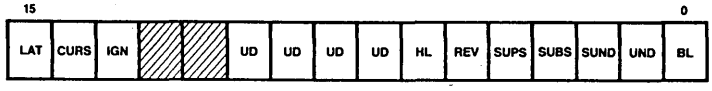


Figure 1. Am8052 Attribute Word

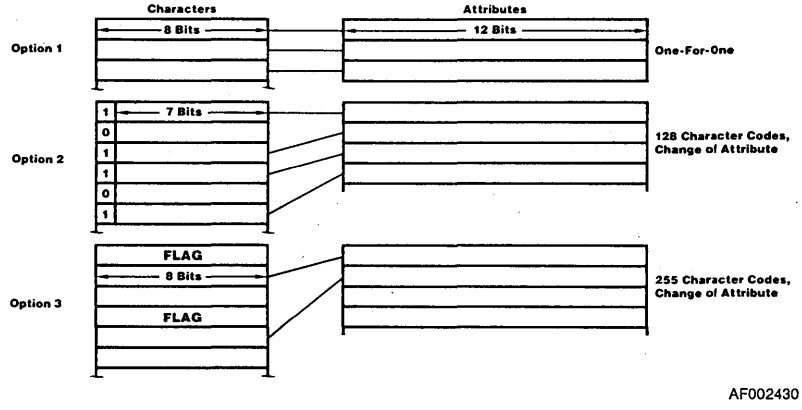


Figure 2. Attribute Fetch

Cursor Generation

The CRTC can generate three different cursor formats: block, underline, and reverse, at variable blink rates and blink duty cycles.

Cursor information for the CRTC comes from two different sources, and each source can be independently steered to one of three different destinations. The two cursor sources are:

1. The XY cursor field which is held in the Main Definition Block for the screen.
2. Attribute word bit 14 of the character attribute word. A cursor designated by an attribute will follow its row and character position whenever text is scrolled. The cursor controlled by positioning X and Y coordinates within the cursor X and Y register will be displayed on a fixed X, Y character position on the screen. The X, Y cursor should be disabled by resetting the CUE bit in Mode Register 2 during smooth-scroll.

The steering of the cursor sources is under software control of the cursor mask field within Mode Register 2. The field is divided into two three-bit segments, one for the XY cursor and

one for the attribute cursor. Three destinations are selectable for each cursor source:

(a) The cursor pin
(b) The underline pin
(c) The reverse video pin.

If (a) is selected, then either the whole character cell or partial character cell is selectable. If whole is selected, the cursor pin will be active for every scan line of the character cell. If part is selected, then the cursor pin will only be active for those scan lines within the limits of CURSOR START and CURSOR END, as specified in the Row Redefinition Block (RRB).

If (b) is selected, then either an underline will be active, if CURSOR START and END have the same values, or a block, if CURSOR START and END are not coincident.

If (c) is selected, then either all or part of the character will be reversed, dependent upon the CURSOR START and CURSOR END setting as explained in (b).

In addition to these choices, either cursor can be made to blink (at the cursor blink rate) and duty cycle (as programmed into the Main Definition Block blink field).

Row Buffers

The on-chip DMA controller accesses the display memory and loads data from linked-list data blocks in memory into one of three row buffers. Each line buffer is 132 characters in length and 20 bits wide. Each 20-bit wide location accommodates an 8-bit character code and 12-bit attribute words. The row buffers operate in a rotating fill-display mode whereby one buffer is being loaded while another is being displayed.

The presence of three row buffers on-chip is of significant advantage in split screen smooth-scrolling operations where a character row may only be displayed for a single scan line. With two row buffers, this would not leave enough time for the reloading of the alternate line buffer. A partially filled buffer results in screen flashing. This can only be prevented by incorporating three line buffers. Figure 3 highlights this advantage.

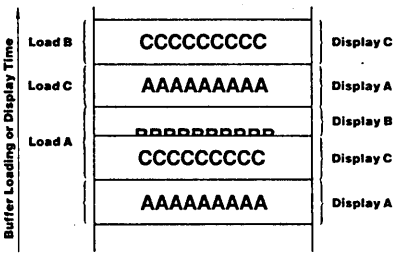


Figure 3. Triple Row Buffers

In the rotating fill-display mode, Row Buffer C is displayed when Row Buffer B is being loaded. Likewise, the next Row Buffer C is loaded while Row Buffer A is being displayed. Because of the split-screen, Row Buffer B is displayed for one scan line only, while Row Buffer A is being loaded. By virtue of the third row buffer, the loading of Buffer A can spill over into the next buffer display, thus eliminating screen flashing.

Smooth-Scrolling

A smooth-scroll is defined as the gradual displacement of a character row on a scan line-by-scan line basis. Smooth-scrolling is achieved by a gradual offsetting of the scan line counter, on a frame-by-frame basis. At the start of the scroll, the offset counter is set to zero or equal to the number of scan lines per character row, depending on whether the scroll is up or down. As the counter is incremented or decremented, the text travels up or down until the offset is equal to the number of scan lines or zero. The start of the screen pointer pointing to the character row is adjusted and the offset counter reset simultaneously to scroll the next successive character row. Smooth-scrolling of the entire screen is thus a simple task.

A number of applications require screen overlays, such as menu or status areas which must remain static while the major portion of the screen is scrolling or vice versa. The Am8052 can support multiple windows, each capable of being scrolled. (Only one window can be scrolled at a time.)

Linked-List Data Structures

The DMA channel on the Am8052 operates via linked-list data structures that allow for the overlaying and independent smooth-scrolling of windows. The linked-list data structures are particularly suited to the manipulation of data strings where insertions and deletions are common. A typical CRTC linked-list structure is shown in Figure 4.

The linked list consists of Row Control Blocks (RCBs) for each character row on the screen. The RCB does not contain any displayable data, but contains the address which points to the character information. Each RCB is linked to the next block via an address link word (RCB ADR). The structure of the RCB linkage is shown in Figure 5. The Top of Page register on-chip points to the Main Definition Block, which in turn points to a linked list of RCBs.

The Am8052 allows for the separation of attribute and character lists. By extending the RCB, split-screen segments can be constructed as in the case of RCB_2 in Figure 5. In parallel with the screen or background data structure, there exists a window structure which contains Window Control Blocks (WCBs) for each row of each window. Windows can exist in any position on the screen and are overlayed on top of the screen or background information. For example, the structure shown in Figure 6 could be used to implement a menu overlay at the top of the screen together with a status overlay.

Main Definition Block

The Main Definition Block is a set of control data and addresses, located in the system memory, which allow the user to specify screen oriented features. The Top of Page register points to the first word of the Main Definition Block. Cursor position, fill code and scroll rate are set by the appropriate fields within the block. The Main Definition Block also points to the first RCB.

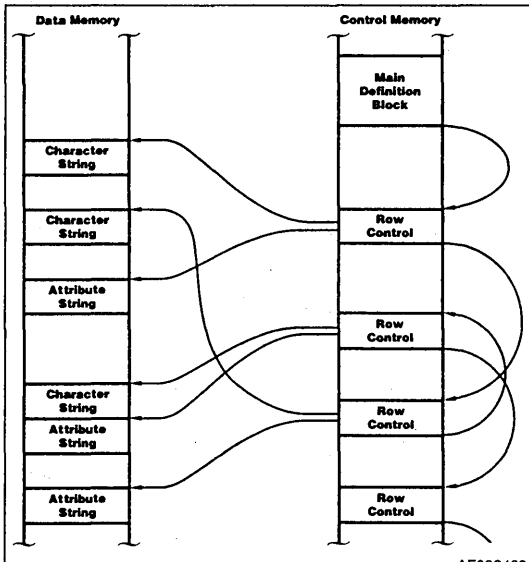


Figure 4. Am8052 Linked-List Structure

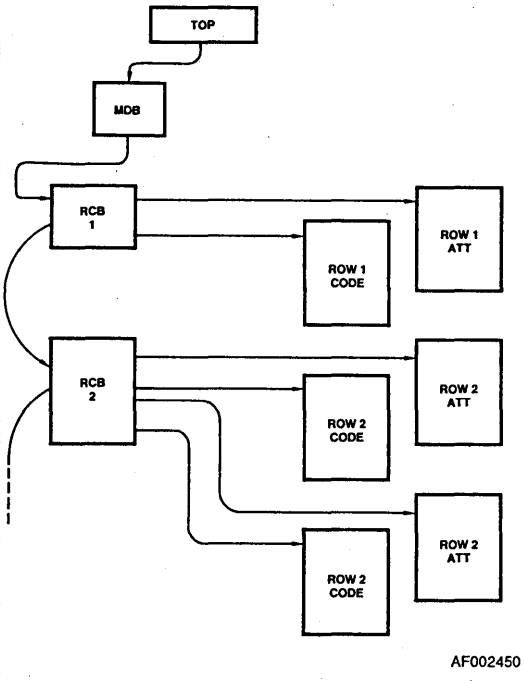


Figure 5. Background Data Structure

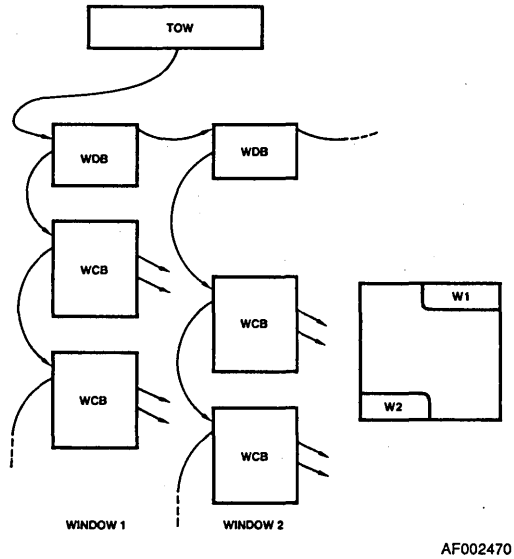


Figure 6. Window Data Structure

Row Control Blocks

The RCB Pointer in the Main Definition Block points to the first word of the first Row Control Block (RCB) of the list. Each RCB in the main chain is linked to the next via the RCB Pointer. Changing the RCB Pointer within the chain allows quick insertion or deletion of character rows.

Attributes associated with characters exist in their own separate lists. A character row may be composed of one or more segments of data. Each segment is a block of words with consecutive addresses. An RCB has a character code pointer (2 words) and an attribute pointer (2 words) for each segment. A fifth word, HIDDEN # and VISIBLE #, defines the number of characters (byte count) contained in the segment as well as the number of displayed characters in the segment. Character attributes are in word format, and there can be as many character attributes as character codes.

Window Definition Block

The Window Definition Block (WDB) defines the size and location of the window. It is the header block for a list of Window Row Control Blocks (WRCB) and can also point to another WDB if more than one window is displayed on the screen. The Top of Window (TOW) register points to the first word of the first WDB. Within the first WDB, the WRCB Pointer points to the current window's first WRCB, while the next WDB Pointer points to the next window's WDB. Window size is specified by two words in the WDB. START WINDOW ROW # and END WINDOW ROW # are byte values which position the window vertically on the screen. The window display becomes active in the character row number specified by START WINDOW ROW # and will become inactive in the character row following END WINDOW ROW #.

Window Row Control Blocks

The Window Row Control Blocks (WRCBs) have the same format as the RCBs.

The WCB Pointer is the address link to the next row's WRCB. A window can also be described with segments, and the WRCB contains five words for each segment.

To hard-scroll a window, it is only necessary to change the WRCB Pointer in the WDB to an adjacent WRCB.

Window Display Mechanism

A window is any bounded area on the screen which is linked in by a WDB. The window has the following size characteristics:

Width: Defined by the number of character code positions occupied within a character row. Maximum width is the length of the line buffer (132 characters), and minimum width is one character.

Height: Defined by the number of displayable character rows contained within the window. The maximum height is the total number of displayed character rows on the face of the screen. The height limit is specified by the number of WRCB in the window linked list. The minimum height of a window is one row.

Window Positioning

The window is originally positioned to occupy any portion of the displayable character rows. It can be as large as the full screen or as small as one row high and one character wide. The window is always unscrolled when first displayed. (The counter holding the value of the first scan line of the uppermost character row of the window is reset.)

The window must be positioned horizontally such that its left- and right-hand sides begin and end at a background character row segment boundary. Any unfilled character positions within the window segment, and following the end of the window segment to the end of the line buffer (character position 131), are filled with the fill character code obtained from the Main Definition Block (MDB).

Multiple Windows

Multiple windows can be displayed simultaneously. Windows cannot be horizontally aligned to each other, and hence must be specified on non-overlapping character row boundaries (see section on virtual windows). Each window is defined by a WDB, and the scrolling windows are designated by a control bit within the WDB.

Window Positioning

The window position is defined in the WDB. The coordinate units are background character rows and background character columns. When the background is scrolling, the window (or windows) remain stationary on the display.

Example of Window Overlays

The example (Figure 7) explains how windows are constructed using the linked-list feature that the Am8052 provides.

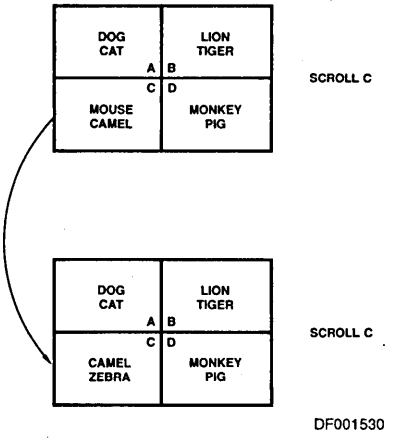


Figure 7. Example of Vertical Split Screen Smooth Scroll

Step 1

The first step toward constructing windows on a CRT screen is to split the screen horizontally and vertically using RCBs with multiple data pointers. The data pointers in each RCB point to the first characters within each subscreen area defined by the horizontal/vertical splits. In this example, the RCB that controls the first character row (DOG/LION) contains two data pointers. The first points to subscreen DOG and the second to subscreen LION. The segment length information in the RCB indicates to the DMA when to switch from data field DOG to data field LION. The linked-list structure for this example is shown in Figure 8. Note that in most applications, this split screen will have been set up prior to the invocation of the window.

Step 2

A window can now be overlayed on to the background by the creation of a window linked-list as shown in Figure 9. The scrollable window has a linked list structure pointed to by the Top of Window (TOW) Pointer which functions similarly to Top of Page (TOP). The other information required for window definition is the START WINDOW CHAR # and END WINDOW CHAR # which define the start/end coordinates of the window. To effect a window scroll, just one change to the toW value is required, which significantly relieves CPU overhead.

Virtual Windows

Although the rules of multiple windows do not permit overlapping windows, the background and window structures can be used to implement virtual horizontally aligned windows. This can be best described by using the illustration in Figure 7. The screen is divided into 4 subscreens: A, B, C and D; each can be independently defined as a window using a linked-list structures similar to Figure 9.

If subscreen C is defined as a window, subscreens A, B, and D are configured to be the background. Window C can be scrolled independently of the background by TOW Pointer manipulation. Similarly, subscreen D can be defined as a window with A, B and C configured as background. Thus, two aligned subscreens can be independently defined as windows

by intelligent use of linked-list structures, giving the user the illusion of aligned windows.

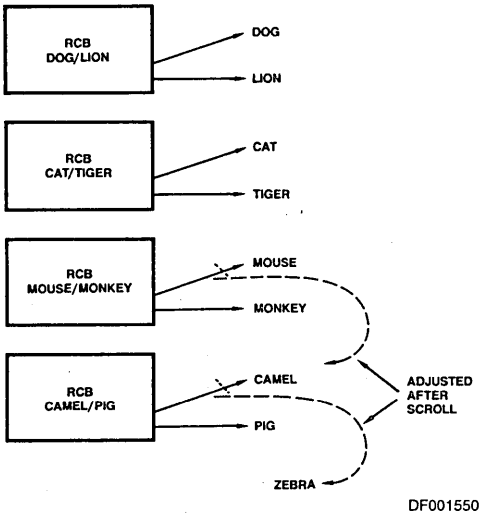


Figure 8. Split Screen Control Blocks

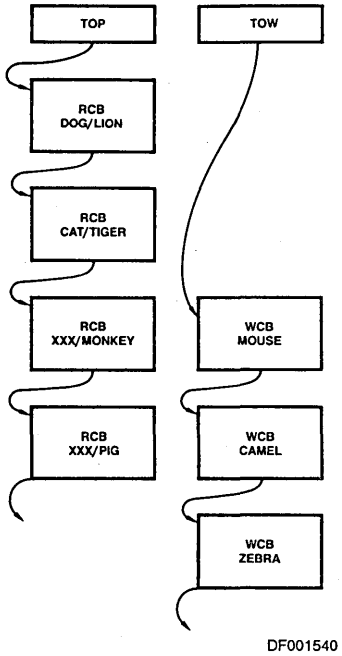


Figure 9. Window Overlay Structure

Horizontal Screen Format

The horizontal format defines the general timing of a single raster scan line. The scan line consists of two basic periods: visible raster line scan from left-to-right across the CRT screen and the right-to-left beam retrace period (or horizontal sync). The beam is always blanked during the retrace period. The front and back porch periods on either side of the horizontal sync are also blanked because no active video is desired during that time.

Horizontal scan frequencies range from a minimum of 15 kHz for small screen, low bandwidth CRTs up to about 60 kHz for 100 MHz bandwidth large screen CRTs. The horizontal format versatility must accommodate this wide range of scan frequencies. The horizontal circuit generates two basic timing signals: horizontal sync and blanking. The horizontal blanking signal is "ORed" with the vertical blanking signal prior to output at the BLANK pin.

Horizontal Timing Control

Horizontal timing is controlled by the $\overline{RST}$ signal and the DE (Display Enable) bit in the mode register.

The HSYNC output is disabled (inactive) and the BLANK output active whenever the CRTC is reset by $\overline{RST}$ input (active LOW) or whenever the DE bit is reset (display disabled). $\overline{RST}$ active LOW is a hardware reset to the CRTC (this action also resets DE bit), and the DE bit is a software reset of the CRTC.

Am8052 Vertical Screen Format

The vertical format defines the number of horizontal scan lines to be displayed in each frame. The front and rear porches, as well as the vertical retrace time, are also defined.

The CRTC operates in either an interlace or non-interlace mode. The I_1 bit, in Mode Register 1, determines if the CRTC will operate in the interlace or non-interlace mode. See below for each of the interlace options.

The Vertical Line Counter is clocked by either the horizontal sync rate in the non-interlaced or twice the horizontal sync rate in the interlaced mode. In non-interlaced mode all vertical frames (period between two vertical sync pulses) are *even*. In interlaced mode, the first vertical frame following a Display Enable (setting of DE bit in the mode register) is always *even* and alternates between odd and even from there on.

External SYNC (ES) Operation

The ESYNC input allows synchronization of the CRT display vertical frame rate to the power line frequency to eliminate interference effects. The ES bit in Mode Register 1 specifies whether the ESYNC input is used to control the vertical sync rate.

The ESYNC input is recognized by the CRTC during every frame. It causes the VSYNC signal to become active at the occurrence of HSYNC. In non-interlaced mode, VSYNC becomes active at the rising edge of HSYNC active. In interlaced mode, VSYNC either becomes active at the next HSYNC, active when in the even frame, or active at the next half point between HSYNCs (2x HSYNC) in the odd frame.

Interlace

There are two types of interlace, Repeat Field Interlace (RFI) and Interlaced Video (IV). The effect of both schemes is to offset the vertical position of the scan lines of the odd numbered fields so that they will be physically interleaved with the scan lines of the even fields. For RFI, the same video information is displayed on both odd and even fields, the slight offset of the odd field tending to eliminate the horizontal stripes that sometimes occur between scan lines of non-interlaced displays.

Interlaced Video (IV) is used to increase the amount of information displayed on a monitor without increasing the horizontal or vertical scan rates. IV takes advantage of the odd field scan line offset by displaying half the video in the even field (alternating lines) and half in the odd field. The effect is to essentially double the vertical character density with respect to RFI or non-interlace.

REGISTER SUMMARY

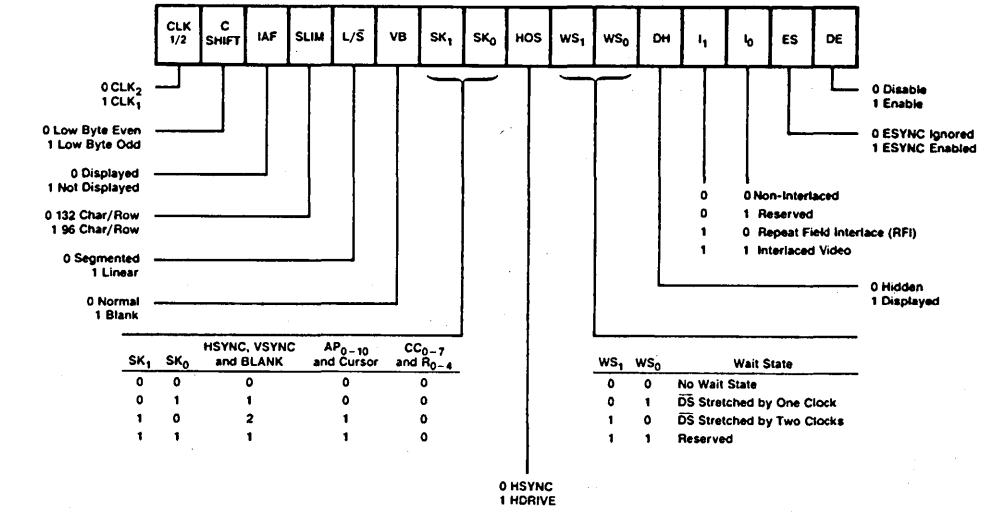
ADDRESS ($AD_4 - AD_0$)

BINARY	HEX	TYPE	ACTIVE BITS	REGISTER MODE
0 0 0 0 0	00	R/W	16	MODE 1
0 0 0 0 1	01	R/W	16	MODE 2
0 0 0 1 0	02	W	12	ATTRIBUTE ENABLE
0 0 0 1 1	03	W	5	ATTRIBUTE REDEFINITION
0 0 1 0 0	04	R/W	8	TOP OF PAGE SOFT (HI-ORDER)
0 0 1 0 1	05	R/W	16	TOP OF PAGE SOFT (LO-ORDER)
0 0 1 1 0	06	R/W	8	TOP OF WINDOW SOFT (HI-ORDER)
0 0 1 1 1	07	R/W	16	TOP OF WINDOW SOFT (LO-ORDER)
0 1 0 0 0	08	W	16	ATTRIBUTE FLAG
0 1 0 0 1	09	R/W	8	TOP OF PAGE HARD (HI)
0 1 0 1 0	0A	R/W	16	TOP OF PAGE HARD (LO)
0 1 0 1 1	0B	R/W	8	TOP OF WINDOW HARD (HI)
0 1 1 0 0	0C	R/W	16	TOP OF WINDOW HARD (LO)
1 0 0 0 0	10	W	16	DMA BURST
1 0 0 0 1	11	W	12	*VSYNC WIDTH/SCAN DELAY
1 0 0 1 0	12	W	12	*VERTICAL ACTIVE LINES
1 0 0 1 1	13	W	12	*VERTICAL TOTAL LINES
1 0 1 0 0	14	W	16	*HSYNC/VERTINT
1 0 1 0 1	15	W	9	*HDRIVE
1 0 1 1 0	16	W	9	*H SCAN DELAY
1 0 1 1 1	17	W	10	*H TOTAL COUNT
1 1 0 0 0	18	W	10	*H TOTAL DISPLAY

*These registers should only be accessed when Display Enable ("DE" bit in Mode Register 1) is reset, since they control the video timing signals.

Mode Register 1

ADDRESS: 0 0 0 0 0
READ/WRITE



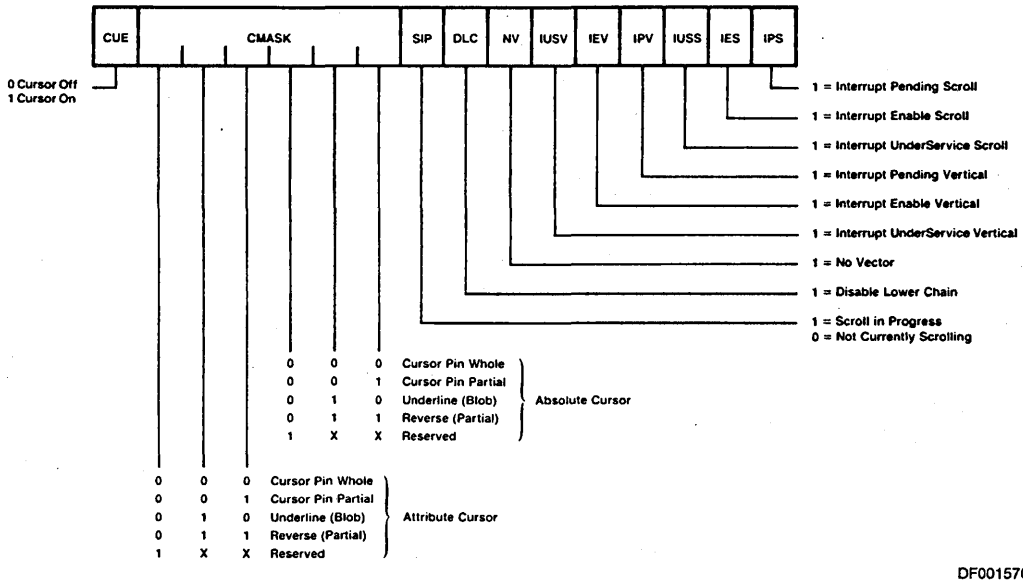
SK_1	SK_0	HSYNC, VSYNC and BLANK	AP_{0-10} and Cursor	CC_{0-7} and R_{0-4}
0	0	0	0	0
0	1	1	0	0
1	0	2	1	0
1	1	1	1	0

WS_1	WS_0	Wait State
0	0	No Wait State
0	1	$\overline{DS}$ Stretched by One Clock
1	0	$\overline{DS}$ Stretched by Two Clocks
1	1	Reserved

DF001561

Mode Register 2

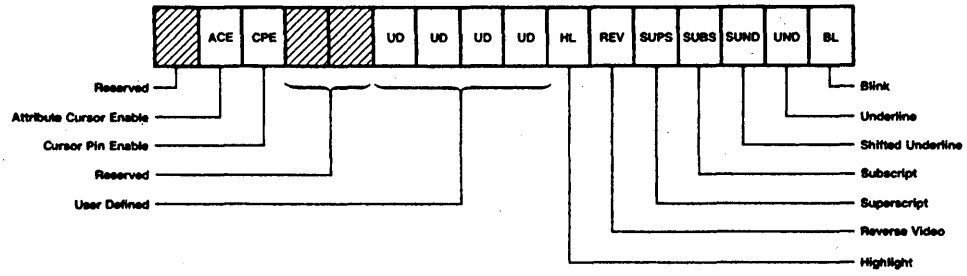
ADDRESS: 0 0 0 0 1
READ/WRITE



DF001570

Attribute Port Enable Register

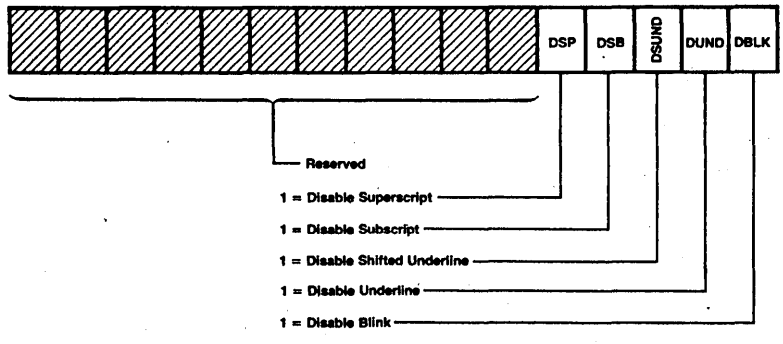
ADDRESS: 0 0 0 1 0
WRITE ONLY



DF001580

Attribute Redefinition Register

ADDRESS: 0 0 0 1 1
WRITE ONLY

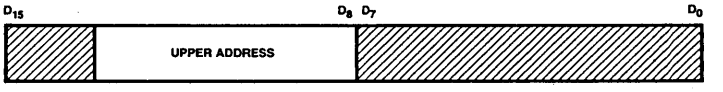


DF001590

Top of Page/Top of Window Registers L/$\overline{S}$ = 0

READ/WRITE

ADDRESS	REGISTER	ACTIVE BITS
0 0 1 0 0	Top of Page Soft (HI)	14 . . . 8
0 0 1 0 1	Top of Page Soft (LO)	15 . . . 0
0 0 1 1 0	Top of Window Soft (HI)	14 . . . 8
0 0 1 1 1	Top of Window Soft (LO)	15 . . . 0
0 1 0 0 1	Top of Page Hard (HI)	14 . . . 8
0 1 0 1 0	Top of Page Hard (LO)	15 . . . 0
0 1 0 1 1	Top of Window Hard (HI)	14 . . . 8
0 1 1 0 0	Top of Window Hard (LO)	15 . . . 0



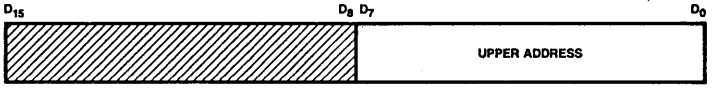
DF001600

DF001610

Top of Page/Top of Window Registers L/$\overline{S}$ = 1

READ/WRITE

ADDRESS	REGISTER	ACTIVE BITS
0 0 1 0 0	Top of Page Soft (HI)	7 . . . 0
0 0 1 0 1	Top of Page Soft (LO)	15 . . . 0
0 0 1 1 0	Top of Window Soft (HI)	7 . . . 0
0 0 1 1 1	Top of Window Soft (LO)	15 . . . 0
0 1 0 0 1	Top of Page Hard (HI)	7 . . . 0
0 1 0 1 0	Top of Page Hard (LO)	15 . . . 0
0 1 0 1 1	Top of Window Hard (HI)	7 . . . 0
0 1 1 0 0	Top of Window Hard (LO)	15 . . . 0

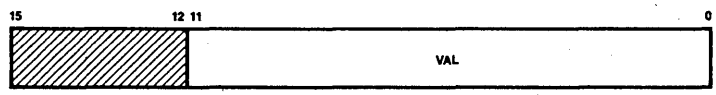


DF001620

DF001630

Vertical Active Lines Register

ADDRESS: 1 0 0 1 0
WRITE ONLY



VSYNC ↓ TO VBLANK ↑ = VAL+1 NON-INTERLACED
(VAL*+1)/2 INTERLACED

DF001640

* Must be odd

Vertical Total Lines Register

ADDRESS: 1 0 0 1 1
WRITE ONLY

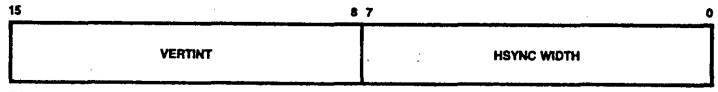
VSYNC TO VSYNC = VTOT+1 SCAN LINES NON-INTERLACED
= (VTOT**+1)/2 SCAN LINES INTERLACED

DF001651

** Must be even

Horizontal SYNC and Vertical Interrupt Row Register

ADDRESS: 1 0 1 0 0
WRITE ONLY



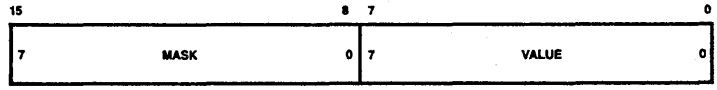
ROW NUMBER AT WHICH VERTICAL INTERRUPT OCCURS

IN NUMBER OF CLK_1 OR CLK_2 PERIODS DEPENDING ON $CLK_{1/2}$ IN MODE REGISTER 1

DF001661

Attribute Flag Register

ADDRESS: 0 1 0 0 0
WRITE ONLY

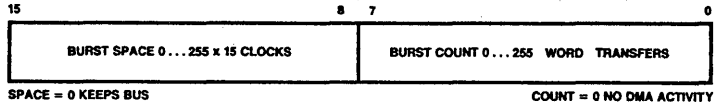


Note: When a mask-bit is set to 0, the corresponding value-bit must be 0.

DF001670

Burst Register

ADDRESS: 1 0 0 0 0
WRITE ONLY



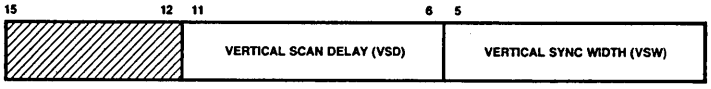
SPACE = 0 KEEPS BUS

COUNT = 0 NO DMA ACTIVITY

DF001681

Vertical SYNC Width/Vertical Scan Delay Register

ADDRESS: 1 0 0 0 1
WRITE ONLY



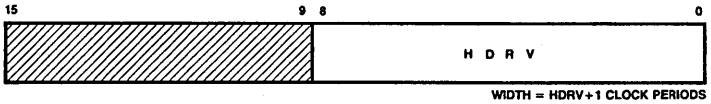
NON-INTERLACE DELAY = (VSD+1) SCAN LINES WIDTH = (VSW+1) SCAN LINES
INTERLACE DELAY = (VSD*+1)/2 SCAN LINES WIDTH = (VSW*+1)/2 SCAN LINES

*Must be odd

DF001690

Horizontal Drive Register

ADDRESS: 1 0 1 0 1
WRITE ONLY

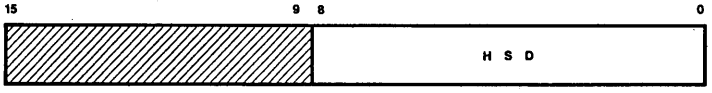


WIDTH = HDRV+1 CLOCK PERIODS

DF001700

Horizontal Scan Delay Register

ADDRESS: 1 0 1 1 0
WRITE ONLY

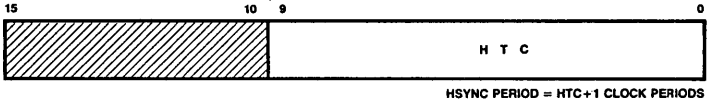


DELAY = HSD+1 CLOCK PERIODS

DF001710

Horizontal Total Count Register

ADDRESS: 1 0 1 1 1
WRITE ONLY

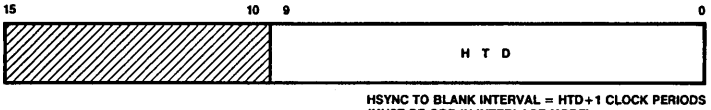


HSYNC PERIOD = HTC+1 CLOCK PERIODS

DF001721

Horizontal Total Display Register

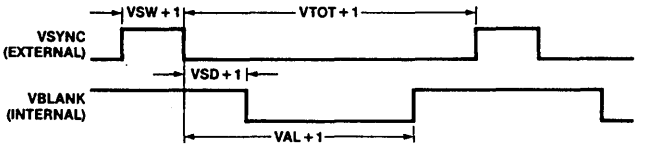
ADDRESS: 1 1 0 0 0
WRITE ONLY



HSYNC TO BLANK INTERVAL = HTD+1 CLOCK PERIODS
(MUST BE ODD IN INTERLACE MODE)

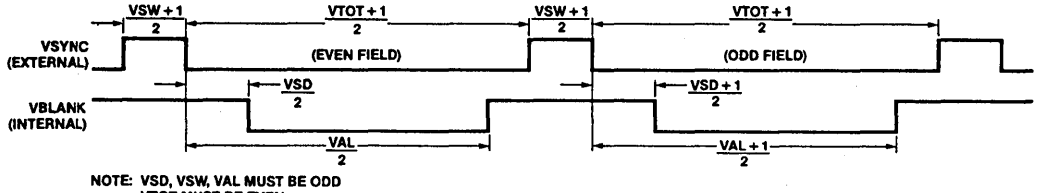
DF001730

Non-Interlaced Video Vertical Sync Timing



WF008910

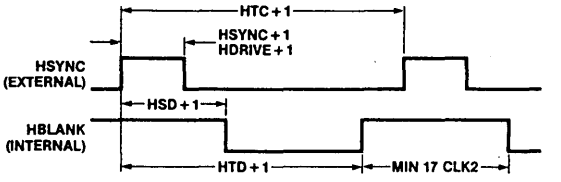
RFI and Video Interlace Sync Timing



NOTE: VSD, VSW, VAL MUST BE ODD
VTOT MUST BE EVEN

WF008920

Horizontal Sync Timing



WF008930

Note: HSD $\geq$ 6
Interlaced Video: HTC must be even.

FRAME TIMING SIGNALS SUMMARY:

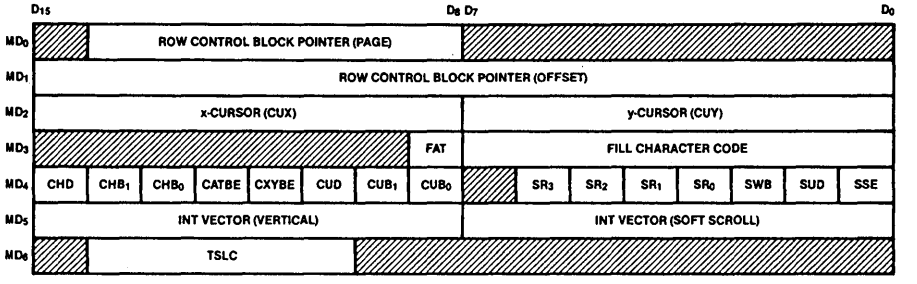
Non-Interlaced Mode

VERTICAL SYNC WIDTH	VSW + 1
FRONT PORCH (VBLANK R.E. TO VSYNC R.E.)	VTOT–VAL
BACK PORCH (VSYNC F.E. TO VBLANK F.E.)	VSD + 1
VSYNC F.E. TO NEXT VBLANK R.E.	VAL + 1
TOTAL SCAN LINES/FRAME-VSYNC WIDTH	VTOT + 1
HORIZONTAL SYNC WIDTH	HSYNC + 1
HORIZONTAL SYNC PERIOD	HTC + 1
HSYNC R.E. TO NEXT HBLANK R.E.	HTD + 1
HSYNC R.E. TO HBLANK F.E.	HSD + 1
HDRIVE R.E. TO HDRIVE F.E.	HDRV + 1

Interlaced Mode

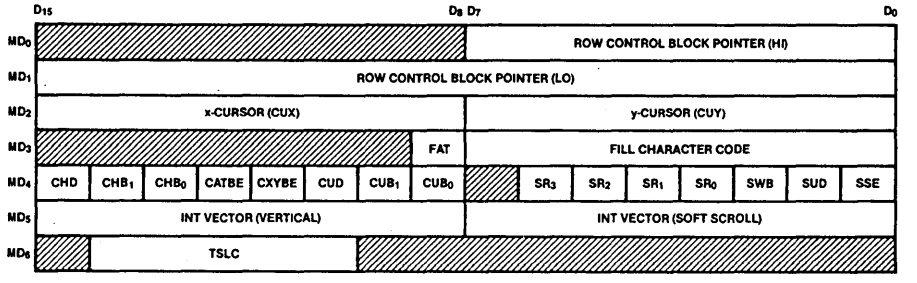
VERTICAL SYNC WIDTH	(VSW + 1)/2, VSW ODD	
BACK PORCH	VSD/2, EVEN FIELDV (VSD + 1)/2, ODD FIELD	VAL ODD
VSYNC F.E. TO NEXT VBLANK R.E.	(VAL + 1)/2, ODD FIELD VAL/2, EVEN FIELD	VSD ODD
TOTAL SCAN LINES/FRAME-VSYNC WIDTH	(VtoT + 1)/2, VTOT EVEN	
HORIZONTAL SYNC WIDTH	HSYNC + 1	
HORIZONTAL SYNC PERIOD	HTC + 1	
HSYNC R.E. TO NEXT HBLANK R.E.	HTD + 1, HTD ODD	
HSYNC E. TO HBLANK F.E.	HSD + 1	
HDRIVE R.E. TO HDRIVE F.E.	HDRV + 1	
FRONT PORCH (VBLANK R.E. TO VSYNC R.E.)	(VTOT-VAL)/2, EVEN FIELD (VTOT + 1-VAL)/2, ODD FIELD VAL ODD, VTOT EVEN	

Main Definition Block (L/$\overline{S}$ = 0)



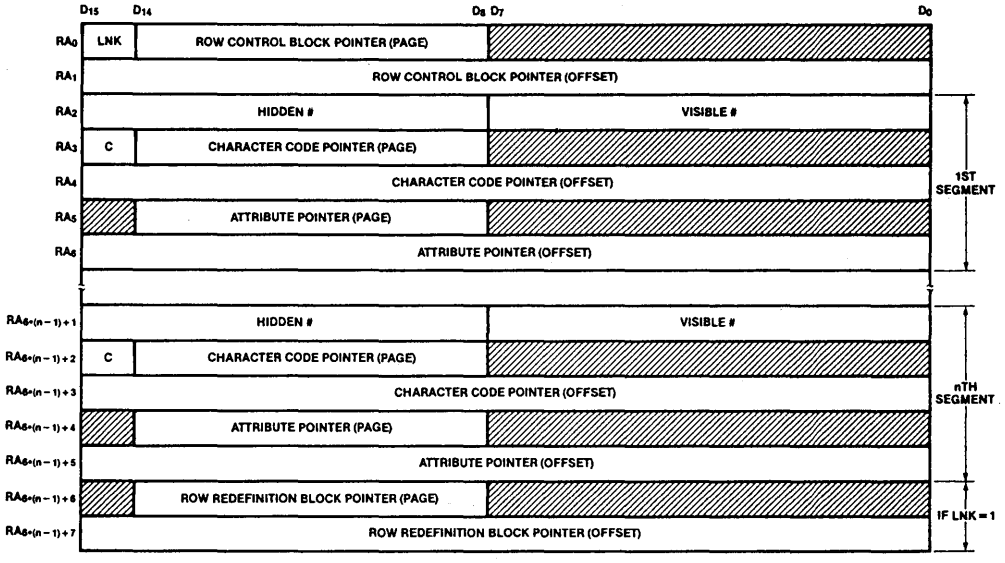
DF003990

Main Definition Block (L/$\overline{S}$ = 1)



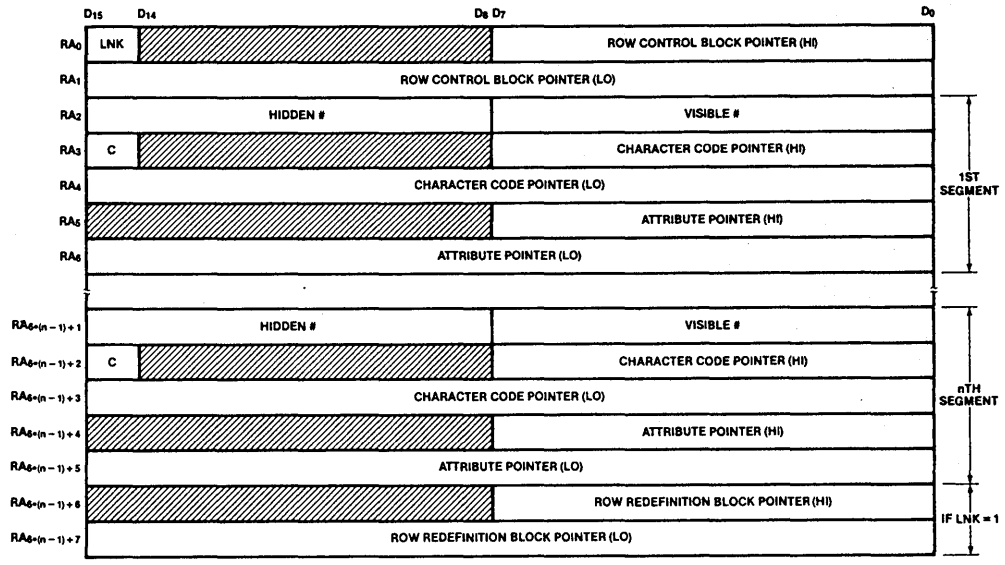
DF004000

Row Control Block (L/$\overline{S}$ = 0)



DF004010

Row Control Block (L/$\overline{S}$ = 1)



DF004020

3

Row Redefinition Block

	D_{15} D_{14}	D_{10} D_9	D_5 D_4	D_0
RR_0	TSLC	NCS		NCE
RR_1	ROW ATTRIBUTES (AP_{10}–AP_6)	SPCS		SPCE
RR_2	ROW ATTRIBUTES (AP_4–AP_0)	SBCS		SBCE
RR_3		CURS		CURE
RR_4	DR_1 DR_0	UND		SUND

DF004030

Window Definition Block (L/$\overline{S}$ = 0)

	D_{15}	D_{14} D_8 D_7 D_0
WD_0	SCW	WINDOW ROW CONTROL BLOCK POINTER (PAGE)
WD_1	WINDOW ROW CONTROL BLOCK POINTER (OFFSET)	
WD_2	0	WINDOW DEFINITION BLOCK POINTER (PAGE)
WD_3	WINDOW DEFINITION BLOCK POINTER (OFFSET)	
WD_4	START WINDOW ROW #	END WINDOW ROW #
WD_5	START WINDOW CHAR #	END WINDOW CHAR #

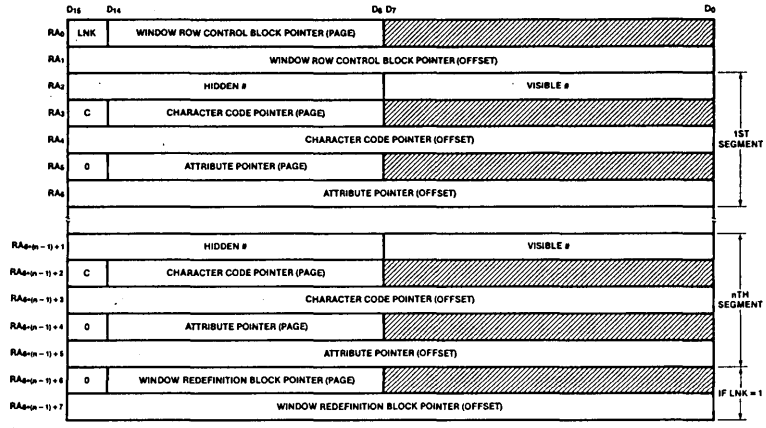
DF004040

Window Definition Block (L/$\overline{S}$ = 1)

	D_{15}	D_{14} D_8 D_7 D_0
WD_0	SCW	WINDOW ROW CONTROL BLOCK POINTER (HI)
WD_1	WINDOW ROW CONTROL BLOCK POINTER (LO)	
WD_2		WINDOW DEFINITION BLOCK POINTER (HI)
WD_3	WINDOW DEFINITION BLOCK POINTER (LO)	
WD_4	START WINDOW ROW #	END WINDOW ROW #
WD_5	START WINDOW CHAR #	END WINDOW CHAR #

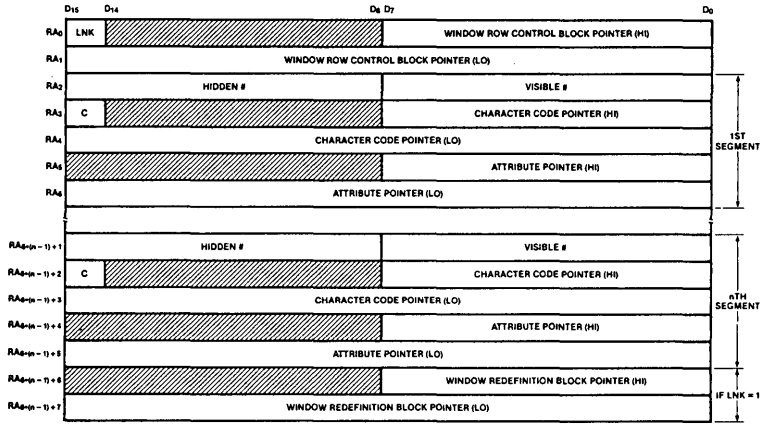
DF004050

Window Row Control Block (L/$\overline{S}$ = 0)



DF004070

Window Row Control Block (L/$\overline{S}$ = 1)



DF004080

Window Redefinition Block

	D15 … D10	D9	D5	D4	D0
WR0		NCS			NCE
WR1		SPCS			SPCE
WR2		SBCS			SBCE
WR3		CURS			CURE
WR4	DR1	DR0		UND	SUND

DF004090

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Supply Voltage (TTL)
with Respect to Ground ..... −0.5 to +7.0 V
Voltage on Any Input Pin
with Respect to Ground ..... −0.5 to +7.0 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V

Military (M) Devices
Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

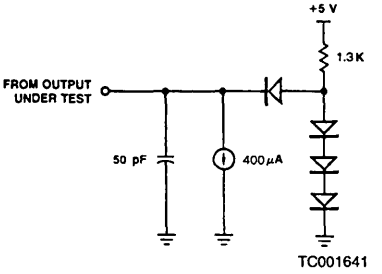
DC CHARACTERISTICS over operating ranges unless otherwise specified; Included in Group A, Subgroup 1, 2, 3, 7, 8 tests unless otherwise noted.

Parameter Symbol	Parameter Description	Min.	Max.	Units
V_{OH}	Output HIGH Voltage (I_{OH} = 400 μA)	2.4		V
V_{OL}	Output LOW Voltage (I_{OL} = 3.2 mA)		0.4	V
V_{IH}	Input HIGH Voltage (except CLK_1 and CLK_2)	2.0	V_{CC} +0.5 †	V
V_{CIH}	CLK_1/CLK_2 Input HIGH Voltage	4.0	V_{CC} +0.5 †	V
V_{IL}	Input LOW Voltage (except CLK_1 and CLK_2)	−0.5 †	0.8	V
V_{CIL}	CLK_1/CLK_2 Input LOW Voltage	−0.5 †	0.3	V
I_{IX}	Input Load Current (except $\overline{RSTI}$), $0 \leq V_{IN} \leq V_{CC}$		±10	μA
I_{IXR}	Input Load Current ($\overline{RSTI}$), $0 \leq V_{IN} \leq V_{CC}$		±100	μA
I_O	Output Leakage Current, $0.45\ V \leq V_{OUT} \leq V_{CC}$		±10	μA
I_{CC}	Supply Current		500	mA
C_{IN} †	Input Capacitance (all pins except CLK_1 and CLK_2), f = 1 MHz		15	pF
C_{CIN} †	Input Capacitance, CLK_1 and CLK_2, f = 1 MHz		80	pF
C_{OUT} †	Output Capacitance, f = 1 MHz		15	pF
$C_{I/O}$ †	Bidirectional Pin Capacitance, f = 1 MHz		20	pF

† Not included in Group A tests.

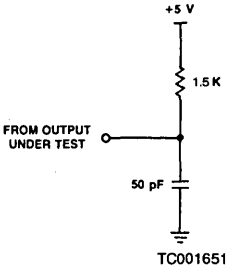
SWITCHING TEST CIRCUITS

Standard Test Load



TC001641

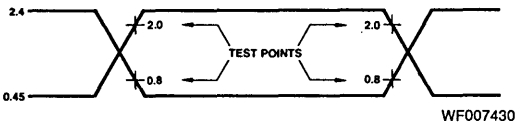
Open-Drain Test Load



TC001651

SWITCHING TEST WAVEFORM

Input Waveform



WF007430

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified; Included in Group A, Subgroup 9, 10, 11 tests unless otherwise noted.

Am8052 Bus Master Read/Write

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
1	t_{PHL}	CLK_1 ↑ to $\overline{AS}$ ↓		65		55		45	ns
2	t_{PLH}	CLK ↓ to $\overline{AS}$ ↑		65		55		45	ns
3	t_{PW}	$\overline{AS}$ Pulse Width	⑲ – 30		⑲ – 10		⑲ – 10		ns
4	t_S	Address Valid to $\overline{AS}$ ↑	⑲ – 50		⑲ – 35		⑲ – 35		ns
5	t_H	Address from $\overline{AS}$ ↑	20		20		20		ns
6	t_{PHL}	CLK_1 ↑ to $\overline{DS}$ ↓		80		65		45	ns
7	t_S	Data In to CLK_1 ↓	20		15		10		ns
8	t_H	Data In from $\overline{DS}$ ↑	0		0		0		ns
9	t_{PLH}	CLK_1 ↓ to $\overline{DS}$ ↑		80		65		45	ns
10	t_{PLH}	CLK_1 ↑ to R/$\overline{W}$	0†	65	0†	55	0	45	ns
11	t_H	CLK_1 ↓ to $\overline{DREN}$ ↑ (Note 2)		70		45		40	ns
12	t_S	$\overline{WAIT}$ Valid to CLK_1 ↓	20		15		10		ns
13	t_H	$\overline{WAIT}$ from CLK_1 ↓	30		20		20		ns
14	t_{PHL}	CLK_1 ↓ to $\overline{DREN}$ ↓		65		55		45	ns
17	t_{PHL}	CLK_1 ↑ to $\overline{DTEN}$ ↓		65		55		45	ns
18	t_{PLH}	CLK_1 ↑ to $\overline{DTEN}$ ↑		65		55		45	ns
19	t_{PW}	CLK_1 HIGH Pulse Width	100	500	70	500	50	500	ns
20	t_{PW}	CLK_1 LOW Pulse Width	100	500	70	500	50	500	ns
40	t_{CYC}	CLK_1 Period	250	1000	165	1000	125	1000	ns
41	t_{AVDV}†	Address Valid to Data In (Note 1)							ns
42	t_{ASDV}†	$\overline{AS}$ ↑ to Data Valid (Note 1)							ns
43	t_{DSDV}†	$\overline{DS}$ ↓ to Data Valid (Note 1)							ns
46	t_{DRT}	$\overline{DREN}$ ↑ to $\overline{DTEN}$ ↓	20		20		20		ns
48	t_H	Data In from $\overline{DREN}$ ↑	0		0		0		ns

Notes: 1. ㊶, ㊷, and ㊸ can be computed with the following equations, but are not tested:

㊶ = 2·㊵ + ⑲ – ① – ③ + ④ – ⑦

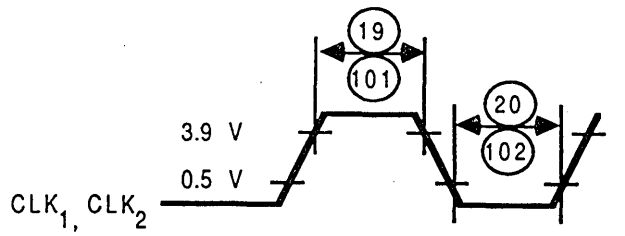
㊷ = 2·㊵ – ② – ⑦ – (CLK_1 Fall time)

㊸ = ㊵ + ⑲ – ⑥ – ⑦

2. This parameter specifies when the Am8052 stops driving $\overline{DREN}$ (open drain) LOW.
3. In the following diagrams (Switching Waveforms), O.D. designates an open-drain output which has turned off and is being pulled up by an external load.

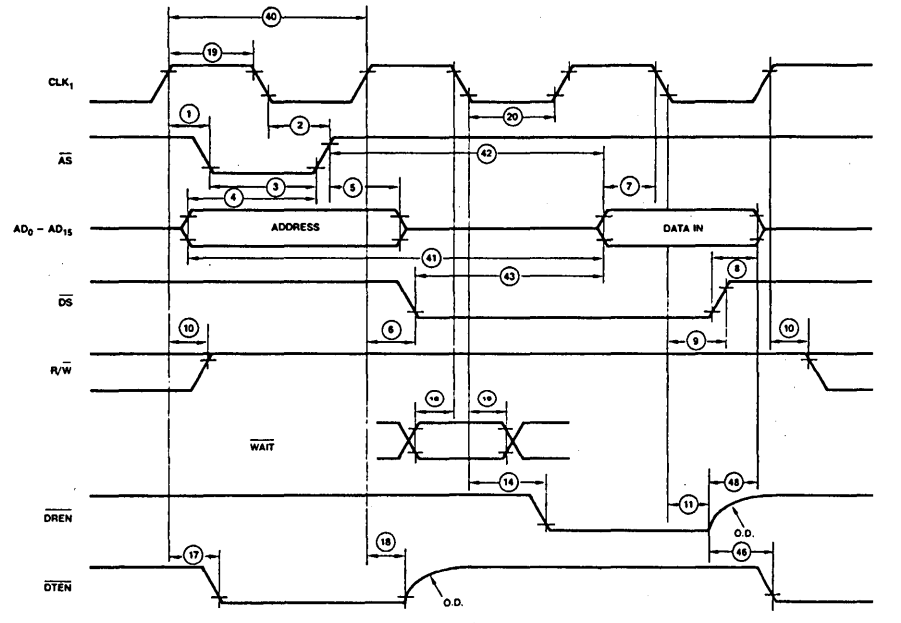
† Not included in Group A tests.

* Commercial products only.

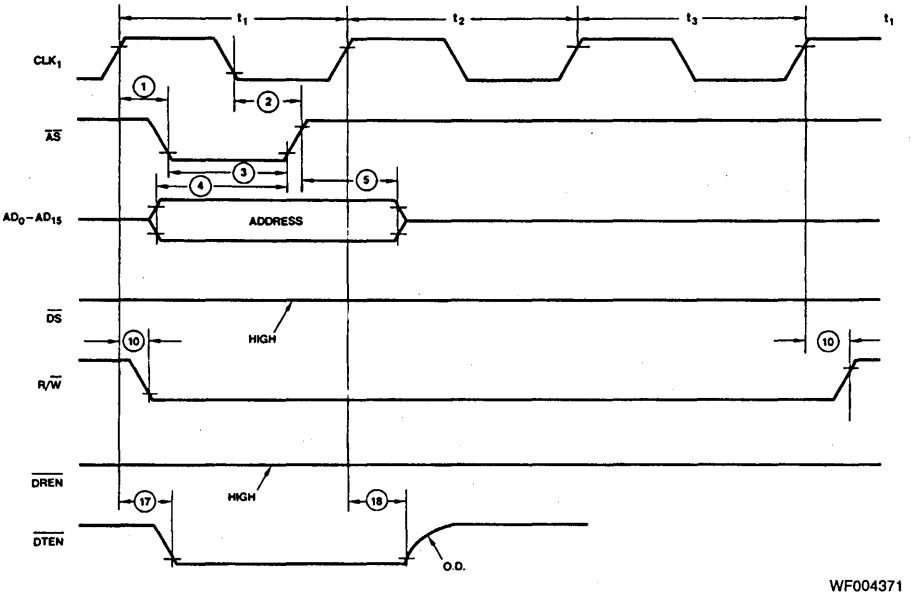


WF023540

Clock



Am8052 Bus Master Read



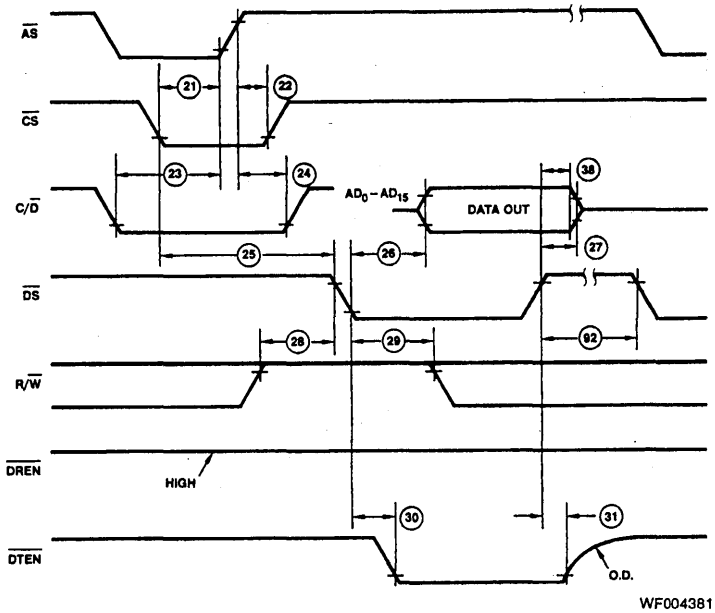
Am8052 Bus Master Write

SWITCHING CHARACTERISTICS (Cont'd.)
Am8052 Bus Slave Read Latched

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
21	t_S	$\overline{CS}$ ↓ to $\overline{AS}$ ↑	0		0		0		ns
22	t_H	$\overline{CS}$ LOW from $\overline{AS}$ ↑	30		25		20		ns
23	t_S	C/$\overline{D}$ to $\overline{AS}$ ↑	0		0		0		ns
24	t_H	C/$\overline{D}$ from $\overline{AS}$ ↑	30		25		20		ns
25	t_{PD}	$\overline{CS}$ ↓ to $\overline{DS}$ ↓	50		40		30		ns
26	t_{DSDV}	$\overline{DS}$ ↓ to Data Valid		180		180		150	ns
27	t_H	Data Valid from $\overline{DS}$ ↑	15		15		10		ns
28	t_S	R/$\overline{W}$ to $\overline{DS}$ ↓	10		10		10		ns
29	t_H	R/$\overline{W}$ Valid from $\overline{DS}$ ↓	50		40		40		ns
30	t_{PD}	Delay from $\overline{DS}$ ↓ to $\overline{DTEN}$ ↓		65		55		45	ns
31	t_{PD}	Delay from $\overline{DS}$ ↑ to $\overline{DTEN}$ ↑		65		55		45	ns
38	t_Z	$\overline{DS}$ ↑ to AD_0–AD_{15} HI-Z (Note 4)	10	70	10	60	10	50	ns

Notes: 1. R/$\overline{W}$ latched internally by $\overline{DS}$ ↓.
2. $\overline{CS}$ latched internally by $\overline{AS}$ ↑.
3. C/$\overline{D}$ latched internally by $\overline{AS}$ ↑.
4. This parameter specifies when the Am8052 stops driving AD_0–AD_{15}.

* Commercial products only.



WF004381

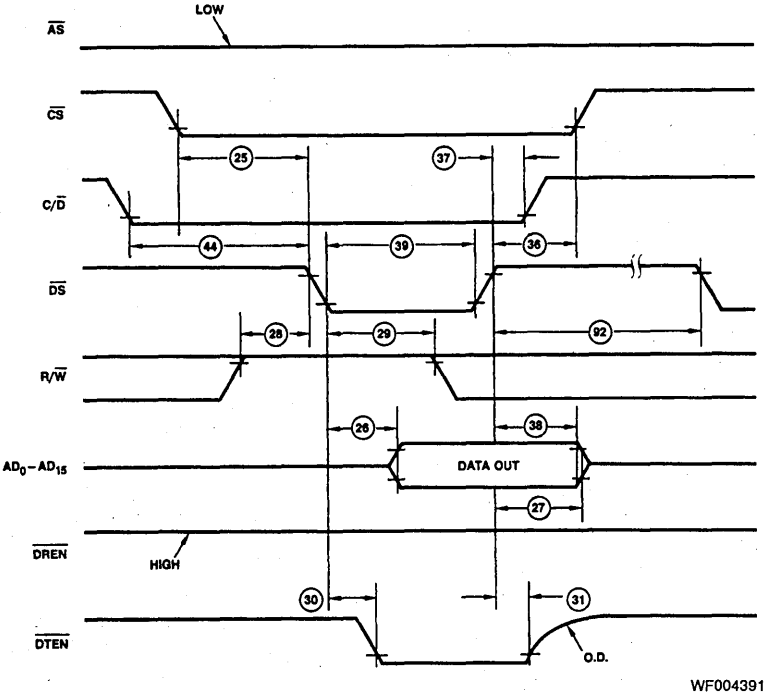
Am8052 Bus Slave Read Latched

SWITCHING CHARACTERISTICS (Cont'd.)

Am8052 Bus Slave Read Unlatched

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
36	t_H	$\overline{CS}$ LOW from $\overline{DS}$ ↑	10		7		5		ns
37	t_H	C/$\overline{D}$ LOW from $\overline{DS}$ ↑	10		7		5		ns
39	t_{PW}	$\overline{DS}$ ↓ to DS ↑ Read	250		200		150		ns
44	t_S	C/$\overline{D}$ to $\overline{DS}$ ↓	50		40		30		ns
92	t_{SRT}	Slave Recovery Time	500		300		225		ns

* Commercial products only.



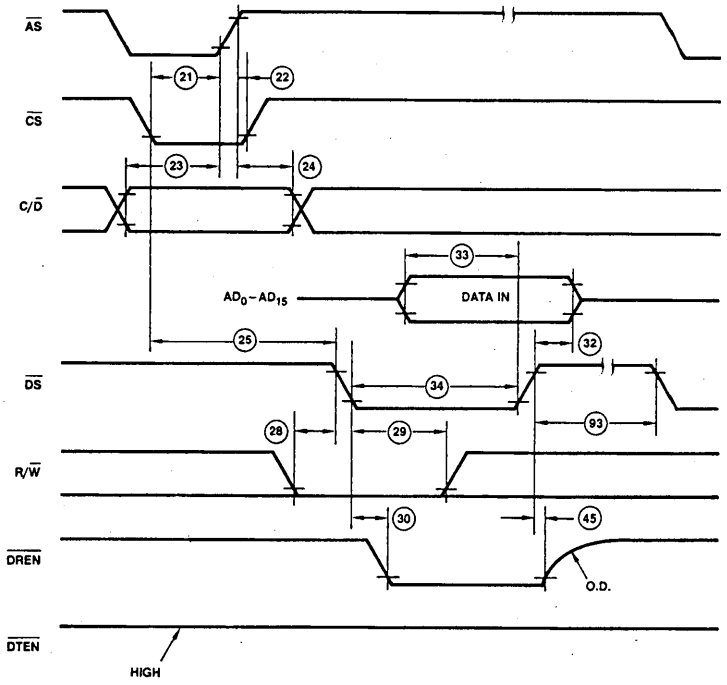
WF004391

Am8052 Bus Slave Read Unlatched

SWITCHING CHARACTERISTICS (Cont'd.)
Am8052 Bus Slave Write Latched

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
32	t_H	Data In Valid from $\overline{DS}$ ↑	20		20		20		ns
33	t_S	Data In Valid to $\overline{DS}$ ↑	100		90		80		ns
34	t_{PW}	$\overline{DS}$ Pulse Width	160		135		125		ns
45	t_H	Delay from $\overline{DS}$ ↑ to $\overline{DREN}$ ↑	20	80	20	70	20	70	ns

* Commercial products only.



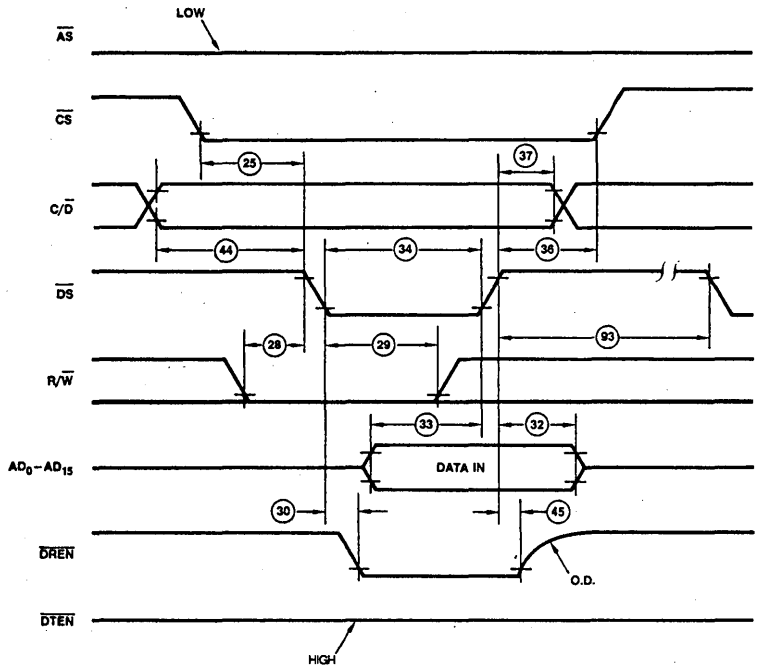
WF004401

Am8052 Bus Slave Write Latched

SWITCHING CHARACTERISTICS (Cont'd.)
Am8052 Bus Slave Write Unlatched

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
93	t_{SCT}	Slave Recovery Time	590		365		250		ns

* Commercial products only.



WF004411

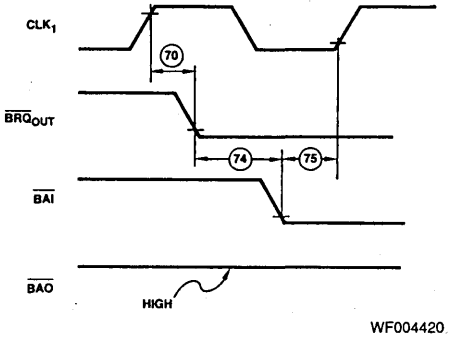
Am8052 Bus Slave Write Unlatched

SWITCHING CHARACTERISTICS (Cont'd.)
Am8052 Bus Exchange

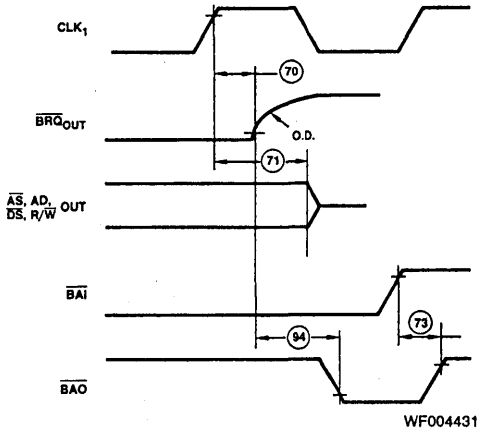
No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
70	t_{PD}	CLK_1 ↑ to $\overline{BRQ}_{OUT}$		130		115		100	ns
71	t_{PZ} †	CLK_1 ↑ to Float (Note 1)		180		160		140	ns
73	t_{PD}	$\overline{BAI}$ to $\overline{BAO}$		60		50		40	ns
74	t_{PD}	$\overline{BRQ}$ ↓ to $\overline{BAI}$ ↓ Delay	0		0		0		ns
75	t_S †	$\overline{BAI}$ ↓ to CLK_1 ↑ (Note 2)	60		50		40		ns
94	t_{PD}	$\overline{BRQ}$ ↑ to $\overline{BAO}$ ↓		70		60		50	ns

Notes: 1. This parameter specifies when the Am8052 stops driving $\overline{AS}$, AD, $\overline{DS}$, and R/$\overline{W}$.
2. This parameter for testing only. For normal operation, this signal may be asynchronous to the clock.
† Not included in Group A tests.
* Commercial products only.

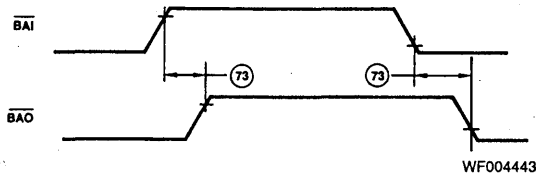
Am8052 Bus Exchange



Requesting



Releasing



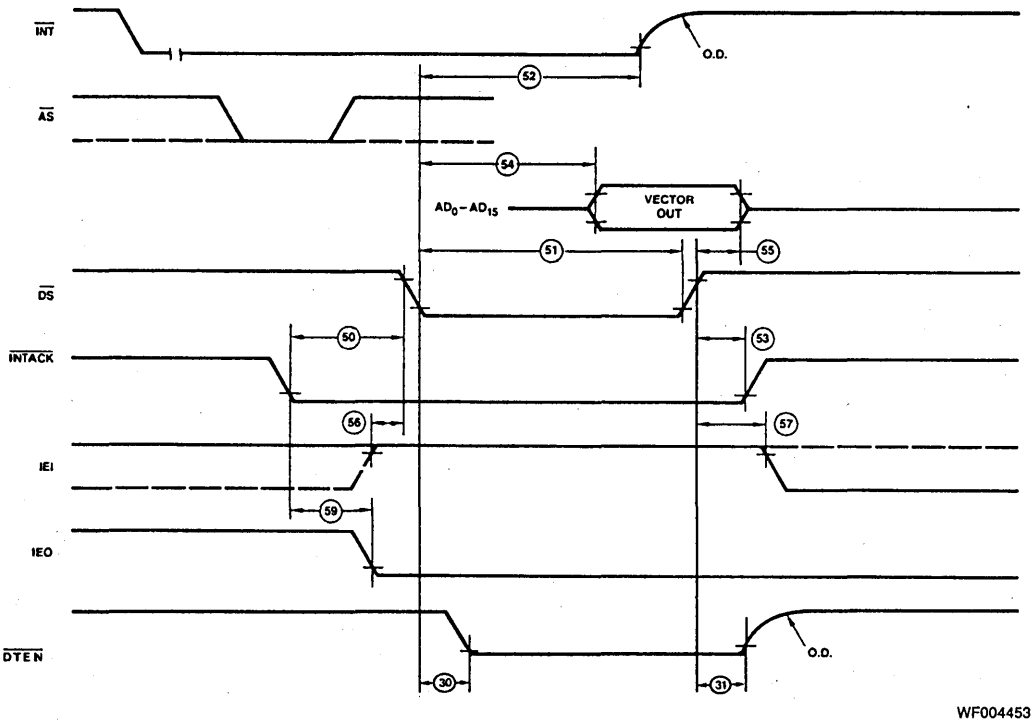
Chain Delay

SWITCHING CHARACTERISTICS (Cont'd.)

Am8052 Interrupt ACK Timing – Device Acknowledged

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
50	t_S	$\overline{INTACK}$ to $\overline{DS}$ ↓	260		230		200		ns
51	t_{PW}	$\overline{DS}$ ↓ to $\overline{DS}$ ↑ ACK	250		200		150		ns
52	t_{PD}	$\overline{DS}$ ↓ to $\overline{INT}$ ↑		260		230		200	ns
53	t_H	$\overline{INTACK}$ from $\overline{DS}$ ↑	0		0		0		ns
54	t_D	$\overline{DS}$ ↓ to Vector Valid		210		180		150	ns
55	t_H	Vector from $\overline{DS}$ ↑	0		0		0		ns
56	t_S	IEI to $\overline{DS}$ ↓	100		90		80		ns
57	t_H	IEI from $\overline{DS}$ ↑	0		0		0		ns
59	t_D	$\overline{INTACK}$ to IEO ↓ (IEI = H)		200		170		150	ns

* Commercial products only.



WF004453

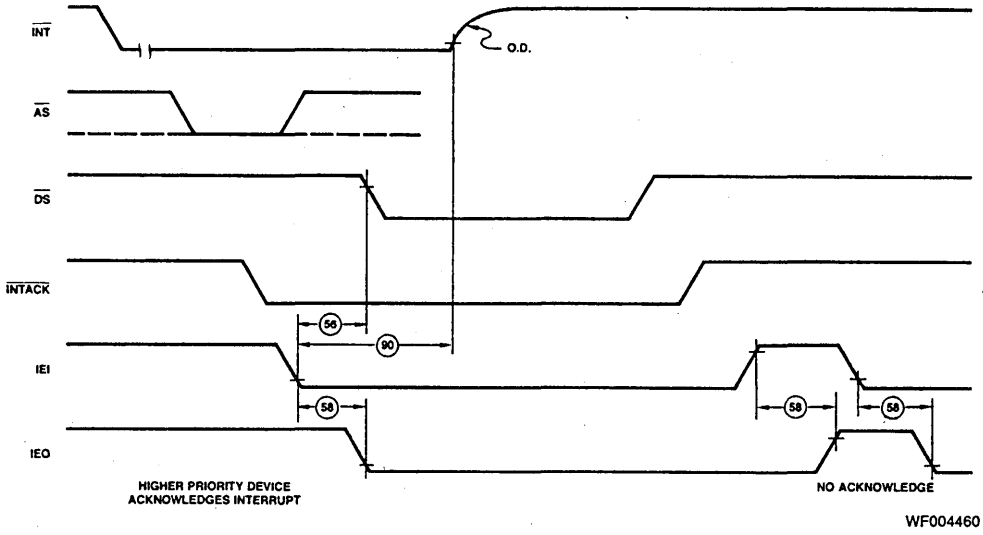
Am8052 Interrupt ACK Timing – Device Acknowledged

SWITCHING CHARACTERISTICS (Cont'd.)

Am8052 Interrupt ACK Timing – Low Priority

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
56	t_S	IEI to $\overline{DS}$ ↓	100		90		80		ns
58	t_D	IEI to IEO		100		90		80	ns
90	t_D	IEI ↓ to $\overline{INT}$ ↑ (Note 1)		100		90		80	ns

Notes 1. $\overline{INT}$ terminated by an acknowledge higher on chain.
* Commercial products only.



WF004460

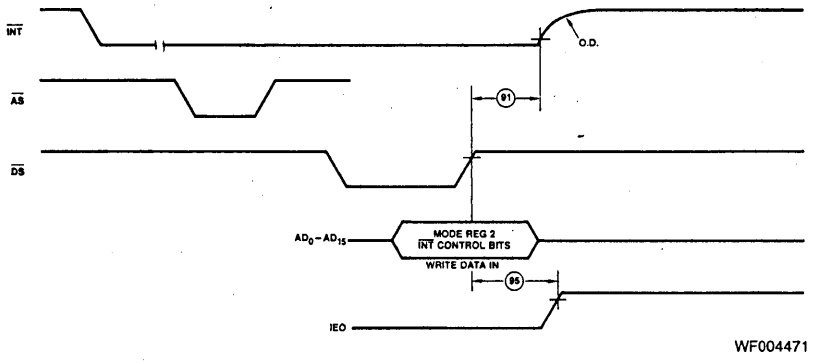
Am8052 Interrupt ACK Timing–Low Priority

SWITCHING CHARACTERISTICS (Cont'd.)

Am8052 Non-Vectored $\overline{INT}$ Timing

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
91	t_D	$\overline{DS}$ ↑ to $\overline{INT}$ (Write) (Note 1)		100		90		80	ns
95	t_D	$\overline{DS}$ ↑ to IEO ↑ (Write) (Note 2)		100		90		80	ns

Notes: 1. This parameter describes the termination of an interrupt request via a write to the appropriate bit in Mode Register 2:
IUSS → 1 IUSV → 1
IES → 0 IEV → 0
IPS → 0 IPV → 0
2. This is the release of IEO LOW due to the slave mode reset of the IUS bit in Mode Register 2.
* Commercial products only.



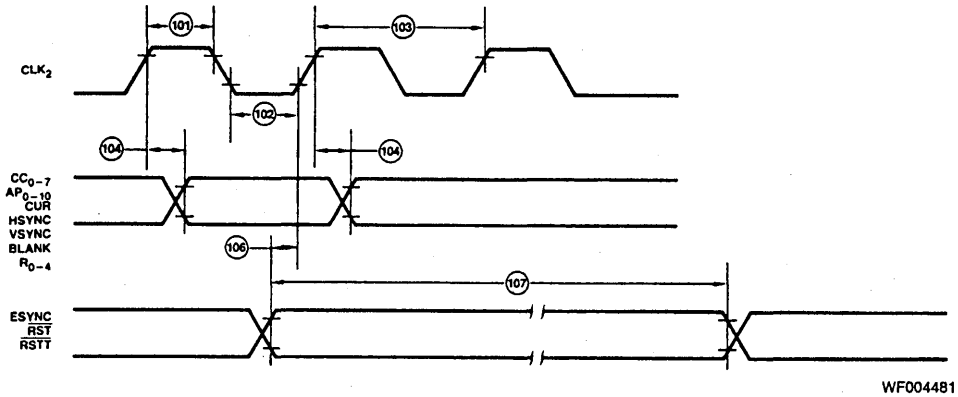
WF004471

Am8052 Non-Vectored $\overline{INT}$ Timing

SWITCHING CHARACTERISTICS (Cont'd.)
Am8052 Video Outputs and Synchronizing Input Timing

No.	Parameter Symbol	Parameter Description	4 MHz		6 MHz		8 MHz*		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
101	t_{PW}	CLK_2 HIGH Pulse Width	100	500	70	500	35	500	ns
102	t_{PW}	CLK_2 LOW Pulse Width	100	500	70	500	35	500	ns
103	t_{CYC}	CLK_2 Period	250	1000	165	1000	100	1000	ns
104	t_{DC}	CLK_2 ↑ to Output Delay (Note 3)		80		55		50	ns
106	t_S †	Input Setup to CLK_2 ↑ (Note 1)	70		60		50		ns
107	t_W	Input Pulse Width (Note 2)	5T		5T		5T		ns

Notes: 1. Parameter 106 is specified for test purposes only. For normal operation, these signals may be Asynchronous to the clock.
2. Parameter 107 is for reset only. T = CLK_2 period.
3. For HSYNC, VSYNC, and BLANK parameter 104 specifies output delay to CLK_1 or CLK_2 (see Mode Register Description).
† Not included in Group A tests.
* Commercial products only.



WF004481

Am8052 Video Outputs and Synchronizing Input Timing

Am8152A/Am8152B

Video System Controller (VSC)

FINAL

DISTINCTIVE CHARACTERISTICS

- Am8152A with Video Dot Clock Rate up to 80 MHz
  Am8152B with Video Dot Clock Rate up to 40 MHz
- Four-level current driven (75Ω) differential video output
- Digital Video output
- On-board crystal driven oscillator
- Proportional Spacing Support (2–17 dots)
- 9-bit dot data parallel input, with expansion capability to seventeen bits
- Trailing blanks (0–3 dots)
- Double Width Characters
- Attribute Support: Character Blink, Underline, Overstrike, Reverse, and Highlight
- Buffered and Synchronized Character Clock Outputs
- Background color selection
- Buffered and Synchronized Vertical and Horizontal Sync Outputs

GENERAL DESCRIPTION

The Am8152A/Am8152B Video System Controller (VSC) provides the interface between a CRT controller and a CRT monitor. The basic chip functions are:

- Support proportional and non-proportional character display
- Correctly synchronize and mix character attributes with video signals
- Output the video information in a four-level analog or digital format
- Serialize parallel video data

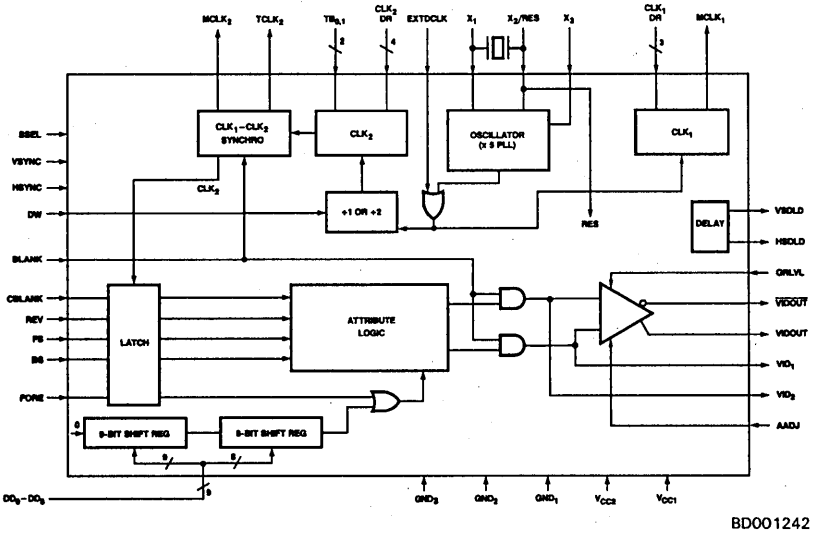
The VSC consists of a parallel-to-serial converter which provides a video bit stream to on-chip attribute logic. This logic, under control of the attribute inputs, operates on the bit stream to generate grey scale video. Video outputs from the VSC are of two forms — analog and digital. The digitally encoded outputs implement four video levels: Blank, Black, Grey and White. Identical information is available in analog form via differential outputs (current driven) into a nominal 75Ω impedance.

The Am8152A/Am8152B also supports proportional spacing using a bit width programmable character clock. Character ROM pixel information is selectable from two to seventeen pixels per character. Up to three blank pixels can be appended to the character ROM input thereby facilitating right justification of text.

The difference between the Am8152A and the Am8152B is that the Am8152A operates up to 80 MHz, while the Am8152B operates up to 40 MHz. When using the PLL, the lower operating frequency limit is 20 MHz.

The Am8152A/Am8152B is fabricated using AMD's advanced bipolar process with internal ECL logic. The device is available in conventional 48-pin dual in-line package as well as the surface mounted 68-pin PLCC package.

BLOCK DIAGRAM

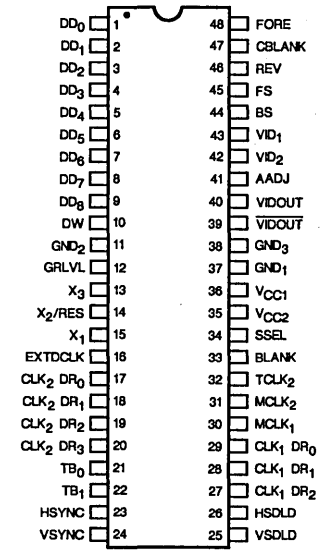


BD001242

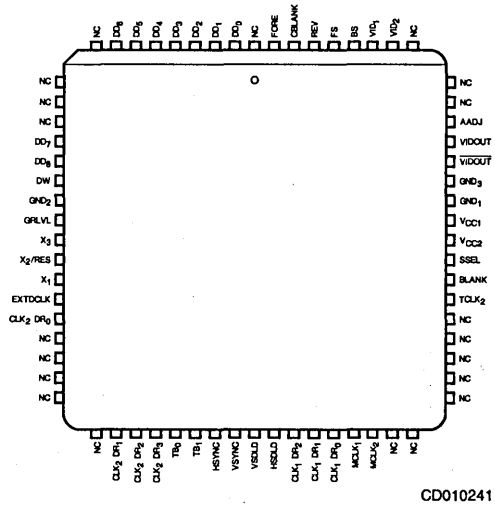
Figure 1.

Publication # 03888 Rev. D Amendment /0
Issue Date: October 1986

CONNECTION DIAGRAM
Top View



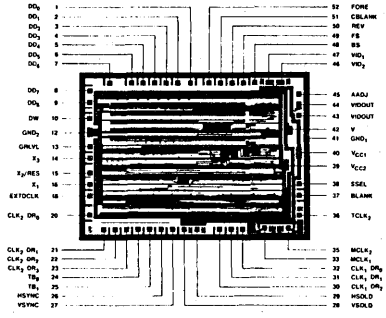
CD001512



CD010241

Note: Pin 1 is marked for orientation

METALLIZATION AND PAD LAYOUT

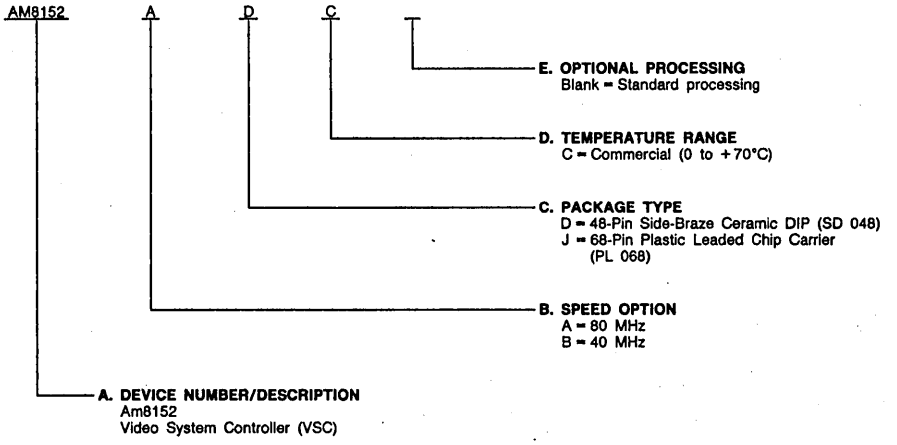


ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

A. Device Number
B. Speed Option (if applicable)
C. Package Type
D. Temperature Range
E. Optional Processing



Valid Combinations	
AM8152A AM8152B	DC, JC

Valid Combinations

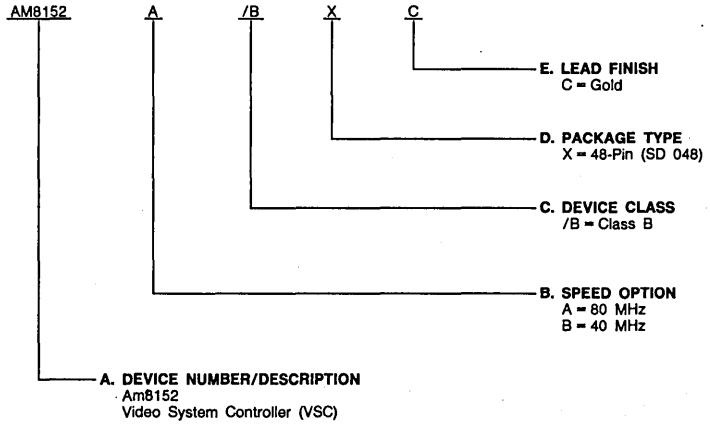
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. CPL (Controlled Products List) products are processed in accordance with MIL-STD-883C, but are inherently non-compliant because of package, solderability, or surface treatment exceptions to those specifications. The order number (Valid Combination) for APL products is formed by a combination of: **A. Device Number**

B. Speed Option (if applicable)
C. Device Class
D. Package Type
E. Lead Finish



Valid Combinations	
AM8152A	/BXC
AM8152B	/BXC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consists of Subgroups:
1, 2, 3, 7, 8, 9, 10, 11

3

PIN DESCRIPTION

$MCLK_1$ $Clock_1$ (non-TTL compatible) (Output)
$MCLK_1$ is a system clock. It is intended to drive the Am8052 CLK_1. $MCLK_1$ output is nominally a square wave divided down from the internal dot clock frequency according to the CLK_1 DR DR_{0-2} (CLK_1 Divide Ratio) inputs.

CLK_1 DR $Clock_1$ Divide Ratio (Input)
CLK_1 DR DR_{0-2} are three inputs which control the $MCLK_1$ divide ratio. The three inputs may be programmed to divide the $MCLK_1$ signal by two, four, six, ..., sixteen. (See Table 1)

$MCLK_2$ $Clock_2$ (non-TTL compatibe) (Output)
$MCLK_2$ is the character clock. Its function is to drive the Am8052 CLK_2. $MCLK_2$ output is a nominal square wave divided down from the internal dot clock frequency according to the sum of the CLK_2, DR_{0-3} and TB_{0-1} inputs.

CLK_2 DR $Clock_2$ Divide Ratio (Input)
CLK_2 DR DR_{0-3} are four inputs which control an internal divider to divide the dot clock frequency by a value from two to seventeen. The TB inputs are added. (See Table 2)

$TCLK_2$ TTL $Clock_2$ (Output)
$TCLK_2$ is a TTL-compatible version of $MCLK_2$.

X_1, X_2/RES X_1, X_2/RESET (X_2 is non-TTL compatible, reset is TTL compatible) (Input)
X_1, X_2/RES are the external crystal inputs when the on-chip oscillator of the VSC is being used. The external crystal frequency is multiplied by five to produce the on-chip dot clock. If the external dot clock flow-through mode is used, the X_1 should be tied Low and X_2/RES may be used as a reset input to synchronize multiple VSCs. Note that the reset signal must be synchronous to the external dot clock.

X_3 X_3 (non-TTL compatible) (Input)
X_3 is used as an input to the on-chip voltage-controlled oscillator. When the on-chip oscillator of VSR is being used, X_3 should be connected to ground by an appropriate capacitor. If the external dot clock flow-through mode is used, X_3 and X_1 should be tied to ground.

VSYNC Vertical Sync (Input)
VSYNC is an input that must be synchronous to either $MCLK_1$ or $MCLK_2$, dependent on the SSEL input. If SSEL is High, VSYNC must be synchronous to $MCLK_1$.

VSDLD Vertical Sync Delayed (Output)
VSDLD is the delayed output of VSYNC, synchronous to $MCLK_1$ or $MCLK_2$, depending on the setting of SSEL.

HSYNC Horizontal Sync (Input)
HSYNC is an input that must be synchronous to either $MCLK_1$ or $MCLK_2$, dependent upon the SSEL input. If SSEL is Low, HSYNC must be synchronous to $MCLK_2$; if SSEL is High, HSYNC must be synchronous to $MCLK_1$.

HSDLD Horizontal Sync Delayed (Output)
HSDLD is the delayed output of HSYNC, synchronous to $MCLK_1$ or $MCLK_2$, depending upon the setting of SSEL.

SSEL Sync Select (Input)
The SSEL line determines if the VSYNC, HSYNC and BLANK are going to be synchronized to the $MCLK_1$ or $MCLK_2$ signals. A High on SSEL also will resynchronize $MCLK_2$ and $MCLK_1$ during blanking. SSEL is kept LOW for applications which do not involve proportional spacing and HIGH for proportional spacing.

BLANK Blank (Input)
BLANK is an input normally synchronous to $MCLK_1$, although it may be synchronous to $MCLK_2$ in non-proportional spacing applications. The active pulse width of BLANK will usually overlap the active portions of HSYNC and VSYNC. While BLANK is active, $TCLK_2$/$MCLK_2$ may be forced to synchronize to the $MCLK_1$ clock. When BLANK goes inactive, the rising edges of $MCLK_1$ and $TCLK_2$/$MCLK_2$ will be synchronized in order to prevent "dot walk" in proportional spacing applications. BLANK active also forces the video output level to "blank" regardless of DD, FORE or other inputs.

CBLANK Character Blank (Input)
CBLANK forces video output levels (VID_1, VID_2, VIDOUT and $\overline{\text{VIDOUT}}$) to switch to the background color level.

FORE Foreground Video (Input)
The FORE video input is "OR'ed" with the dot data output by the parallel-to-serial shift register to switch to the foreground color level (e.g., to implement underlines). FORE is latched with CLK_2 and cannot be used to insert serial data.

REV Reverse (Input)
The REV input causes the foreground color levels to be transposed with the background color level for the total character period (including any trailing blanks).

FS Foreground Shift (Input)
The FS input causes the shift in the video output levels to produce a highlight effect. (See Table 3.)

TB_0, TB_1 Trailing Blanks (Input)
The TB inputs are added to the CLK_2, DR_{0-3} inputs to calculate the total period (in DOT Clock periods) of $MCLK_2$. If the total period is greater than 17, then zeroes will be shifted out of the shift register. If the total period is less than 17 dot clocks, the user must insure that the shift register is filled with zeroes. The maximum CLK_2 period is 19 dot clock periods; therefore, the combination of CLK_2 DR = 17 and TB = 3 is not allowed. The first character after video blanking must be four or more pixels, trailing blanks included, or an extra pixel will be outputted.

DD_0 – DD_8 Dot Data (Inputs)
The DD inputs accept parallel character dot matrix information for serial conversion for video output. DD_0 is shifted out first.

BS Background Select (Input)
The BS input specifies the color level of the background video. This input can be overridden by BLANK active.

VIDOUT, $\overline{\text{VIDOUT}}$ Video Output (non-TTL compatible)
VIDOUT and $\overline{\text{VIDOUT}}$ outputs in a differential mode the composite blank and video dot levels to a nominal 75Ω load impedance from switched current sources.

VID_1, VID_2 Video Digital (Output)
VID_1 and VID_2 are digitally encoded outputs of the video out. VID_1 is the least significant bit. Encoding is as follows:

	VID_2 (VIDEO)	VID_1 (HIGHLIGHT)
Blank Level	0	0
Black	0	1
Grey	1	0
White	1	1

GRLVL Grey Level (Input)
The GRLVL input adjusts the current level output, via the VIDOUT and $\overline{\text{VIDOUT}}$ outputs, of the grey video level. There are two pre-selected grey levels: for GRLVL HIGH, grey is brighter; for GRLVL LOW, grey is darker. GRLVL is not latched with $MCLK_2$ but is sampled on a pixel-by-pixel basis.

DW Double Width (Input)

The DW input, when active HIGH, causes the dot clock supplied to the $TCLK_2/MCLK_2$ clock divide circuitry and the video shift register to be divided by two. This function is used to facilitate doubling the width of a character cell matrix in the horizontal direction. The trailing blank information is also widened during a double width character.

EXTDCLK External Dot Clock (Input)

EXTDCLK is an external, TTL-compatible dot clock input for use in multiple Am8152A/Am8152B configurations. This signal replaces the internal oscillator function. To enable EXTDCLK, both X_1 and X_3 must be grounded.

AADJ Analog Outputs Current Adjust (non-TTL compatible) (Inputs)

Analog output current adjust is used for setting the analog video output current to 13.3 mA. This is done by connecting AADJ to GND_3 via an applicable 1% resistor.

V_{CC1}, V_{CC2}

V_{CC} should be connected to +5 V.

GND_1, GND_2, GND_3

GND_{1-3} should be connected to Ground.

CLK_1 DR 2	1	0	A	B
L	L	L	1	1
L	L	H	2	2
L	H	L	3	3
L	H	H	4	4
H	L	L	5	5
H	L	H	6	6
H	H	L	7	7
H	H	H	8	8

Table 1

Note: A, B are measured in EXTDCLK periods.

CLK_2 Period

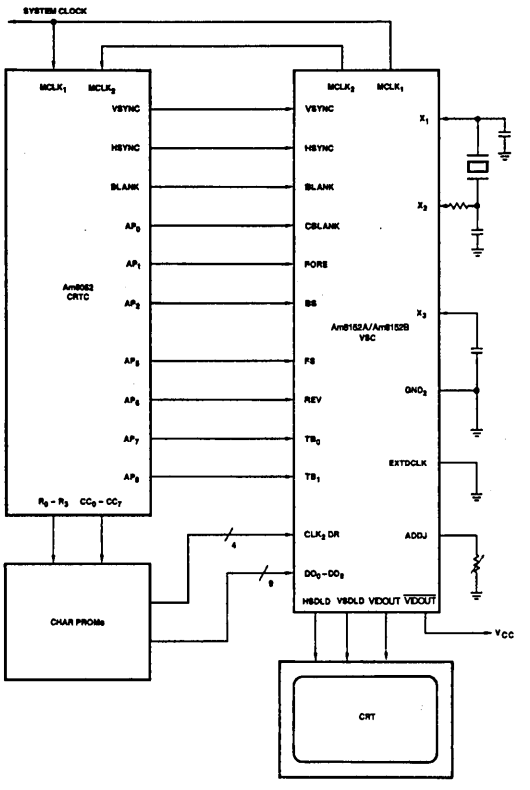
$(CLK_2\ DR_{0-3} + TB_{0-1}) + 2$	C	D
2	1	1
3	1	2
4	2	2
5	2	3
6	3	3
7	3	4
8	4	4
9	4	5
10	5	5
11	5	6
12	6	6
13	6	7
14	7	7
15	7	8
16	8	8
17	8	9
18	9	9
19	9	10

Note: C and D are measured in EXTDCLK periods.

Table 2

CLK_2, DR_{0-3} and TB_{0-1} are a 4-bit and 2-bit binary number respectively. For A, B, C, D refer to the Reset Timing Diagram.

3



AF002103

Figure 2. Am8152A/Am8152B Application with Am8052 CRT Controller

FUNCTIONAL DESCRIPTION

The Am8152A Video System Controller (VSC) supports both black and white and color video applications for CPUs, CRT controllers, and terminals. The essential functions of the VSC are to support proportional and non-proportional character display, to synchronize and mix character attributes with video, and to output the video in a four level analog or 2-bit digital serial format.

PARALLEL PIXEL LOADING

Pixel information that must be serialized for video transmission is loaded into the serial shift register via inputs $DD_0 - DD_8$. Information is loaded on both edges of the $MCLK_2$ character clock, as shown in Figure 3. The information set up on DD(0:7) prior to the falling edge of $MCLK_2$ is loaded into positions $VID_9 - VID_{16}$. Note that DD_8 information is ignored. Information set up on DD(0:8) prior to the rising edge of $MCLK_2$ is loaded into positions $VID_0 - VID_8$. Thus, up to 17 bits of pixel information can be loaded into the shift register. Note that if the character width is nine pixels or less the information captured on the falling edge of the $MCLK_2$ is not used.

CLK_2DR (0:3) and TB (0:1) determine the divide ratio for the character clock. The sum of both values specifies the character clock period in dot clocks. During the trailing blank, the VSC shifts out what was loaded into the shift register. Therefore, it is the responsibility of the user to insure that the pixels output during the trailing blank dot period are set to the blank level. If the total is greater than 17, this occurs automatically.

VIDEO OPERATION

Parallel video data is obtained from the character ROM inputs; bits are shifted out serially and mixed with attribute information such as underline, shifted underline, and any other video sources. Video is internally encoded into one of four levels: White, Grey, Black and Blank. White is the highest analog current level, and Blank is the lowest. This information is then output through two ports. (See Figure 5). One port provides a differential current source output into a 75Ω impedance, and the second port outputs encoded TTL video on two pins.

There are two distinct blank inputs to the Am8152A/Am8152B. BLANK is the CRTC's horizontal and vertical retrace period input which causes a blank output level to the display. CBLANK is an attribute input to selectively blank a character cell by forcing the video information for the particular character cell period to switch to the selected background color level. (See Table 3).

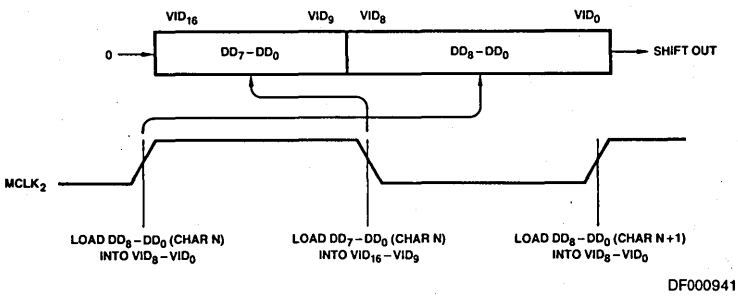


Figure 3. Shift Register Loading

CRYSTAL SPECIFICATION

The crystal used with the VSC may have the following specifications:

AT cut
Series Resonant
Shunt Capacitance: 7 pf maximum

VIDEO INPUTS/OUTPUTS

Video information may be input in a number of different ways. Table 3 depicts all the combinations of video outputs achievable with each of the various inputs. The background color is determined by a separate pin input allowing either a black or white background. Using the REVERSE VIDEO (REV) input, a grey background can also be selected. The foreground then becomes black or white according to the signal on the foreground SHIFT line. Foreground and video sums can be modified depending on the combination of background, foreground shift, and reverse inputs. The user may apply any of his video inputs to the foreground to obtain a desired effect.

TABLE 3. Am8152A/Am8152B VIDEO ATTRIBUTES

INPUTS				Am8152A/Am8152B VIDEO ATTRIBUTES
BS	FS	REV	$\overline{\text{CBLANK}}$ (DD (0 : 8) + FORE)	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

TB4

SYSTEM TIMING

The CPU clock ($MCLK_1$) output is derived from an on-board oscillator by an externally programmable divide ratio. The internal oscillator is capable of operating at a frequency of up to 80 MHz in the Am8152A and 40 MHz in the Am8152B. In crystal oscillator multiplier mode, the crystal should be an AT cut operating in series resonant mode using the fundamental frequency.

The character clock ($MCLK_2$) output to the CRTC is frequency modulated according to the chosen number of dots per character cell. The duty cycle of $MCLK_2$ is 50% (±1 dot clock period) and is derived from the internal dot clock whose divide ratio is set by the width of the character ROM plus the number of trailing blanks. A double width input further modifies $MCLK_2$, doubling the character width. During an active BLANK input, $MCLK_2$ is internally resynchronized to $MCLK_1$ if SSEL is HIGH. This action aligns character cells at the left-end side of the display, thereby eliminating "Dot Walk." The Vertical and Horizontal Sync (VSYNC, HSYNC) inputs from the CRT controller are buffered and delayed by one $MCLK_1$ or $MCLK_2$ clock period.

DOT CLOCK GENERATION WITH PLL

When using the internal oscillator of the VSC, care should be used in laying out the grounds and supplies for the part in order to minimize the jitter of the PLL. Under optimal conditions this jitter is less than 1 ns. If the jitter does occur it is normally less than 4 ns and can, over the operating range, be reduced to less than 1 ns by varying the duty cycle on X_1. The following table shows the worst-case jitter observed on typical parts over the operating range and the percentage Dot Clock at various Dot Clock frequencies.

Dot Clock Frequency (MHz)	Worst Case Jitter Observed (ns)	% Dot Clock
25	3.9	10
40	7.4	30
60	4.4	26
80	6.7	54

PROPORTIONAL/VARIABLE SPACING

Proportional spacing is achieved by programming, on a character-by-character basis, a number from two to nineteen dot clock periods per character. The character ROM pixel information is selectable from two to seventeen per character. Up to three trailing blank pixels can be concatenated to the character ROM input, making it easier to provide a straight right margin for right justification of text.

COLOR APPLICATION

The Am8152A/Am8152B may be used for many high-end color display applications. The foreground video and background information is mixed by the Am8152A/Am8152B, and the encoded TTL video output can be used externally to select a color mix for the particular pixel being displayed. The horizontal and vertical synchronization and video blank are output by the Am8152A/Am8152B.

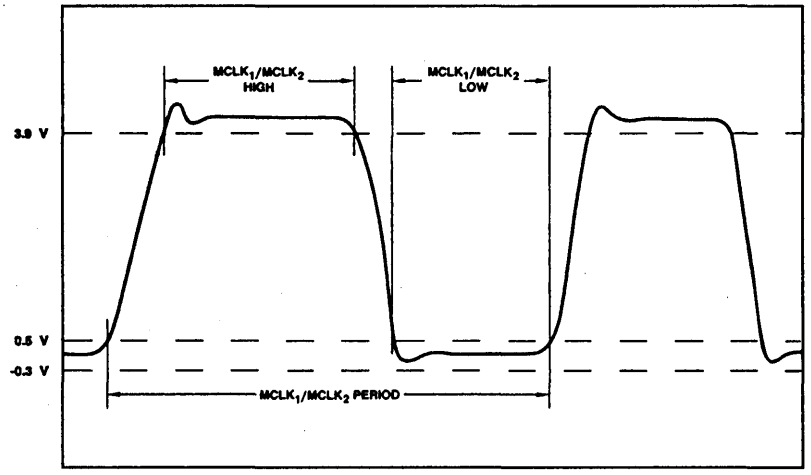
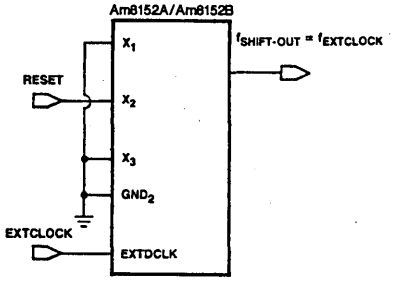


Figure 4. $MCLK_1/MCLK_2$ Output Waveform

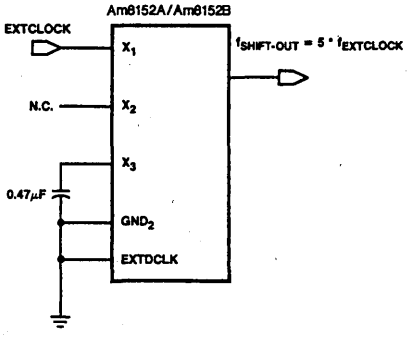
DOTCLOCK GENERATION MODE

EXTERNAL DOT CLOCK FLOW THROUGH MODE



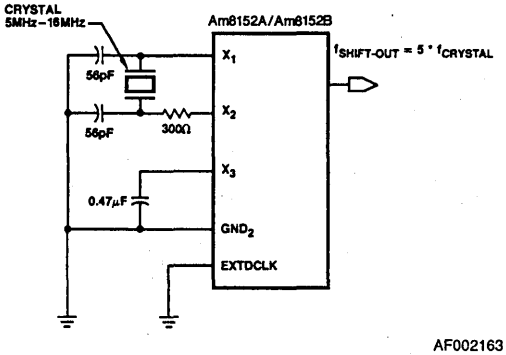
AF002142

EXTERNAL CLOCK MULTIPLIER MODE



AF002152

CRYSTAL OSCILLATOR MULTIPLIER MODE



AF002163

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Supply Voltage to Ground Potential
Continuous ..... −0.5 to +7.0V
DC Voltage Applied to Outputs for
High Output State ..... −0.5V to +V_{CC}
DC Input Voltage ..... −0.5 to +5.5V
DC Output Current into Outputs (See Note 2) 30mA
DC Input Current ..... −30 to +5.0mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +5.0 V ±5%

Military (M) Devices
Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... +5.0 V ±10%

Operating ranges define those limits over which the functionality of the device is guaranteed.

Am8152A/Am8152B

DC CHARACTERISTICS over operating range unless otherwise specified; Included in Group A, Subgroup 1, 2, 3 tests unless otherwise noted.
COM'L T_A = 0 to +70°C, V_{CC} = 5.0 V ±5%
MIL T_C = −55 to +125°C, V_{CC} = 5.0 V ±10%

Parameters	Description	Test Conditions				Min	Max	Units
V_{OH}	Output HIGH Voltage	V_{CC} = Min	$MCLK_1$, $MCLK_2$	I_{OH} = −0.1 mA		4.0		Volts
			TTL Output	I_{OH} = −1.0 mA	MIL	2.4		Volts
				I_{OH} = −2.6 mA	COM'L			
V_{OL}	Output LOW Voltage	V_{CC} = Min	$MCLK_1$, $MCLK_2$	I_{OL} = 0.1 mA			0.3	Volts
			TTL Output	I_{OL} = 16 mA			0.5	Volts
V_{IH}	Input HIGH Level	Guaranteed Input HIGH Voltage				2.0		Volts
V_{IL}	Input LOW Level	Guaranteed Input LOW Voltage					0.8	Volts
V_I	Input Clamp Voltage	V_{CC} = Min, I_{IN} = −18 mA					−1.2	Volts
I_{IL}	Input LOW Current	V_{CC} = Max, V_{IN} = 0.4 V		All Inputs (Except RES, EXTDCLK)			−0.4	mA
				RES, EXTDCLK			−1.0	mA
I_{IH}	Input HIGH Current	V_{CC} = Max, V_{IN} = 2.7 V		All Inputs (Except RES)			+50	µA
				RES			+600	µA
I_I	Input HIGH Current at Max Input Voltage	V_{CC} = Max, V_{IN} = 5.5 V					+1.0	mA
I_{SC}	Output Short Current (Notes 1, 2)	V_{CC} = Max		$MCLK_1$, $MCLK_2$		−50	−250	mA
				Others		−40	−130	mA
I_{CC}	Power Supply Current	V_{CC1} = Max, V_{CC2} = Max	Over Operating Ranges				415	mA

ANALOG ELECTRICAL CHARACTERISTICS (Notes 2–6)

The following conditions apply unless otherwise specified:
COM'L T_A = 0 to +70°C V_{CC} = 5.0V ±5%
MIL T_C = −55 to +125°C, V_{CC} = 5.0 V ±10%

Grey Level	VID_2	VID_1		VIDOUT		$\overline{VIDOUT}$	
				Min (%)	Max (%)	Min (%)	Max (%)
X	I	I	I_{White}	0	0	100	100
I	I	O	I_{Grey1}	37	45.5	54.5	63
O	I	O	I_{Grey2}	45	54	46	55
X	O	I	I_{Black}	89.5	93	7	10.5
X	O	O	I_{Blank}	100	100	0	0

Notes:
1. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
2. The absolute maximum rating for VIDOUT, $\overline{VIDOUT}$ and AADJ is 22 mA. Shorting VIDOUT, $\overline{VIDOUT}$, or AADJ to ground will destroy the device.
3. Test condition: Normal I_{white} for $\overline{VIDOUT}$ = 13.3 mA.
4. Positive Current flowing into VIDOUT/$\overline{VIDOUT}$.
5. $\overline{VIDOUT}$ output currents normalized to I_{white}
VIDOUT output currents normalized to I_{blank}.
6. VIDOUT and $\overline{VIDOUT}$ typically will not drift by more than 2% over the operating conditions.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified; included in Group A, Subgroup 9, 10, 11 tests unless otherwise noted.

COM'L $T_A = 0$ to 70°C, $V_{CC} = 5.0$ V ±5%
MIL $T_C = -55$ to +125°C, $V_{CC} = 5.0$ V ±10%

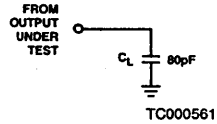
	Number	Description	Am8152A Min	Am8152A Max	Am8152B Min	Am8152B Max	Units
†	1	$MCLK_1$ Period (Note 7)	100		165		ns
†	2	$MCLK_2$ Period (Note 7)	70		165		ns
†	3	$MCLK_1$ HIGH (See Notes 5, 7) (See Figure 4)	38		70		ns
†	4	$MCLK_1$ LOW (See Notes 3, 5, 7) (See Figure 4)	37.5		70		ns
†	5	$MCLK_2$ HIGH (See Notes 6, 7) (See Figure 4)	23		70		ns
†	6	$MCLK_2$ LOW (See Notes 3, 6, 7) (See Figure 4)	20		70		ns
†	7	Data to $MCLK_2/TCLK_2$ RE (See Note 1, 7)	25		30		ns
	8	$MCLK_2/TCLK_2$ to Data Not Valid	0		0		ns
	9	VSYNC/HSYNC to $MCLK_1$ RE Setup (SSEL = HIGH)	25		30		ns
†	10	VSYNC/HSYNC to $MCLK_2$ RE Setup (SSEL = LOW) (Note 7)	25		30		ns
	11	$TCLK_2$ RE to $MCLK_2$ RE Delay		8		8	ns
	12	$TCLK_2$ FE to $MCLK_2$ FE Delay		12		15	ns
	13	$MCLK_1$ to VSLD, HSLD (SSEL = HIGH) (See Note 2)		$6 + T_D$		$6 + T_D$	ns
	14	$TCLK_2$ to VSLD, HSLD (SSEL = LOW) (See Note 2)		$6 + T_D$		$6 + T_D$	ns
†	15	DD(0 – 7) to $TCLK_2$ FE (See Note 7)	25		30		ns
	16	$TCLK_2$ RE to VID_1 VID_2 VAL (See Note 2)		$6 + T_D$		$6 + T_D$	ns
	17	BLANK FE to $MCLK_2$ RE Setup (SSEL = LOW)	22		25		ns
	18	BLANK FE to $MCLK_1$ FE Setup (SSEL = HIGH)	20		25		ns
	19	BLANK RE to $MCLK_1$ RE Setup (SSEL = HIGH)	22		25		ns
	20	BLANK RE to $MCLK_2$ RE Setup (SSEL = LOW)	22		25		ns
†	22	VID_1 to VID_2 Skew (See Note 7)	–5	+5	–7	+7	ns
†	24	EXTDCLK to $MCLK_1$ (See Note 7)		20		25	ns
†	25	EXTDCLK to $TCLK_2$ (See Note 7)		18		25	ns
	26	EXTDCLK to $MCLK_2$		23		25	ns
	27	EXTDCLK to VID_1/VID_2		18		20	ns
	28	EXTDCLK to HSDLD/VSDLD (SSEL HI)		18		20	ns
	29	EXTDCLK to HSDLD/VSDLD (SSEL LO)		18		20	ns
†	30	EXTDCLK to Data in Setup (See Note 7)	12		18		ns
	31	EXTDCLK to Data Not Valid Hold	18		20		ns
†	32	EXTDCLK to H/V SYNC Setup (See Note 7)	14		18		ns
†	33	EXTDCLK Period (See Note 7)	12.5		25		ns
†	34	EXTDCLK LOW Cycle (See Note 7)	5		7		ns
†	35	EXTDCLK HIGH Cycle (See Note 7)	5		7		ns
†	36	Reset Pulse Width (High) (See Note 7)	15		20		ns
†	37	Reset Low to EXTDCLK Setup	8.0		10		ns
†	38	EXTDCLK Period in PLL Mode (See Note 7)	12.5	50	25	50	ns

Notes: 1. Data includes CBLANK, FORE, REV, FS, DD_0–DD_8, TB0, TB1, BS, CLK_1DR, CLK_2DR, DW.
2. First Pixel of character. T_D is the dot clock period.
3. Max undershoot on these outputs is guaranteed to be –0.3V.
4. T_D is the dot clock period.
5. Guaranteed to 100ns $MCLK_1$ cycle time.
6. Guaranteed to 70ns $MCLK_2$ cycle time (even divide ratio only).
7. These parameters are guaranteed by device characterization or tested using bench top equipment.

† = Not included in Group A tests.

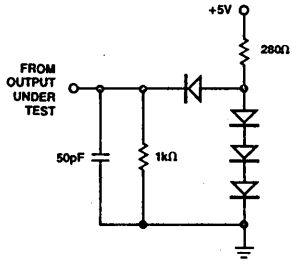
SWITCHING TEST CIRCUIT

$MCLK_1/MCLK_2$ OUTPUT



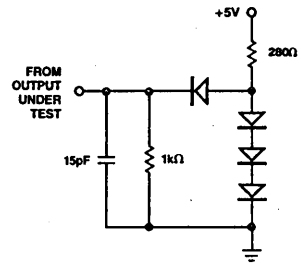
TC000561

TTL OUTPUTS EXCEPT $TCLK_2$



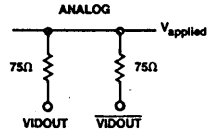
TC000701

$TCLK_2$ OUTPUT



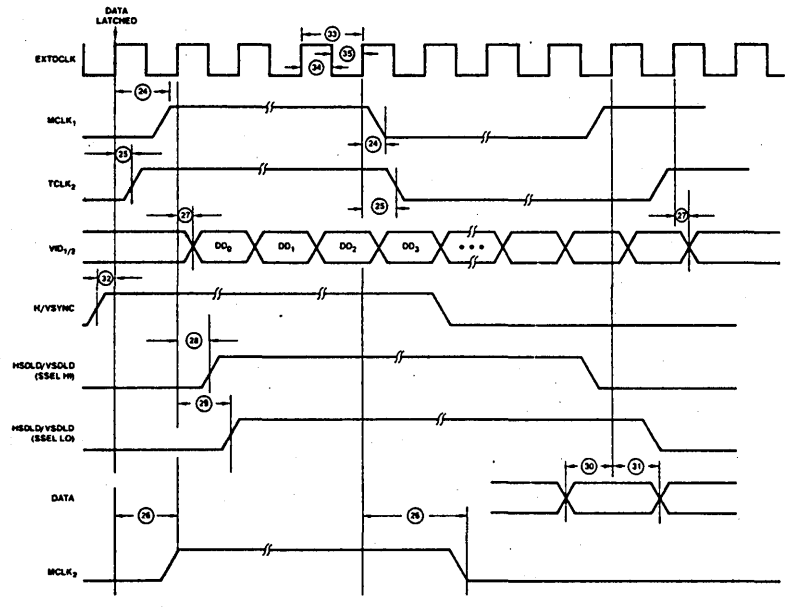
TC001531

ANALOG OUTPUTS



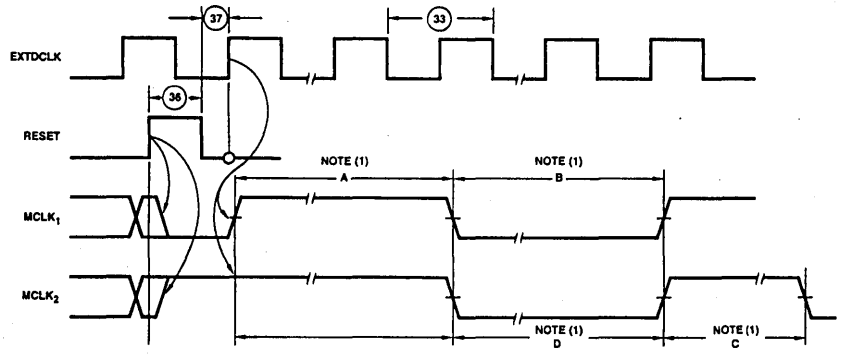
TC000551

Am8152A/Am8152B TIMING
(PARAMETERS MEASURED WITH RESPECT TO EXTDCLK)



WF003213

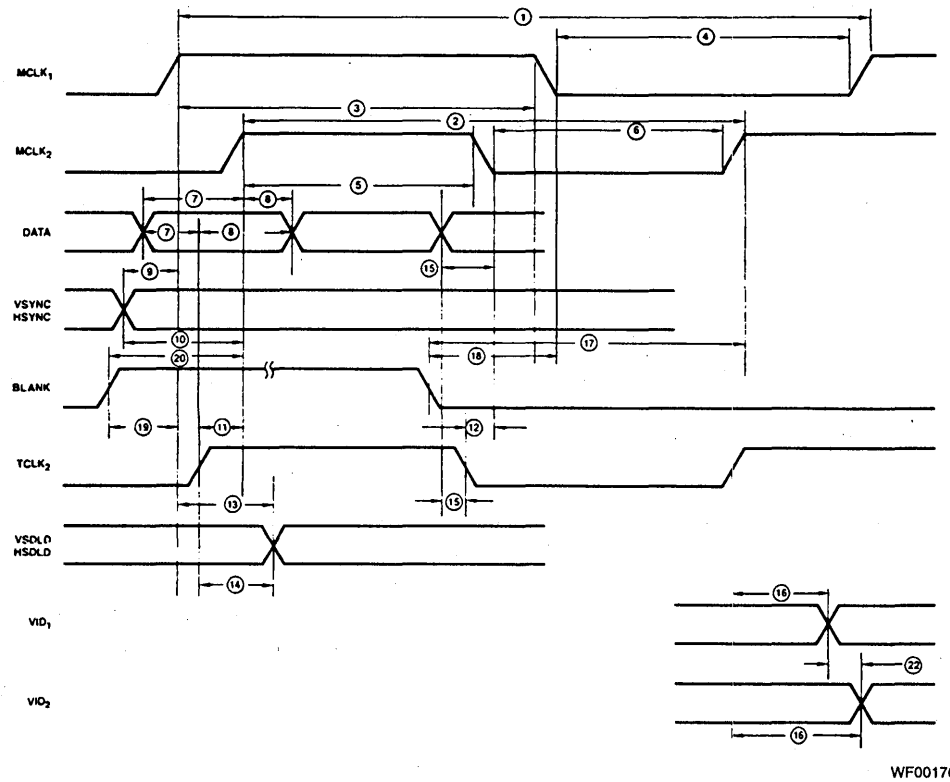
RESET TIMING FOR Am8152A/Am8152B



WF003192

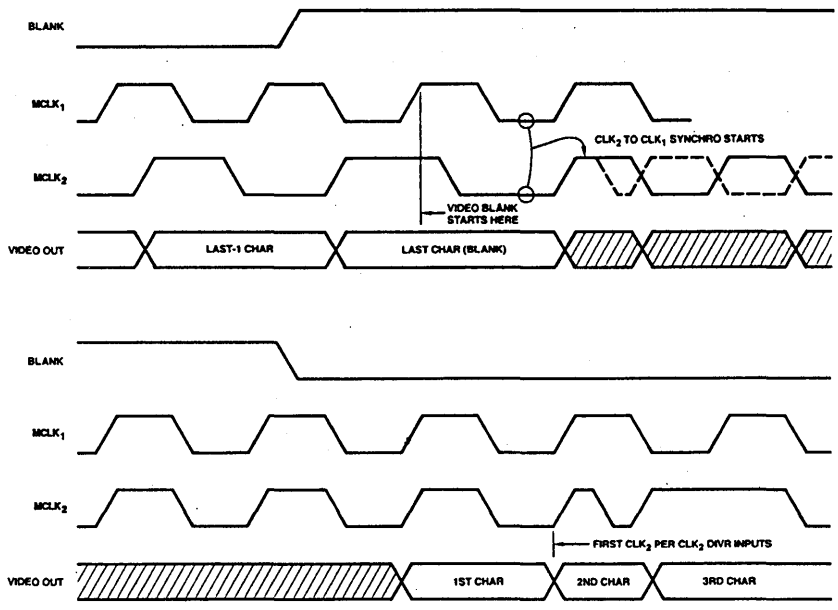
Note 1. See Pin Description section.

SWITCHING TIMING DIAGRAM — $MCLK_1/MCLK_2$

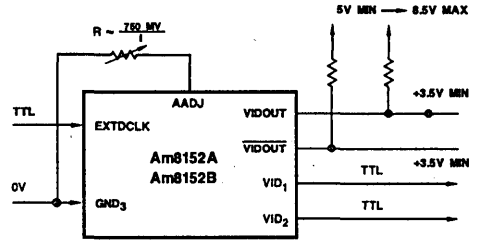


WF001762

VSC CLK_2 SYNCHRONIZATION (ONLY OCCURS IF SSEL IS HIGH)



WF001752



DF000952

Figure 5. Analog Video Outputs and Digital Video Outputs for Am8152A/Am8152B

Am8172

Video Data Assembly FIFO (VDAF)

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

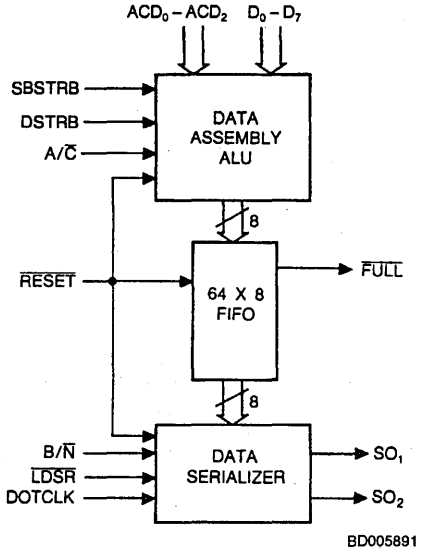
- Supports smooth panning and hardware windows
- Allows panning and window resolution of a single pixel
- Provides a temporary buffer between the memory and display
- Single 8-bit or dual 4-bit operation
- 10KH ECL with pixel rates of up to 200 MHz

GENERAL DESCRIPTION

The Am8172 Video Data Assembly FIFO is a high-speed ECL data assembler and serializer for use in bit-mapped graphics systems where flexible windowing and panning are desired. Positioned between the display memory and the color palette or monitor, the VDAF supports smooth panning and hardware windows on pixel boundaries. This is contrasted with traditional systems which support panning and hardware windows on word boundaries with words typically containing four to 32 pixels. Resolution on pixel boundaries is achieved through the use of a Data Assembly ALU, a 64 x 8 FIFO, and a Data Serializer.

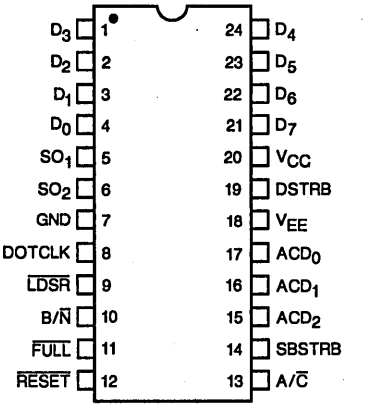
The Data Assembly ALU accepts display memory data as either one 8-bit word or two 4-bit words. Leading and trailing pixels not required in the final bit stream, as indicated by the control data, are removed from the words. The remaining pixels are then shifted and concatenated to form a continuous stream of video data. The FIFO provides a temporary buffer so that data can be provided to the display during periods when the display memory is incapable of doing so. This occurs when the display memory address is being updated by the system controller at a window boundary. A FIFO Full signal is provided by the VDAF to assist in preventing data overflows. The Data Serializer converts the parallel data at the output of the FIFO to serial data to be used by the color palette.

BLOCK DIAGRAM



BD005891

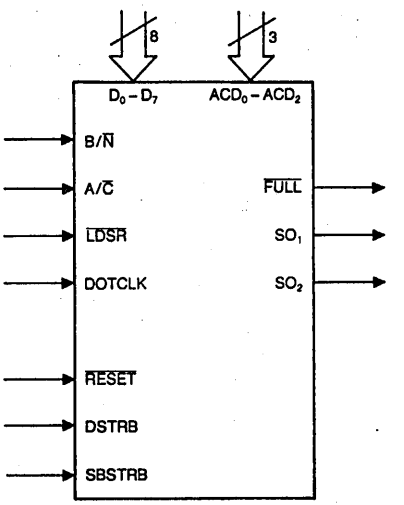
CONNECTION DIAGRAM
Top View



CD009141

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



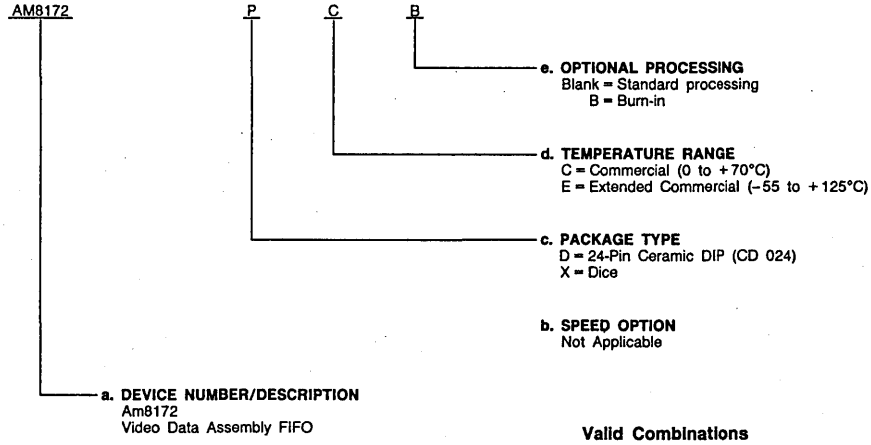
LS002191

V_{EE} = ECL Negative Supply
V_{CC} = TTL Positive Supply
GND = Ground

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM8172	DC, DCB, DE, DEB, XC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

A/$\overline{C}$ Address/$\overline{Count}$ (Input, TTL)
This pin selects how the control data loaded in with DSTRB is interpreted. When A/$\overline{C}$ is HIGH, the control data loaded with DSTRB is assumed to be the end bit address. When A/$\overline{C}$ is LOW, the control data is assumed to be a valid bit count.

ACD_0 - ACD_2 Assembly Control Data (Inputs, TTL)
Control data used to assemble the video data is latched from these pins on the rising edge of SBSTRB or DSTRB.

B/$\overline{N}$ Byte/$\overline{Nibble}$ (Input, TTL)
This pin selects whether the video data is interpreted to be one 8-bit word or two 4-bit words. When B/$\overline{N}$ is HIGH, the Data pins represent one 8-bit word. When B/$\overline{N}$ is LOW, the Data pins represent two 4-bit words.

D_0 - D_7 Data (Inputs, TTL)
Video data is latched from these pins on the rising edge of DSTRB. The data to be assembled may be one 8-bit word (D_0 - D_7 with D_0 shifted out first) or two 4-bit words (D_0, D_2, D_4, D_6 and D_1, D_3, D_5, D_7 with D_0 and D_1 shifted out first).

DOTCLK Dot Clock (Input, ECL)
DOTCLK is the Data Serializer clock. All operations of the Data Serializer take place on the rising edge of this clock.

DSTRB Data Strobe (Input, TTL)
The video data and the end bit address or valid bit count are latched internally from the Data and Assembly Control Data pins on the LOW-to-HIGH transition of this clock.

$\overline{FULL}$ $\overline{FIFO\ Full}$ (Output, TTL)
$\overline{FULL}$ goes HIGH on reset and stays HIGH until there are more than 56 bytes in the FIFO. $\overline{FULL}$ will go HIGH again when there are less than 56 bytes in the FIFO.

GND Ground

$\overline{LDSR}$ $\overline{Load\ Shift\ Register}$ (Input, ECL)
When $\overline{LDSR}$ is LOW, the Data Serializer is loaded in parallel from the FIFO on the rising edge of DOTCLK. When $\overline{LDSR}$ is HIGH, the Data Serializer shifts on the rising edge of DOTCLK.

$\overline{RESET}$ $\overline{Reset}$ (Input, TTL)
$\overline{RESET}$ when LOW clears the FIFO address counters and sets the internal registers to zero. Data should not be read from the FIFO at this time since it will be invalid. The VDAF should be reset on power-up.

SBSTRB Start Bit Strobe (Input, TTL)
The start bit address is latched internally from the Assembly Control Data pins on the LOW-to-HIGH transition of this clock.

SO_1, SO_2 Serial Outputs (Outputs, ECL)
SO_1 and SO_2 are the serial output pins of the Data Serializer. If 8-bit words are used the words are shifted out SO_1. If 4-bit words are used the word loaded in from D_0, D_2, D_4, and D_6 is shifted out SO_1 and the word loaded in from D_1, D_3, D_5, and D_7 is shifted out SO_2.

V_{CC} TTL Positive Supply

V_{EE} ECL Negative Supply

FUNCTIONAL DESCRIPTION

The Am8172 VDAF has three functional parts, the Data Assembly ALU, the FIFO, and the Data Serializer. The Data Assembly ALU removes unwanted bits from the display memory words and concatenates the remaining bits into words to be placed in the FIFO. The FIFO allows the RAM timing to be decoupled from the Data Serializer timing. This is important at window boundaries where two words may be accessed and merged to form one word of video data. The FIFO is 64 words deep (one 64 x 8 FIFO or two 64 x 4 FIFOs). A FIFO Full signal ($\overline{FULL}$) is provided to allow the controller to keep the FIFO from overflowing. The output of the FIFO is connected to the Data Serializer which operates at up to 200 MHz. A synchronous load input is provided to load the Data Serializer from the FIFO.

The Am8172 accepts either one 8-bit or two 4-bit words from the display memory according to the input on the B/$\overline{N}$ pin. With B/$\overline{N}$ HIGH, 8-bit words are being used and eight bits from one plane of the display memory should be presented on D_0 - D_7. With B/$\overline{N}$ LOW, 4-bit words are used and four bits from each of two planes of the display memory should be presented on D_0, D_2, D_4, D_6 and D_1, D_3, D_5, D_7. For both 8-bit and 4-bit words the data on lower numbered Data lines is shifted out before the data on higher numbered lines (i.e., D_0 for 8-bit words, D_0 and D_1 for 4-bit words).

The Data Assembly ALU merges words to provide panning and hardware windows on pixel boundaries. An example of what is required is shown in Figure 1. The top two lines are the words as read from the display memory. The bottom line shows the words as required for the display. As can be seen in the figure, the Data Assembly ALU removes bits and realigns word boundaries to achieve the proper sequence for the display. This example and all further examples are given for 8-bit words. The examples apply equally well for 4-bit words. For panning on a pixel-by-pixel basis, pixels must be removed from the first word read from the display memory. This is the first operation shown in the figure. For hardware window with pixel resolution, two words must be merged, and in the process, bits removed from each word. This is the second operation shown in the figure.

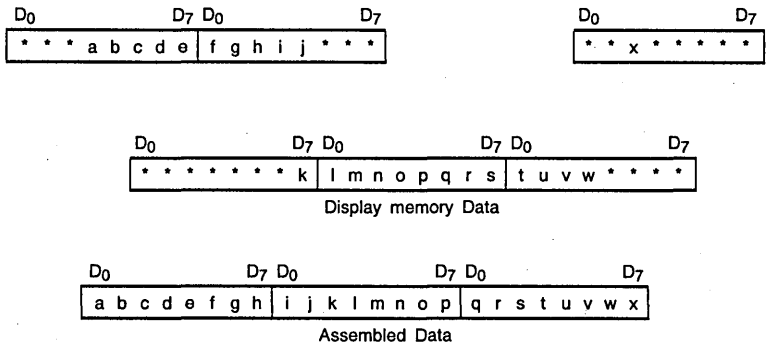


Figure 1. Data Assembly Operation

Generalizing from Figure 1, there are four possible types of words the Am8172 can receive. These are shown in Table 1. With each type is given an example word from Figure 1 and the control data required for that example word. The first column under Control Data provides the information to find the first valid bit. The second and third columns each provide enough information for the Data Assembly ALU to find the last valid bit. The Am8172 only requires one of these last two columns. The A/$\overline{C}$ pin indicates to the chip whether to expect an end bit address or a valid bit count. The 'first word' type requires bits to be removed from the start of the word. The 'middle word' type does not require any bits to be removed. The 'last word' type requires bits to be removed from the end of the word. The 'small word' type requires bits to be removed from the start and the end of the word.

Bit addresses from the point of view of an Am8172 in 8-bit mode go from 0 to 7 with 0 corresponding to D_0 and 7 corresponding to D_7. Bit addresses from the point of view of the Am8172 in 4-bit mode are 0, 2, 4, and 6 with 0 corresponding to D_0 and D_1, and 6 corresponding to D_6 and D_7. The valid bit count is the number of valid bits including the start bit and the end bit. In 8-bit mode the valid bit count goes from 1 to 8, with 8 encoded as a 0. In 4-bit mode the valid bit count is 2, 4, 6, or 8, with 2 corresponding to one valid bit per plane and 8 corresponding to four valid bits per plane. Again, 8 is encoded as a 0.

As discussed above, each display memory word requires two words of control data. First, the Am8172 expects the start bit address on the Assembly Control Data lines. This address is strobed into the VDAF by the rising edge of SBSTRB. Secondly, the Am8172 expects an end bit address or valid bit count which is strobed into the VDAF along with the actual display memory word by the rising edge of DSTRB. In the case of 4-bit words, the same assembly operation is performed on each word, so only one set of control data is needed. In the case of display words in which all eight bits are valid, the start bit is always 0 and therefore redundant. To reduce the number of strobes required by the Am8172, the start bit address is internally reset to 0 following each DSTRB. This means that a start bit address must be strobed into the VDAF only for display words of the 'first word' or 'small word' type, as described in Table 1. Since these types are only found at the start of a scan line or at window boundaries, a start bit address is only required at these points. In the case of a display memory consisting of VRAMs, this conveniently corresponds to the times when a transfer cycle is required. In the case of a display memory consisting of standard RAMs operating in page mode, this corresponds to the times at which the Row Address must be changed. Table 2 shows the sequence of strobes to load the example sequence of words shown in Figure 1. The start bit addresses are only strobed in where required and valid bit counts are used instead of end bit addresses.

TABLE 1. WORD TYPES RECEIVED BY THE Am8172

WORD TYPES	EXAMPLE WORDS D_0 ... D_7	CONTROL DATA		
		Start Bit Address	Valid Bit Count	End Bit Address
First Word	* * * a b c d e	3	5	7
Middle Word	l m n o p q r s	0	0	7
Last Word	f g h i j * * *	0	5	4
Small Word	* * x * * * * *	2	1	2

TABLE 2. STROBE SEQUENCE

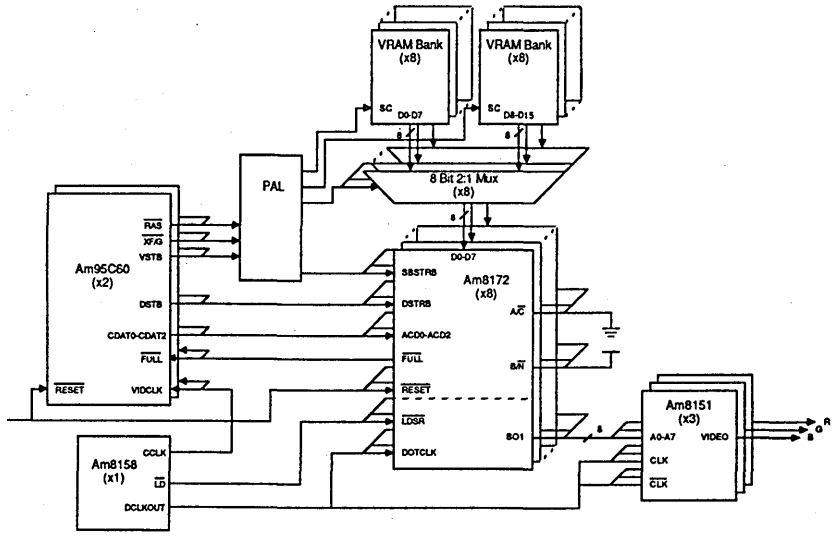
ASSEMBLY CONTROL DATA				VIDEO DATA								STROBE
ACD_2		ACD_0	Description	D_0							D_7	
0	1	1	3 (start bit address)									SBSTRB
1	0	1	5 (valid bit count)	*	*	*	a	b	c	d	e	DSTRB
1	0	1	5 (valid bit count)	f	g	h	i	j	*	*	*	DSTRB
1	1	1	7 (start bit address)									SBSTRB
0	0	1	1 (valid bit count)	*	*	*	*	*	*	*	k	DSTRB
0	0	0	0 (valid bit count)	l	m	n	o	p	q	r	s	DSTRB
1	0	0	4 (valid bit count)	t	u	v	w	*	*	*	*	DSTRB
0	1	0	2 (start bit address)									SBSTRB
0	0	1	1 (valid bit count)	*	*	x	*	*	*	*	*	DSTRB

At a set time following the DSTRB, a word is available at the output of the Data Asssembly ALU. If less than eight bits (two sets of four bits in 4-bit mode) have been assembled, the result is stored in an internal temporary register until another display memory word is given to the VDAF. If exactly eight bits have been assembled, these bits are written into the FIFO and, if the FIFO is empty, are available at the inputs of the Data Serializer. If more than eight bits have been assembled, the first eight bits are written into the FIFO and the remaining bits stored in an internal temporary register.

The $\overline{FULL}$ signal is set LOW when more than 56 words are contained in the FIFO. Since the FIFO has room for 64 words, up to eight words may be strobed into the VDAF after the FIFO has indicated it is full. The $\overline{FULL}$ signal is set HIGH when the number of words in the FIFO falls below 56.

All operations of the Data Serializer are synchronous with the DOTCLK. DOTCLK may be asynchronous with the SBSTRB and DSTRB of the data assembler. The serializer is loaded with words from the output of the FIFO on any rising edge of DOTCLK while $\overline{LDSR}$ is LOW. In 8-bit mode, one word of eight bits is loaded into the Data Serializer and shifted out SO_1. In 4-bit mode, two words of four bits are loaded into the Data Serializer. Bits from D_0, D_2, D_4, or D_6 are loaded into one half of the serializer and shifted out SO_1. Bits from D_1, D_3, D_5, or D_7 are loaded into the other half of the serializer and shifted out SO_2. The Data Serializer shifts on each DOTCLK rising edge with $\overline{LDSR}$ HIGH. When all the data loaded into the Data Serializer has been shifted out the serializer will shift out 0s.

APPLICATIONS



BD007450

Figure 2. Interfacing VDAF Input to 16-Bit VRAM-Based Display Memory In a QPDM Graphics System

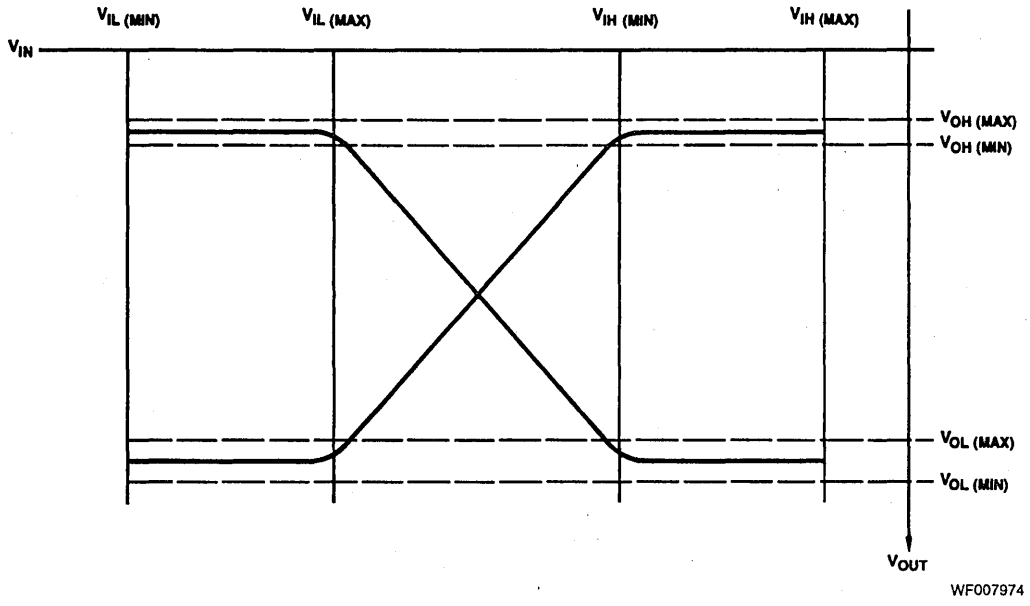


Figure 3. Am8172 ECL Specifications

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature Under Bias ..... −55 to +125°C
Supply Voltage to Ground Potential
Continuous (TTL) ..... −0.5 to +7.0 V
Supply Voltage to Ground Potential
Continuous (ECL) ..... +0.5 to −7.0 V
DC Voltage Applied to Outputs for
HIGH Output State (TTL) ..... −0.5 to +V_{CC} Max.
DC Input Voltage (TTL) ..... −0.5 to + 7.0 V
DC Output Current, Into Outputs (TTL) ..... 30 mA
DC Input Current (TTL) ..... −30 to +5.0 mA
DC Input Voltage (ECL) ..... +0.5 to V_{EE}
DC Output Current Into Outputs (ECL) .... −30 to +0.1 mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
TTL Supply Voltage (V_{CC}) ..... +5 V ±5%
ECL Negative Supply Voltage (V_{EE}) ..... −5.2 V ±5%

Extended Commercial (E) Devices
Case Temperature (T_C) ..... −55 to +125°C
TTL Supply Voltage (V_{CC}) ..... +5 V ±5%
ECL Negative Supply Voltage (V_{EE}) ..... −5.2 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified (T_A = 0 to +70°C; V_{CC} = 5 V ±5%, V_{EE} = −5.2 V ± 5%)

Parameter Symbol	Parameter Description	Test Conditions (Note 6)	Min.	Typ. (Note 5)	Max.	Units
V_{OH}	Output Voltage HIGH Level	V_{CC} = Min I_{OH} = −1.0 mA	2.4			Volts
V_{OL}	Output Voltage LOW Level	V_{CC} = Min. All I_{OL} = 16 mA			0.5	Volts
V_{IH}	Input Voltage HIGH Level	Guaranteed Input HIGH Voltage for All Inputs (Note 7)	2.0			Volts
V_{IL}	Input Voltage LOW Level	Guaranteed Input LOW Voltage for All Inputs (Note 7)			0.8	Volts
V_{IK}	Input Clamp Voltage	V_{CC} = Min., I_I = −18 mA			−1.2	Volts
I_{VL}	Input Current at Maximum Input Voltage	V_{CC} = Max., V_I = 5.5 V			1.0	mA
I_{IH}	HIGH-Level Input Current, TTL Input	V_{CC} = Max., V_{IH} = 2.4 V			40	µA
I_{IL}	LOW-Level Input Current, TTL Input	V_{CC} = Max., V_{IL} = 0.4 V			−250	µA
I_{SC}	Short-Circuit Output Current	V_{CC} = Max.	−10		−50	mA
I_{CC}	TTL Supply Current	V_{CC} = Max.; Outputs Open		25	40	mA
I_{EE}	ECL Supply Current	V_{EE} = Max.; Outputs Open		280	385	mA

ECL Characteristics (Note 3)

Parameter Description	Parameter Symbol	Test Conditions	−55°C	0°C	25°C	70°C	125°C	Unit
ECL Outputs SO_1 and SO_2	V_{OH} (Max.)	50 Ω to −2 V	−860	−840	−810	−730	−650	mV
	V_{OH} (Min.)		−1100	−1030	−990	−935	−860	
	V_{OL} (Max.)	50 Ω to −2 V	−1690	−1665	−1650	−1625	−1570	
	V_{OL} (Min.)		−1900	−1900	−1900	−1900	−1900	
ECL Inputs $\overline{LDSR}$, DOTCLK	V_{IH} (Max.)	(Note 7)	−860	−840	−810	−730	−650	mV
	V_{IH} (Min.)		−1215	−1145	−1105	−1045	−1005	
	V_{IL} (Max.,)	(Note 7)	−1590	−1565	−1550	−1525	−1470	mV
	V_{IL} Min.)		−1900	−1870	−1850	−1830	−1800	
	I_{IH}	V_{EE} = Max V_{IN} = V_{IH} (Max.)	250	200	200	200	200	µA
	I_{IL}	V_{EE} = Max. V_{IN} = V_{IL} (Min.)	200	150	150	150	150	µA

Notes: See notes following Switching Characteristics section.

PRELIMINARY

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified (Note 4)

No.	Parameter Symbol	Parameter Description	Min.	Max.	Units
1	t_{CLK}	DOTCLK Clock Period		5	ns
2	t_W	DOTCLK HIGH Pulse Width	2		ns
3	t_W	DOTCLK LOW Pulse Width	2		ns
4	t_S	DSTRB to DOTCLK (FIFO Fall-Through Time)		60	ns
5	t_S	$D_0 - D_7$ to DSTRB ↑ Setup	5		ns
6	t_H	$D_0 - D_7$ to DSTRB ↑ Hold	2		ns
7	t_S	$ACD_0 - ACD_2$ to DSTRB ↑ Setup	5		ns
8	t_H	$ACD_0 - ACD_2$ to DSTRB ↑ Hold	10		ns
9	t_S	$ACD_0 - ACD_2$ to SBSTRB ↑ Setup	10		ns
10	t_H	$ACD_0 - ACD_2$ to SBSTRB ↑ Hold	10		ns
11	t_W	DSTRB and SBSTRB HIGH Pulse Width	18		ns
12	t_W	DSTRB and SBSTRB LOW Pulse Width	18		ns
13	t_{PHL}	DSTRB to $\overline{FULL}$ ↓		40	ns
14	t_{PLH}	DOTCLK to $\overline{FULL}$ ↑		30	ns
15	t_W	$\overline{RESET}$ Pulse Width		30	ns

No.	Parameter Symbol	Parameter Description		C Devices (Note 2) 0°C	C Devices (Note 2) 25°C	C Devices (Note 2) 70°C	E/M Devices (Note 3) −55°C	E/M Devices (Note 3) 125°C	Units
16	t_S	$\overline{LDSR}$ ↓ to DOTCLK ↑ Setup (Load)	Min.	1.6	1.5	1.4	2.0	1.4	ns
17	t_S	$\overline{LDSR}$ ↑ DOTCLK ↑ Setup (No Load)	Min.	0.8	0.8	0.8	0.8	1.0	ns
18	t_H	$\overline{LDSR}$ ↑ to DOTCLK ↑ Hold (Load)	Min.	0.6	0.6	0.5	0.8	0.5	ns
19	t_H	$\overline{LDSR}$ ↓ to DOTCLK ↑ Hold (No Load)	Min.	0.5	0.5	0.5	0.5	0.5	ns
20	t_{PD}	DOTCLK ↑ to SOUT	Max.	3.5	3.6	4.1	2.8	4.2	ns
	t_{PD}	DOTCLK ↑ to SOUT	Min.	1.7	1.9	2.2	1.0	2.3	ns

PRELIMINARY

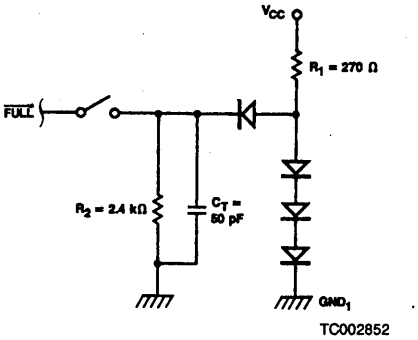
Notes:
1. Devices are not subjected to ABSOLUTE MAXIMUM RATINGS in production. Non-production samples have been subjected to ABSOLUTE MAXIMUM RATINGS.
2. OPERATING RANGES are guaranteed for steady-state conditions (no air flow). Hot temperature testing is elevated to simulate steady-state conditions when using pulse test techniques. Cold testing is at the specified temperatures.
3. Guaranteed with transverse air flow exceeding 500 linear F.P.M and two minute warm-up period. Typical thermal resistance values of the package are:

	Hermetic DIP
θ_{JA} (Junction-to-Ambient) = °C/Watt (still air)	50
θ_{JA} (Junction-to-Ambient) = °C/Watt (at 500 F.P.M. air flow)	15
θ_{JC} (Junction-to-Case) = °C/Watt	15

Die Size = 0.190 x 0.217
Transistor Count = 5600

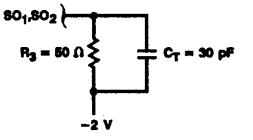
4. Parameter numbers 1, 2, 3, 6, 16-20 are guaranteed through characterization and correlation to other tests and are not directly measured in production.
5. All typical values are $V_{CC} = 5.0$ V, $V_{EE} = -5.2$ V, and $T_A = 25°C$.
6. For conditions shown as Min. or Max., use the appropriate values specified under recommended operating ranges.
7. V_{IH} threshold is measured at V_{CC} = Max. and V_{EE} = Min. with all other inputs HIGH.
   V_{IL} threshold is measured at V_{CC} = Min. and V_{EE} = Max. with all other inputs LOW.
   This test method is used to guarantee V_{IH} and V_{IL}.

SWITCHING TEST CIRCUITS



TC002852

A. TTL Output



TC002861

B. ECL Outputs

Test Output Loads

Pin Name	Test Circuit	R_1 (Ω)	R_2 (kΩ)	R_3 (Ω)	C_T (pF)*
FULL	A	270	2.4	–	50
SO_1	B	–	–	50	30
SO_2					

PRELIMINARY

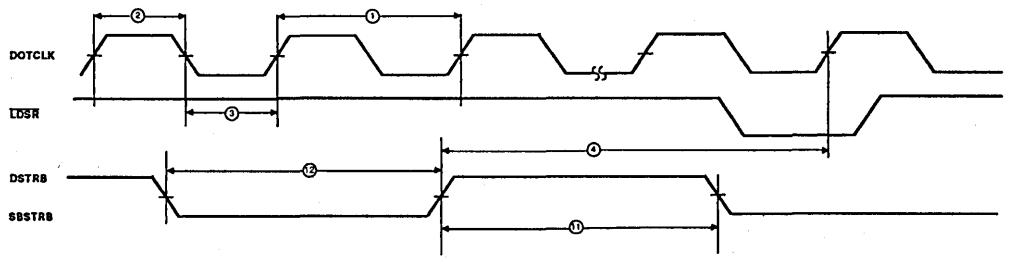
*C_T includes probe and jig capacitance.

SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS

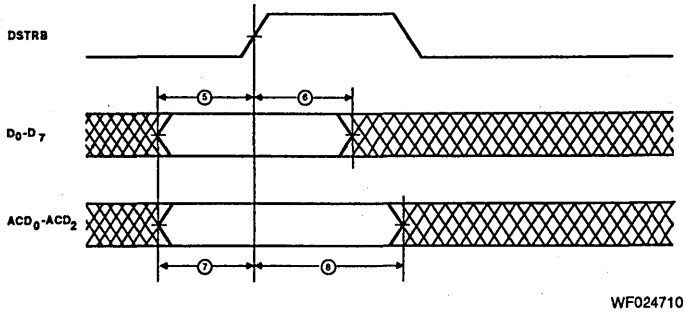
WAVEFORM	INPUTS	OUTPUTS
———	MUST BE STEADY	WILL BE STEADY
———		
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

KS000010



WF024701

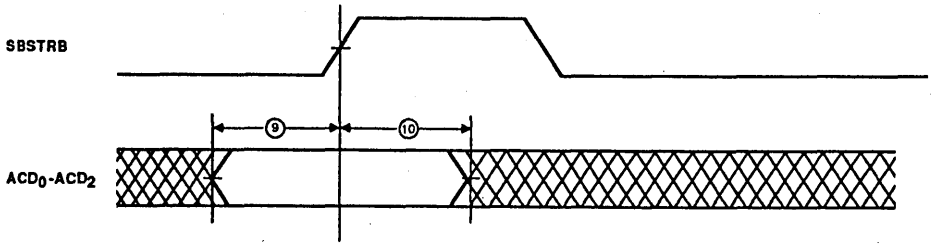
Strobe Timing



WF024710

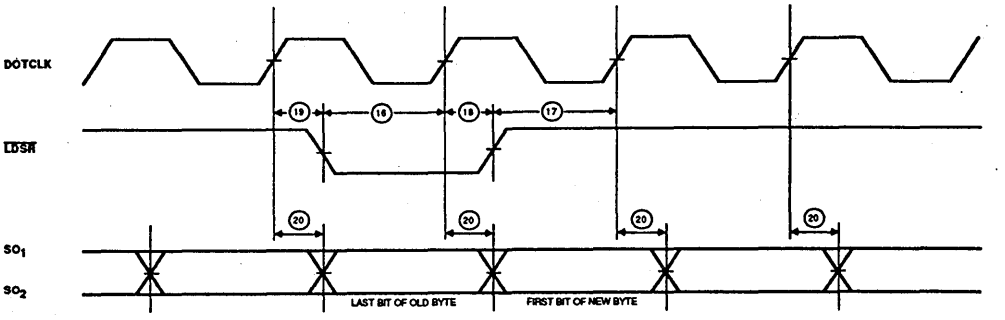
Data Timing

SWITCHING WAVEFORMS (Cont'd.)



WF024720

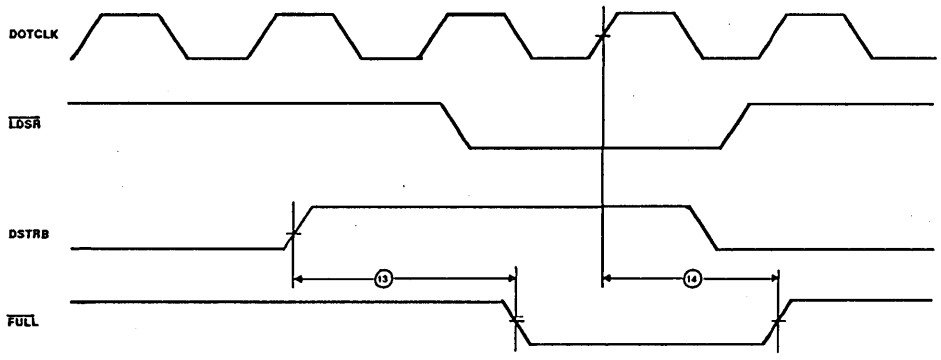
Address Control Data Timing



WF024730

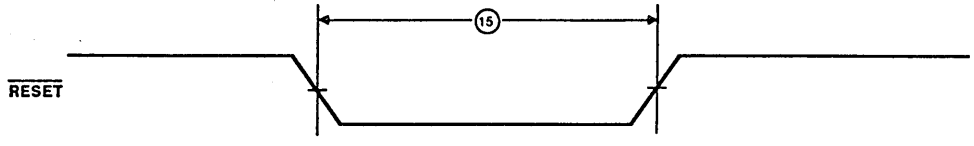
Load and Shift Timing

SWITCHING WAVEFORMS (Cont'd.)



WF024741

FIFO Timing



WF024750

Reset Timing

Am8177

Video Data Serializer

FINAL

DISTINCTIVE CHARACTERISTICS

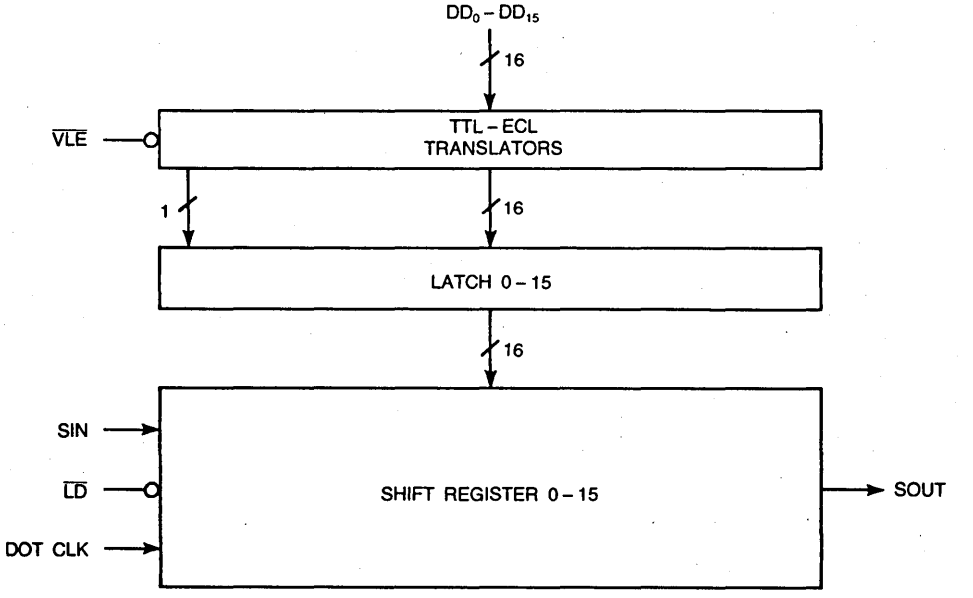
- 200-MHz parallel-to-serial shift register
- Cascadable in increments of 16 bits
- 24-pin slim-line DIP

GENERAL DESCRIPTION

The Am8177 Video Data Serializer (VDS) is a 16-bit parallel-to-serial shift register for use in bit-mapped display applications. The VDS can accommodate video words of up to 16 bits; wider display memories can be handled by cascading VDSs using the Serial In (SIN) line.

The Am8177 VDS is a part of AMD's Display Products Family which also includes the Am8151A Graphics Color Palette, Am8172 Video Data Assembly, FIFO, and the Am95C60 Quad Pixel Dataflow Manager.

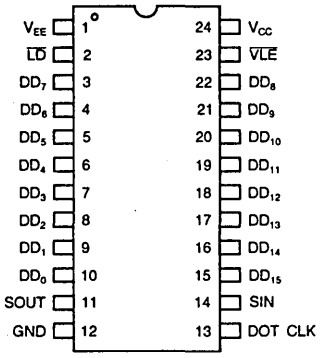
BLOCK DIAGRAM



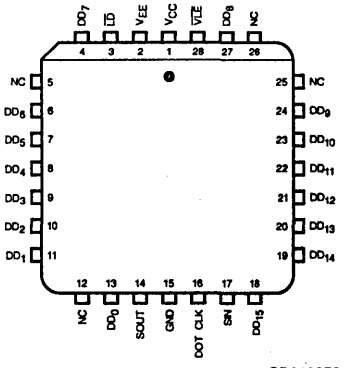
BD005370

Publication # 07080 Rev. C Amendment /0
Issue Date: August 1989

CONNECTION DIAGRAMS
Top View



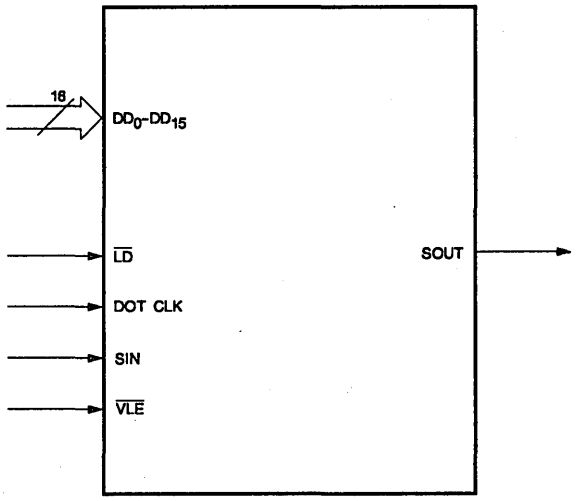
CD006090



CD010070

Note: Pin #1 is marked for orientation

LOGIC SYMBOL

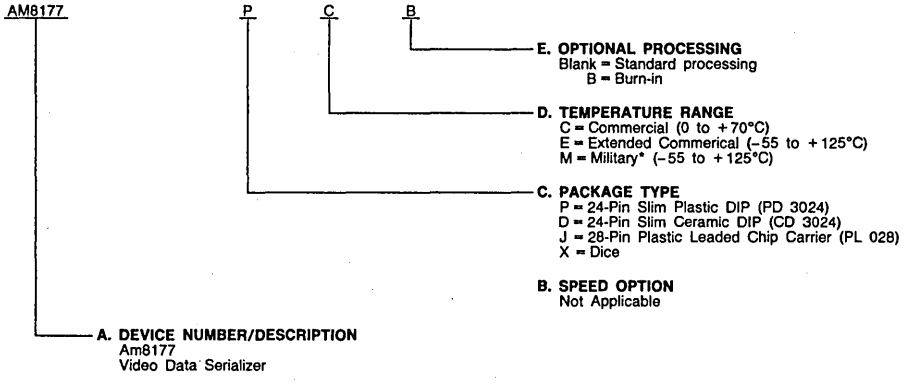


LS002740

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **A. Device Number**
B. Speed Option (if applicable)
C. Package Type
D. Temperature Range
E. Optional Processing



Valid Combinations	
AM8177	PC, PCB, DC, DCB, DE, DEB, JC, XC

* Military or Limited Military temperature range products are "NPL"(Non-Complaint Products List) or Non-MIL-STD-883C Compliant products only.

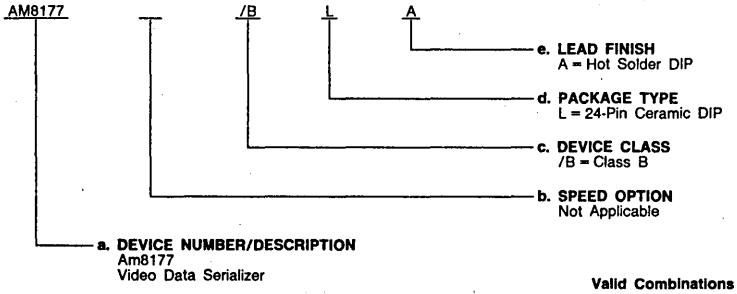
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM8177	/BLA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

PIN DESCRIPTION

DOT CLK Dot Clock (Input, ECL)
Shift Register Clock. All operations in the shift register take place on the rising edge of this clock.

$DD_0 - DD_{15}$ Parallel Data In (Inputs, TTL)
$DD_0 - DD_{15}$ are the parallel data input pins. DD_0 is the first bit to be shifted out; DD_1 is the next.

$\overline{LD}$ Load (Input, ECL)
When $\overline{LD}$ is active (LOW), the shift register is loaded in parallel on the next rising edge of the clock. If $\overline{VLE}$ is HIGH, data is loaded from the latch; if $\overline{VLE}$ is LOW, data is loaded directly from the parallel data input pins.

SIN Serial In (Input, ECL)
SIN is used when cascading VDSs and is connected to SOUT of the higher order VDS.

SOUT Serial Out (Output, ECL)
SOUT is the serialized output of the VDS. It is also used when cascading VDSs and is connected to SIN of the lower order VDS.

$\overline{VLE}$ Video Latch Enable (Input, TTL)
Active LOW enable for the latch. The latch may be used to provide a pipeline between the display RAM and the shift register. The parallel data may be loaded into the latch as soon as it is available at the display RAM outputs and then loaded into the shift register when it is needed. The latch may be kept transparent by keeping $\overline{VLE}$ LOW. When $\overline{VLE}$ goes HIGH, the parallel data is latched into the latch and remains so until $\overline{VLE}$ goes LOW again.

V_{CC} TTL Positive Supply

V_{EE} ECL Negative Supply

GND Ground

FUNCTION TABLE

$\overline{LD}$	$\overline{VLE}$	ACTION
0	0	Load data into shift register from data pins
0	1	Load data into shift register from latch
1	0	Shift
1	1	Shift

$\overline{VLE}$	ACTION
1	Latches data from data pins into latch
0	Latch appears transparent

FUNCTIONAL DESCRIPTION

The Am8177 Video Data Serializer (VDS) is a 16-bit parallel-to-serial shift register intended for use in bit-mapped video applications. The VDS is loaded in parallel with up to 16 bits from a single bit-plane. The bits are then serialized at the DOT CLK rate.

The VDS parallel data inputs are TTL for ease in communicating with the bit map, while the clock and serializer controls are ECL to allow the fast bit rates required for high-density screen formats.

A set of latches are provided between the parallel data input pins and the shift register. This allows the next word of data from the RAMs to be captured without affecting the contents of the shift register. This provides a means of decoupling (to some degree) the video RAM timing and the serializer timing. $\overline{VLE}$ is used to load the latch and then $\overline{LD}$ is used to load the data into the shift register.

The VDS can accommodate video words of up to 16 bits. Wider display memories can be handled by cascading VDSs using the Serial In (SIN) line.

APPLICATIONS

In a typical video system, parallel pixel data is accessed from the bit-map RAMs during the character clock cycle. At the completion of the access, this data can be latched into the Video Data Serializer using $\overline{VLE}$. $\overline{CAS}$, for example, would have the correct timing to drive $\overline{VLE}$, but external logic would be required to discriminate between video cycles and update cycles.

The transfer from the latch to the shift register will normally take place immediately before the next character clock, using the $\overline{LD}$ input.

The $\overline{LD}$ signal occurs every character clock and must be synchronous to DOT CLK. An appropriate signal is provided by the Am8158 Video Timing Controller which receives an asynchronous $\overline{VLE}$ and outputs the required synchronous $\overline{LD}$.

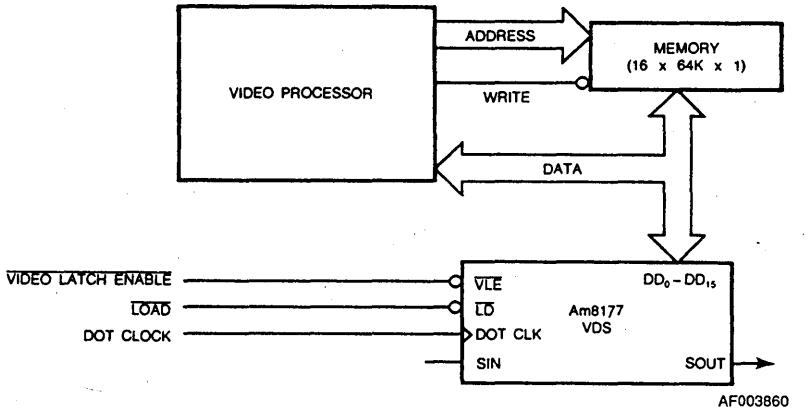


Figure 1. Single Bit-Plane Application

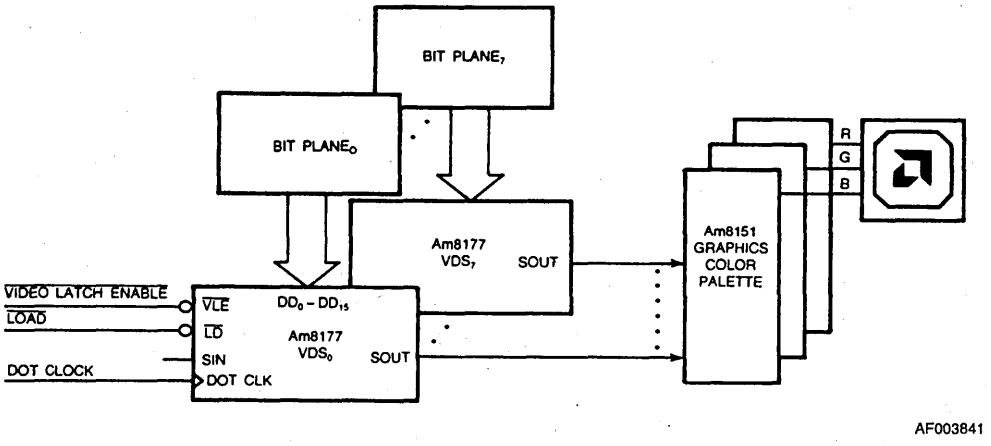


Figure 2. Multiple Bit-Plane Application

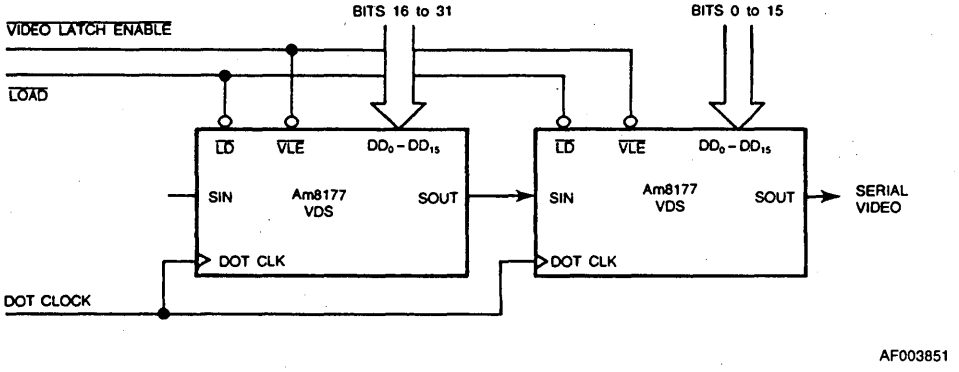


Figure. 3. Typical Cascading Application

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature Under Bias ..... −55 to +125°C
Supply Voltage to Ground Potential
Continuous (TTL) ..... −0.5 to +7.0 V
Supply Voltage to Ground Potential
Continuous (ECL) ..... +0.5 to −7.0 V
DC Input Voltage (TTL) ..... −0.5 to +7.0 V
DC Input Current (TTL) ..... −30 to +5.0 mA
DC Input Voltage (ECL) ..... +0.5 to V_{EE}
DC output Current into Outputs (ECL) ..... −30 to +0.1 mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices (Note 2)
Temperature (T_A) ..... 0 to +70°C
TTL Supply Voltage (V_{CC}) ..... +5.0 V ±5%
ECL Negative Supply Voltage (V_{EE}) ..... −5.2 V ±5%

Extended Commercial (E) Devices (Note 6)
Temperature (T_C) ..... −55 to +125°C
TTL Supply Voltage (V_{CC}) ..... +5.0 V ±5%
ECL Supply Voltage (V_{EE}) ..... −5.2 V ±5%

Military* (M) Devices (Note 6)
Temperature (T_C) ..... −55 to +125°C
TTL Supply Voltage (V_{CC}) ..... +5.0 V ±10%
ECL Supply Voltage (V_{EE}) ..... −5.2 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

*Military Product 100% tested at T_C = +25°C, +125°C, and −55°C.

DC CHARACTERISTICS over operating range (TTL) unless otherwise specified (Note 6)

Symbol	Parameter	Test Condition (Note 3)	Min.	Typ. (Note 5)	Max.	Units
V_{IH}	Input HIGH Level	Guaranteed Input HIGH Voltage (Note 4)	2.0			Volts
V_{IL}	Input LOW Level	Guaranteed Input LOW Voltage (Note 4)			0.8	Volts
V_I	Input CLAMP Voltage	V_{CC} = Min., I_{IN} = −18 mA			−1.2	Volts
I_{IL}	Input LOW Current	V_{CC} = Max., V_{IN} = 0.4 V			−0.4	mA
I_{IH}	Input HIGH Current	V_{CC} = Max., V_{IN} = 2.7 V			100	μA
I_I	I_{IH} at Max. V_{IN}	V_{CC} = Max., V_{IN} = 5.5 V			1.0	mA
I_{CC}	TTL Supply Current	V_{CC} = Max., V_{EE} = Max.	21	29	43	mA
I_{EE}	ECL Supply Current	V_{EE} = Max., V_{CC} = Max.	98	140	213	mA

Notes:
1. Devices are not subjected to ABSOLUTE MAXIMUM RATINGS in production. Non-production samples have been subjected to ABSOLUTE MAXIMUM RATINGS.
2. OPERATING RANGES are guaranteed for steady state conditions (no air flow). Hot temperature testing is elevated to simulate steady state conditions when using pulse test techniques. Cold testing is at the specified temperatures.
3. For conditions shown as Min. or Max., use the appropriate values specified under recommended operating ranges.
4. V_{IH} threshold is measured at V_{CC} = Max. and V_{EE} = Min. with all other inputs HIGH.
   V_{IL} threshold is measured at V_{CC} = Min. and V_{EE} = Max. with all other inputs LOW.
   This test method is used to guarantee V_{IH} and V_{IL}.
5. All typical values are V_{CC} = 5.0 V, V_{EE} = −5.2 V, T_A = 25°C.
6. Guaranteed with transverse air flow exceeding 500 linear F.P.M. and two minute warm-up period. Typical thermal resistance values of the package are:
   θ_{JA} (Junction-to-Ambient) = °C/Watt (still air)
   θ_{JA} (Junction-to-Ambient) = °C/Watt (at 500 F.P.M. air flow)
   θ_{JC} (Junction-to-Case) = °C/Watt

Plastic	Hermetic DIP
70	55
40	15
40	15

7. f_{CLK}, Setup, Hold Time, and DOT CLK to SOUT limits are guaranteed through characterization and correlation to other tests and not directly measured in production.

DC CHARACTERISTICS over operating range (ECL) unless otherwise specified

	Symbol	Test Conditions	−55°C	0°C	25°C	70°C	125°C	Unit
ECL Output: SOUT	V_{OH} (Max.) V_{OH} (Min.)	50 Ω to −2 V	−860 −1070	840 −1000	−810 −960	730 −910	−650 −860	mV
	V_{OL} (Max.) V_{OL} (Min.)	50 Ω to −2 V	−1690 −1900	−1665 −1870	−1650 −1850	−1630 −1835	−1570 −1800	mV
ECL Inputs: SIN, $\overline{LD}$, DOT CLK	V_{IH} (Max.) V_{IH} (Min.)	(Note 4) (Note 4)	−860 −1215	−840 −1145	−810 −1105	−730 −1055	−650 −1005	mV
	V_{IL} (Max.) V_{IL} (Min.)	(Note 4) (Note 4)	−1515 −1900	−1490 −1870	−1475 −1850	−1455 −1835	−1395 −1800	mV
	I_{IH} I_{IL}	V_{EE} = Max. V_{IN} = V_{IH} (Max.) V_{EE} = Max. V_{IN} = V_{IL} (Min.)	250 200	200 150	200 150	200 150	200 150	μA

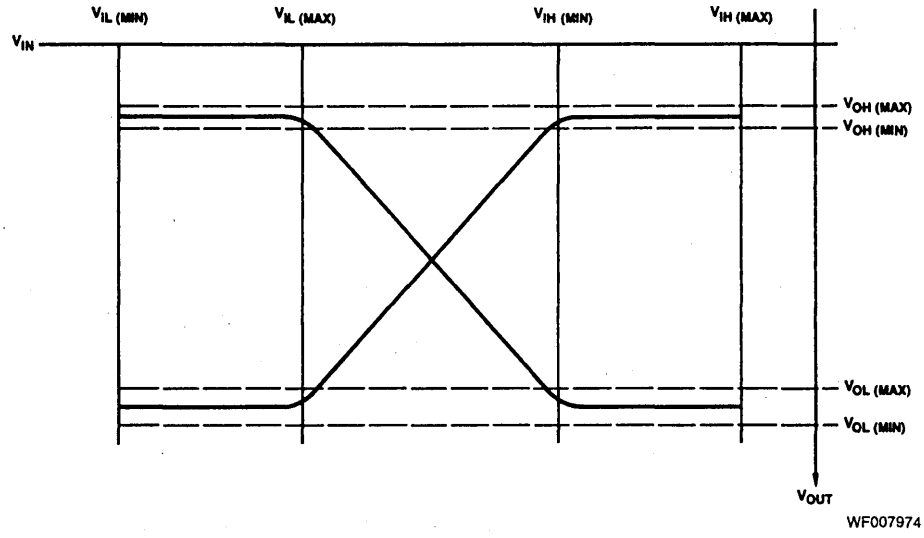


Figure 4. Am8177 ECL Specifications

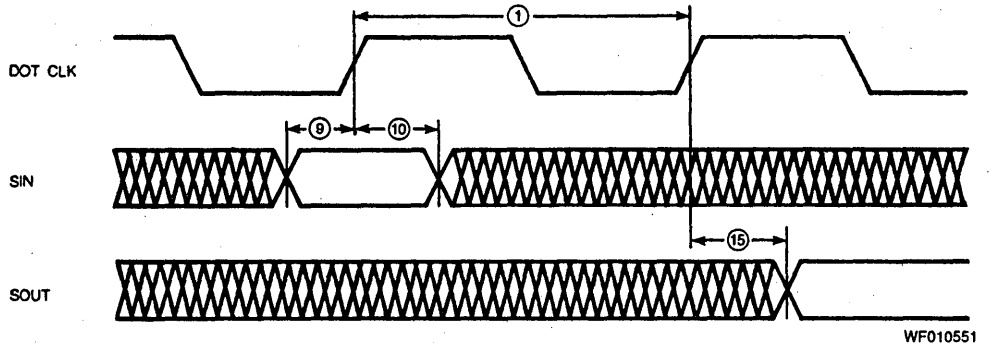
SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified (Note 7)

No.	Parameter Symbol	Parameter Description		Min.	Max.	Units
1	f_{CLK}	Clock Frequency		0	200	MHz
2	t_S	$\overline{VLE}$ ↓ to DOT CLK ↑ Setup (Transparent)		4		ns
3	t_H	$\overline{VLE}$ ↓ to DOT CLK ↑ Hold (Latch)		0		ns
4	t_W	$\overline{VLE}$ Pulse Width		5		ns
5	t_S	$DD_0 - DD_{15}$ to $\overline{VLE}$ ↑ Setup		2		ns
6	t_H	$DD_0 - DD_{15}$ to $\overline{VLE}$ ↑ Hold		4.5		ns
7	t_S	$DD_0 - DD_{15}$ to DOT CLK ↑ Setup		5		ns
8	t_H	$DD_0 - DD_{15}$ to DOT CLK ↑ HOLD		2		ns

No.	Parameter Symbol	Parameter Description		C Devices (Note 2) 0°C	25°C	70°C	E/M Devices (Note 6) −55°C	125°C	Units
9	t_S	SIN to DOT CLK ↑ Setup	Min.	0.8	0.7	0.6	1.1	0.6	ns
10	t_H	SIN to DOT CLK ↑ Hold	Min.	1.2	1.2	1.4	0.8	1.4	ns
11	t_S	$\overline{LD}$ ↓ to DOT CLK ↑ Setup (Load)	Min.	1.2	1.1	1.0	1.4	0.9	ns
12	t_S	$\overline{LD}$ ↑ to DOT CLK ↑ Setup (No Load)	Min.	0.8	0.8	0.6	1.0	0.6	ns
13	t_H	$\overline{LD}$ ↑ to DOT CLK ↑ HOLD (Load)	Min.	0.6	0.6	0.5	0.8	0.5	ns
14	t_H	$\overline{LD}$ ↓ to DOT CLK ↑ Hold (No Load)	Min.	0.5	0.4	0.3	0.7	0.3	ns
15	t_{PLH}	DOT CLK ↑ to SOUT ↓	Max.	3.3	3.6	4.0	2.8	4.2	ns
	t_{PHL}	DOT CLK ↑ to SOUT ↓	Min.	1.9	2.1	2.5	1.8	2.6	

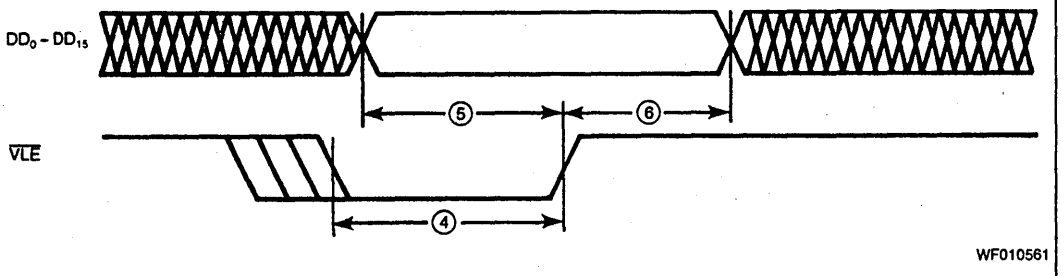
Notes: See notes following the DC Characteristics table.

SWITCHING WAVEFORMS

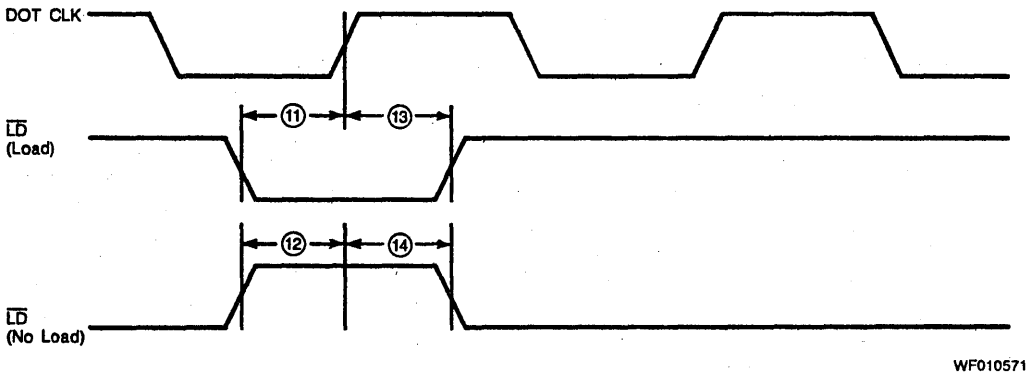


Note: $\overline{LD}$ = HIGH

Shift Timing

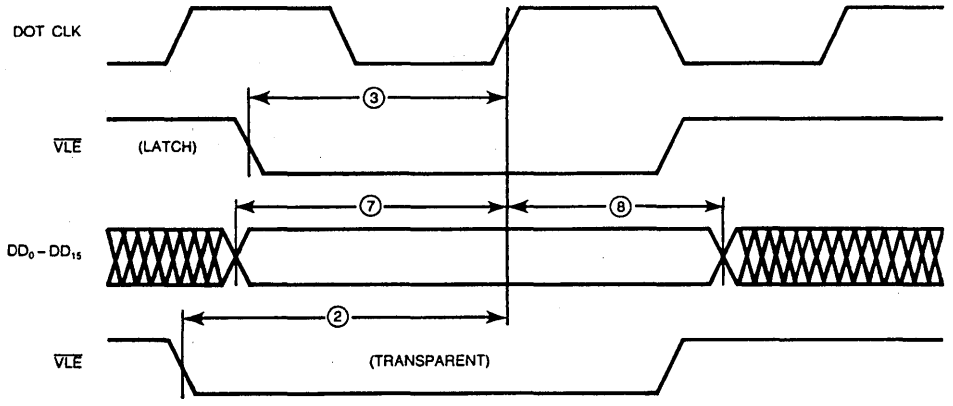


Data Latch Timing



Load and No-Load Timing

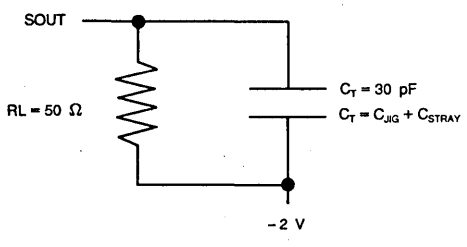
SWITCHING WAVEFORMS (Cont'd.)



WF010581

$\overline{VLE}$ LATCH /TRANSPARENT TIMING

SWITCHING TEST CIRCUIT



TC002520

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

KS000010

Am8151A

Graphics Color Palette (GCP)

FINAL

DISTINCTIVE CHARACTERISTICS

- A 256 x 8 color map and a video DAC on a single chip
- Pixel rates up to 200 MHz
- Full military range specifications
- Connects directly to singly or doubly terminated 50Ω or 75Ω RS-343A monitors
- No external compensation capacitors required
- Three Am8151As in parallel allow 256 displayable colors out of a palette of over 16 million colors
- 8-bit settling in 9.5 ns
- Glitch energy of 10 pV-sec
- Power consumption only 2/3 of the Am8151
- Plug-in replacement for Am8151

GENERAL DESCRIPTION

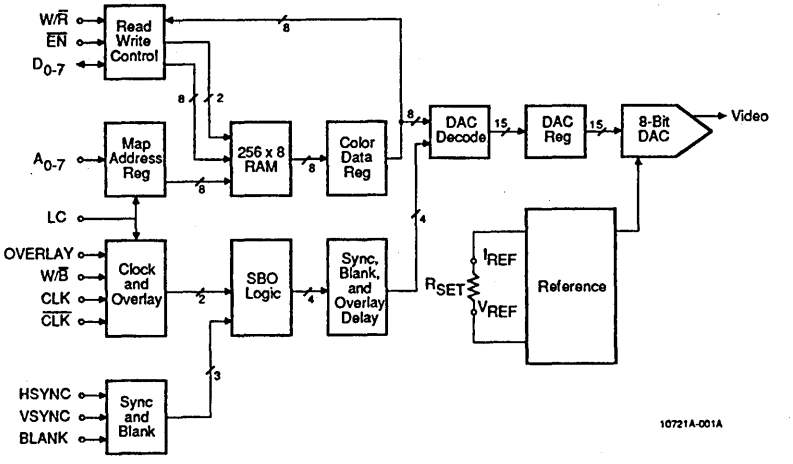
The Am8151A is a Graphics Color Palette providing a color lookup table and a video DAC for use in high-performance graphics systems. It is a low-power replacement for the original Am8151 Color Palette, featuring operation with video output loads from 25Ω to 75Ω. Logical structure of the Am8151A is identical to the Am8151 and is implemented with noise-free CML circuits to eliminate synchronous noise output from the video DAC. Three Am8151As connected in parallel provide simultaneous display of 256 colors from a palette of over 16 million colors.

The Am8151A has three levels of pipeline to allow a high pixel rate and to ease system timing requirements. The first level is prior to a 256 x 8 color lookup table in high-speed RAM. Eight bits of color address, $A_0 - A_7$ (ECL or TTL) are latched and used to select one of 256 intensities to be fed to the DAC. At the second level, the intensity selected is latched at the output of the RAM prior to being decoded to select which of the 15 current sources will be turned on. An additional high-speed register between the DAC Decoder and the DAC ensures that all DAC inputs switch simultaneously, thereby reducing the maximum duration of the glitch at the output of the DAC. This additional register adds a third level of pipeline to the pixel data path.

The color lookup table is stored in RAM and may therefore be read and writtten by a graphics processor. In addition to the 8 address lines, and 8 control lines, an 8-bit data path $D_0 - D_7$ (TTL), and two control signals $\overline{EN}$ (TTL) and W/$\overline{R}$ (TTL), are provided for this purpose.

The text capabilty of the AMD Alphanumeric Display Products or a generic overlay may be added with the OVERLAY (ECL or TTL) and W/$\overline{B}$ (ECL or TTL) inputs. A HIGH on the OVERLAY input overrides the color pixel data and drives the DAC output to peak white or reference black depending on the state of the W/$\overline{B}$ input. When three Am8151As are used in a system these pins can be connected to provide text or overlay capability in eight colors.

BLOCK DIAGRAM



TC004770

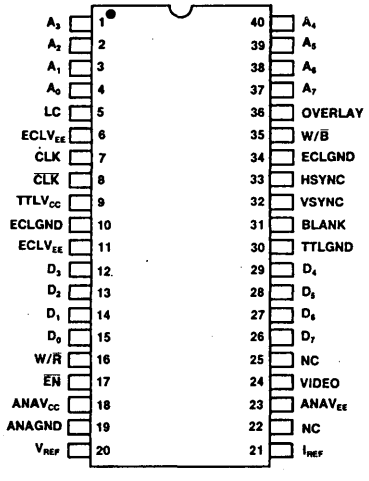
Publication # 10721 | Rev. A | Amendment /0
Issue Date: August 1988

RELATED AMD PRODUCTS

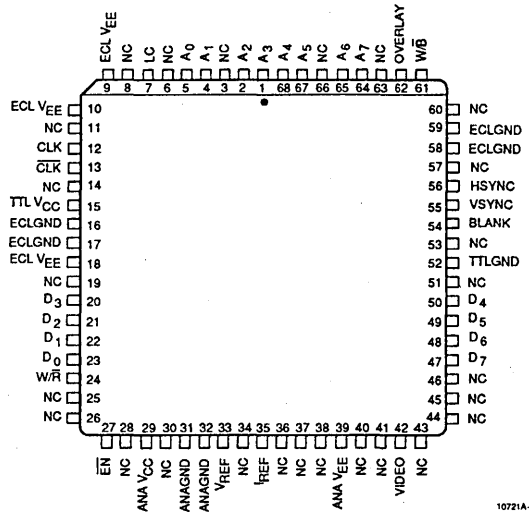
Part No.	Description
Am81C458	CMOS Color Palette
Am8172	Video Data Assembly FIFO (VDAF)
Am8177	Video Data Serializer
Am95C60	Quad Pixel Dataflow Manager

CONNECTION DIAGRAMS

Top View



CD005973

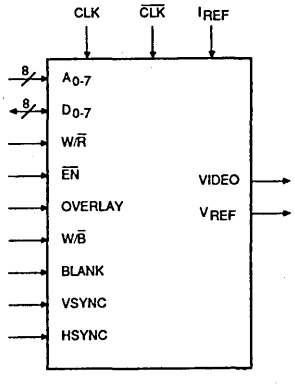


10721A-002A

CD011431

Note: Pin 1 is marked for orientation, NC = No Connection

LOGIC SYMBOL



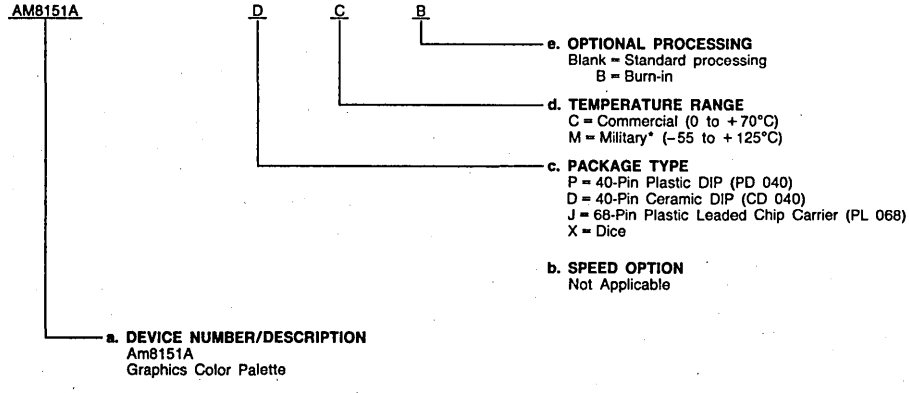
10721A-003A

LS003220

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM8151A	PC, DC, DCB, DM, DMB, JC, XC

*Military or Limited Military temperature range products are "NPL" (Non-Compliant Products List) or Non-MIL-STD-883C Compliant products only.

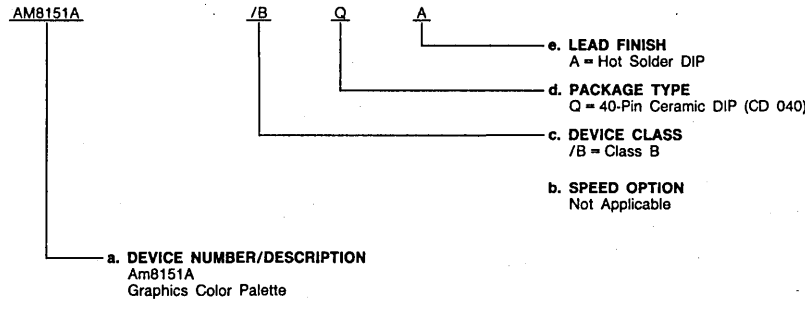
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM8151A	/BQA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

3

PIN DESCRIPTION

A_0–A_7 Address (Inputs; TTL/ECL)
These eight pins are used to address data stored in the color lookup table. The address on these pins is latched on the first rising CLK edge and decoded to select one of 256 intensities stored in the color lookup table. During a video refresh these pins should be connected to the color pixel data. During a color lookup table update these pins should be connected to the graphics processor's address bus. The logic compatability of these pins is determined by the LC pin.

D_0–D_7 Data (Inputs/Outputs; TTL)
These eight pins are used to write data in to the color lookup table or to read data out of the color lookup table. The MSB is D_7.

W/$\overline{R}$ Write/$\overline{Read}$ (Input; TTL)
The W/$\overline{R}$ controls the direction of color lookup table access by the system processor. When W/$\overline{R}$ is HIGH and $\overline{EN}$ is LOW, data is written in to the color lookup table. When W/$\overline{R}$ is LOW and $\overline{EN}$ is LOW, data is read from the color lookup table.

$\overline{EN}$ $\overline{Enable}$ (Input; TTL, Active LOW)
The $\overline{EN}$ pin is used to enable color lookup table data onto the data bus D_0–D_7 during a read operation and to enable a write into the color lookup table during a write operation. When $\overline{EN}$ is HIGH, the eight data lines D_0–D_7 are three-stated.

CLK, $\overline{CLK}$ Clock, $\overline{Clock}$ (Inputs; TTL/ECL)
CLK and $\overline{CLK}$ are the pixel clock inputs. In ECL mode these pins operate differentially. In TTL mode, $\overline{CLK}$ must be tied to ground. The clock is used internally to latch the address pins, data at the output of the color lookup table, and the decoded DAC inputs. The logic compatability of CLK is controlled by the LC pin.

OVERLAY Overlay (Input; TTL/ECL)
The OVERLAY pin, when active, overrides the color pixel data to force the DAC output to a peak white or reference black level. The level the DAC output is forced to is set by the W/$\overline{B}$ pin. The overlay signal is kept synchronized with the color pixel data inside the Am8151A by delaying the overlay signal the same number of clock cycles as the color pixel data. The logic compatibility of this pin is determined by the LC pin.

W/$\overline{B}$ White/$\overline{Black}$ (Input; TTL/ECL)
The W/$\overline{B}$ pin determines the level the OVERLAY pin will force the DAC output to. When W/$\overline{B}$ is HIGH, a HIGH on the OVERLAY pin will force the DAC output to a peak white level. When W/$\overline{B}$ is LOW, a HIGH on the OVERLAY pin will force the DAC output to a reference black level. The logic compatibility of this pin is determined by the LC pin.

BLANK Blanking (Input; TTL)
The BLANK pin, when active, overrides the color pixel and overlay data to force the DAC output to a "blacker than black" blank level. A "blacker than black" level is required during the monitor's horizontal and vertical retrace. The blank signal is kept synchronized with the pixel data inside the Am8151A by delaying the blank signal the same number of clock cycles as the pixel data.

VSYNC, HSYNC Vertical Sync Horizontal Sync (Input; TTL)
The HSYNC and VSYNC signals are internally x-ored to generate a composite sync signal. The composite sync signal, when HIGH, overrides the pixel, overlay, and blank signals to force the DAC output to a sync level. If both HSYNC and VSYNC are HIGH (composite sync is LOW) the DAC output is forced to a blank level. The composite signal is kept synchronized with the pixel data inside the Am8151A by delaying the sync signal the same number of clock cycles as the pixel data.

VIDEO Video (Output; Analog)
The VIDEO pin is the output of the DAC and is intended to directly drive monitor inputs which are singly or doubly terminated into 50 or 75Ω.

V_{REF} Voltage Reference (Output; Analog)
The V_{REF} pin provides a precision reference voltage for use in setting the full scale current of the DAC. The reference input current (I_{REF}) for the DAC, which determines the full scale current, may be generated from V_{REF} by connecting an external resistor from V_{REF} to I_{REF}. The reference resistor value may be calculated by the relation $R_{REF} = 28.44/IFS$.

I_{REF} Reference Current (Input; Analog)
A scaling current to the DAC should be provided at the I_{REF} pin. The full scale current of the DAC can be determined from the relation $IFS = 13.22\ I_{REF}$.

$TTLV_{CC}$ TTL-Positive Supply
Positive supply voltage for the TTL portions of the chip.

$ANAV_{CC}$ Analog Positive Supply
Positive supply voltage for the analog portions of the chip.

$ECLV_{EE}$ (2 pins) ECL Negative Supply
Negative supply voltage for the ECL portions of the chip.

LC Level Control (Power Suppy)
Level Control determines the logic compatability of the twelve TTL/ECL input pins. If LC is tied to V_{CC}, the logic levels are TTL. If LC is tied to ground, the logic levels are ECL.

$ANAV_{EE}$ Analog Negative Supply
Negative supply voltage for the analog portions of the chip.

TTLGND TTL Ground
Ground for the TTL portions of the chip.

ECLGND (2 pins) ECL Ground
Ground for the ECL portions of the chip.

ANAGND Analog Ground
Ground for the analog portions of the chip.

FUNCTIONAL DESCRIPTION

The Am8151A is a Graphics Color Palette (GCP) providing a color lookup table and a video DAC for use in high performance graphics systems. The Am8151A is pipelined with digital color pixel data, overlay, blank, and sync inputs entering the pipeline, and an analog signal exiting 3 CLK cycles later. The three levels of pipeline are prior to the 256 x 8 RAM color lookup table, the DAC decoder, and the 15 current sources to generate an analog signal.

Color Lookup Table

Eight lines of color pixel data are read through the pins $A_0 - A_7$ and latched into the Address Registers on the first rising edge of the CLK. $A_0 - A_7$ may be at either ECL or TTL logic levels. The LC pin is used to select the logic compatibility of these pins. These eight lines ($A_0 - A_7$) are used as an address for the color lookup table and are decoded to select one of the 256 intensities stored in the color lookup table. Each intensity stored is 8 bits wide, with 255 corresponding to reference white and 0 corresponding to reference black. On the next

rising CLK edge the intensity is latched into the color data registers to be decoded for the DAC. On a third rising CLK edge the decoded DAC inputs are latched and used to turn on or off the current sources making up the DAC.

Color Lookup Table Update

The color lookup table may be loaded and read back by the graphics processor. For this purpose 8 bidirectional data lines ($D_0 - D_7$) have been provided. D_7 is the Most Significant Bit (MSB) and D_0 is the Least Significant Bit (LSB) of data. In addition to the 8 data lines, 2 control lines are provided ($\overline{EN}$ and W/$\overline{R}$). $\overline{EN}$ is an active LOW input that selects the chip for both write and read operations. When $\overline{EN}$ is HIGH, the 8 bidirectional data lines are three-stated. W/$\overline{R}$ controls the direction of the operation. If this pin is LOW, color lookup table data is read from the table and placed on the data lines $D_0 - D_7$. If W/$\overline{R}$ is HIGH, data is read from the 8 data lines and written in to the color lookup table. For both the read and write operations the address of the data stored in the color lookup table is taken from the eight address lines ($A_0 - A_7$). Because both the address inputs and the outputs of the color lookup table are latched, the clock must be left running during an update. Time must be allowed for the address to be latched before beginning a write cycle and for the address and then data to be latched before ending a read cycle. To insure the update cycle does not interfere with the screen refresh, modifications to the color lookup table should occur while blank is active. The ten additional lines ($D_0 - D_7$, $\overline{EN}$, and W/$\overline{R}$) used for a color lookup table update are all TTL-compatible.

Overlay

Some graphics systems require a separate bit plane in addition to the color bit planes. An example of this might be separate hardware and a separate bit plane to handle text processing. The Am8151A provides two pins (OVERLAY and W/$\overline{B}$) which override the color pixel data to provide an additional video source. If OVERLAY is HIGH, the pixel data on $A_0 - A_7$ is overridden. W/$\overline{B}$ selects the intensity of the overlay. If W/$\overline{B}$ is HIGH, the DAC output will be at the Peak White level (10% brighter than Reference White. If W/$\overline{B}$ is LOW, the DAC output will be at the Reference Black level. OVERLAY and W/$\overline{B}$ may be at either TTL or ECL logic levels. The LC pin is used to select the logic compatibility of these pins. The OVERLAY and W/$\overline{B}$ signals are delayed the same number of clock cycles as the color pixel data before being fed into the DAC Decoder to keep overlay and color pixel data synchronized.

Blank

During horizontal and vertical retrace, pixel data should be ignored and the intensity output of the DAC should be driven to a "blacker than black" blank state. This is done by means of the BLANK input. BLANK is TTL-compatible and latched on the same clock as the pixel data. The BLANK signal is delayed the same number of clock cycles as the pixel data before being fed into the DAC Decoder to keep the BLANK signal and the pixel data synchronized. BLANK, when active, overrides the data and overlay inputs to drive the DAC output to the blank level.

Composite Sync

In some systems, the monitor control signals HSYNC and VSYNC are mixed with the Red, Green, and Blue signals. These control signals synchronize the monitor sweep oscillators to the R, G and B signal information. The Am8151A provides the necessary circuitry to mix these signals with the pixel information. Two inputs are provided, HSYNC and VSYNC, which are combined to generate the composite sync. These inputs are TTL-compatible and are latched with the same clock as the pixel data. Internally to the chip, HSYNC and VSYNC are XORed to generate a composite sync signal. The composite sync signal is generated in this manner to provide inverted HSYNC pulses during the much longer VSYNC pulse. This prevents the horizontal oscillator from losing synchronization during a vertical retrace and causes the horizontal oscillator to change phase by the width of HSYNC. The composite sync signal is delayed the same number of cycles as the pixel data and then, if active, overrides the data, OVERLAY, and BLANK signals to drive the output to the composite sync level.

Voltage Reference

The Am8151A provides an on-chip precision voltage supply between the V_{REF} and I_{REF} pins. A scaling current is generated at the I_{REF} input by connecting an external R_{SET} resistor from V_{REF} to I_{REF}. The SET value can be calculated from $R_{SET} = 28.44/IFS$, where IFS is the nominal Sync level output current. Nominal Reference Black output of −714 mV is obtained with R_{SET} = 1000 ohms and RT = 37.5 ohms. For these values, the Am8151A will have a Reference Black output level between −728 mV and −700 mV for operation over specified temperature range and ±10% supplies. The output drive current of the Am8151A has been increased to 9.7 mA to allow operation of three Am8151A's with 667-ohm resistors connected to their respective I_{REF} inputs for doubly terminated 50-ohm loads.

The Am8151A provides the I_{REF} input to scale the VIDEO output. The relationship between I_{REF} and IFS is IFS = 13.22 I_{REF}. The reference amplifier in the Am8151A is internally compensated, eliminating the need for external bypass capacitors. In addition, the analog V_{EE} may be connected directly to the ECLV_{EE} supply. Analog V_{CC} should be bypassed separately from TTLV_{CC}. The Am8151A has no connections on pins 22 and 25 (the Am8151 COMP1 and COMP2 pins).

DAC and DAC Decoder

The VIDEO output of the Am8151A is obtained by switching identical multiples of the 255 gray scale LSB currents. Blank and Peak White outputs are 28 LSBs each, adding 0.71 mV to VIDEO output; and Sync is 112 LSBs. All output switching occurs at the third pipeline clock. Blank and Peak White currents are delay-matched grey scale data. The video gray scale is generated by decoding the three most significant bits into seven 32 LSB equal segments and binary scaling the five LSBs. The reference feedback current, I_{REF}, is two 16 LSB currents matched with D4 for a total of 32 LSBs.

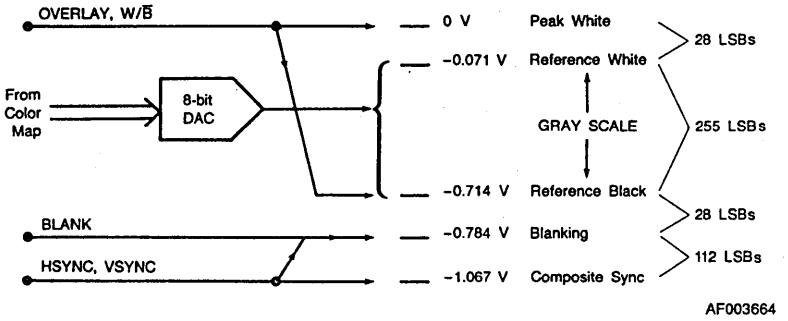
The twelve gray scale switches (seven MSB segments and five LSB segments) and the three control switches are operated from glitch-free internal CML differential logic gates which eliminate current spikes characteristic of normal Bipolar Emitter Followers. All logic in the Am8151A, as well as the output switches, use glitch-free CML to reduce internal current spikes synchronous with CLK transitions. As a result, the clock feedthrough in the Am8151A is less than 4 pV-sec for ECL operating mode. The glitch energy output of the Am8151A is reduced to 10 pV-sec which allows true 8-bit operation at up to 125 MHz clock rate. The low glitch energy error produced at the Am8151A VIDEO output is a result of matching the five LSB's switching to the seven MSB segments within 0.2 ns. The elimination of external compensation capacitors for the Reference Amplifier also contribute to the reduction in Glitch Energy Output.

The following truth table lists the nominal DAC output for all combinations of inputs assuming a reference current of 1.076 mA. These output levels are obtained with R_{SET} = 2000 ohms and R_T = 75 ohms.

TRUTH TABLE

W/$\overline{B}$	OVERLAY	BLANK	HSYNC	VSYNC	Data	Current	Voltage into 75 Ω	Level	Cum LSBs
1	1	0	0	0	X	0 mA	0 mV	Peak White	0
X	0	0	0	0	255	.94 mA	−71 mV	Reference White	28
						•			
						•	255 Equal Steps		
						•			
X	0	0	0	0	0	9.52 mA	−714 mV	Reference Black	283
0	1	0	0	0	X	9.52 mA	−714 mV	Reference Black	283
X	X	1	0	0	X	10.46 mA	−784 mV	Blanking	311
X	X	X	1	1	X	10.466 mA	−784 mV	Blanking	311
X	X	X	0	1	X	14.22 mA	−1067 mV	Composite Sync	423
X	X	X	1	0	X	14.22 mA	−1067 mV	Composite Sync	423

Am8151A DAC OUTPUT LEVELS



AF003664

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature Under Bias ..... −55 to +125°C
Supply Voltage to Ground Potential
Continuous (TTL V_{CC} and $ANAV_{CC}$) ..... −0.5 to +7.0 V
Supply Voltage to Ground Potential
Continuous (ECL V_{EE} and $ANAV_{EE}$) ..... +0.5 to −7.0 V
DC Input Voltage (TTL) ..... −0.5 to 5.5 V
DC Input Current ..... −30 to +5.0 mA
DC Input Voltage (LC Controlled) ..... −2 V to 5.5 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commericial (C) Devices
Ambient Temperature (T_A) ..... 0°C to +70°C
Supply Voltage (V_{CC}) ..... +4.5 V to +5.5 V
Supply Voltage (V_{EE}) ..... −5.72 V to −4.68 V

Military (M) Devices
Case Temperature (T_C) ..... −55°C to +125°C
Supply Voltage (V_{CC}) ..... +4.5 V to +5.5 V
Supply Voltage (V_{EE}) ..... −5.72 V to −4.68 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

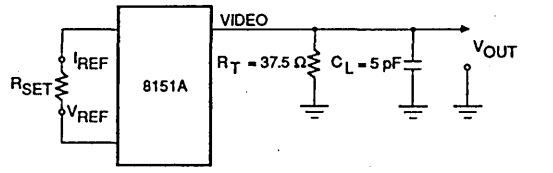
DC CHARACTERISTICS over operating ranges (TTL) (for APL Products, Group A Subgroups 1, 2, 3, 7, 8 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions (Note 1)	Min.	Typ. (Note 2)	Max.	Unit
V_{IH}	Input HIGH Level	Guaranteed Input Logical HIGH Voltage for All Inputs (Note 9)	2.0			V
V_{IL}	Input LOW Level	Guaranteed Input Logical LOW Voltage for All Inputs (Note 9)			0.8	V
I_{IL}	Input LOW Current	V_{CC} = Max., V_{IN} = 0.4 V		−0.01	−0.4	mA
I_{IH}	Input HIGH Current	V_{CC} = Max., V_{IN} = 2.7 V		.01	40	μA
I_I	Input HIGH Current	V_{IN} = V_{CC} = 5.25 V			1	mA
V_{OH}	Output HIGH Voltage	V_{CC} = Min., V_{IN} = V_{IH} = V_{IL} I_{OH} = −400 μA	2.4	3.3		V
V_{OL}	Output LOW Voltage	V_{CC} = Min., V_{IN} = V_{IH} = V_{IL} I_{OL} = 8 mA		0.3	0.5	V
I_{OS}	Output Short Circuit Current (Note 3)	V_{CC} = Max.	−20	−30	−70	mA
$TTLI_{CC}$	Power Supply Current	$TTLV_{CC}$ = Max.		33	55	mA
$ANAI_{CC}$	Power Supply Current	$ANAV_{CC}$ = Max.		7.1	10	mA
$ECLI_{EE}$	Power Supply Current	$ECLV_{EE}$ = Max.		195	300	mA
$ANAI_{EE}$	Power Supply Current	$ANAV_{EE}$ = Max. R_{SET} = 667 Ω		51	60	mA
LCI_{CC}	Logic Control Supply Current	LC = Max.		15	25	mA

ECL CHARACTERISTICS (Notes 4 & 5)

Parameter Description	Parameter Symbol	Test Conditions	−55°C	0°C	25°C	70°C	125°C	Unit
ECL Inputs	V_{IH} (Max.)	(Note 9)	−860	−840	−810	−730	−650	mV
	V_{IH} (Min.)		−1215	−1145	−1105	−1045	−1005	
	V_{IL} (Max.)	(Note 9)	−1590	−1565	−1550	−1525	−1470	mV
	V_{IL} (Min.)		−1900	−1870	−1850	−1830	−1800	
	I_{IH} (Max.)	V_{EE} = Max. V_{IN} = V_{IH} (Max.)	250	200	200	200	200	μA
	I_{IL} (Max.)	V_{EE} = Max. V_{IN} = V_{IL} (Min.)	200	150	150	150	150	μA

Notes:
1. For conditions shown as Min. or Max. use the appropriate value specified under recommended operating range.
2. Typical values are for $TTLV_{CC}$ and $ANAV_{CC}$ = 5.0 V, $ECLV_{EE}$ and $ANAV_{CC}$ = −5.2 V, LC = 5.0 V or 0 V as appropriate, T_A = 25°C.
3. Not more than one output should be shorted at a time. Duration of short not to exceed one second.
4. With airflow ⩾ 500 lfpm and two-minute warmup.
5. A combination of skewing the limits and adjusting the pulse test ambient temperature is used to ensure that the data sheet steady state limits are met at the ambient temperatures specified.
6. Trimmable to −714 mV
7. Gray scale is defined as output levels between reference white and reference black, −71 mV to −714 mV. For these conditions the absolute value of 0.5 LSB = 1.26 mV.
8. Clock feedthrough for rising or falling input may be measured with either constant memory address or constant memory data. For the Am8151A it is expressed as an Energy error with units of pV−S, consistent with glitch energy.
9. Not all tests are being performed in manufacturing. Tests are guaranteed by Engineering characterization.
10. Tests are performed in manufacturing under worst temperature conditions.



TC004780

DAC SPECIFICATIONS over COMMERCIAL operating range

Parameter Symbol	Parameter Description	Test Conditions		Min.	Typ.	Max.	Unit
	Resolution	Gray Scale (Note 6)		8	8	8	bits
	Linearity	Gray Scale (Note 7)			±0.15	±.5	LSB
	Differential Linearity	Gray Scale (Note 7)			±.05	±.5	LSB
V_{OC}	Output Compliance Voltage	$I_O \leq I_{MAX}$ (Note 9)		±1.5	±2.2		V
I_{ZS}	Zero Scale Current				.01	1	LSB
$I_{Max.}$	Output Maximum Current			42.7	60		mA
PSS IFS±	Power Supply Sensitivity, Full Scale Positive	V_{CC} = 5 V ±10% V_{EE} = −5.2 V ±10%			±0.1	±0.5	%/% Δ V
	Glitch Energy Feedthrough	LC = GND, CLK ≤ Max.			±10		pV-sec
	Output Capacitance				10		pF
I_{REF}	DAC Reference Current					3.4	mA
CLK	Clock Frequency	LC = GND, ECL Mode		D.C	250	200	MHz
		LC = V_{CC}, TTL Mode		D.C		83	
t_R	Risetime: 10 to 90% (Note 9)	R_{SET} = 1000Ω R_T = 37.5Ω			1.8	2.3	ns
t_F	Falltime: 90 to 10% (Note 9)	R_{SET} = 1000Ω R_T = 37.5Ω			1.55	2.1	ns
t_S	Setting Time (255 Level of Gray)	% Gray Scale (Notes 7 & 9)	Bits Accuracy				
		±0.2	8		9.5		
		±0.4	7		5.2		
		±0.8	6		3.2		
		±1.6	5		2.8		
		±3.2	4		2.4		ns
V_{OB}	Reference Black Output (Note 6)	R_{SET} = 1000Ω R_T = 37.5Ω		−728	−714	−700	mV
I_{FSTC}	Output Current Temp Coefficient				±20		ppm/°C
V_{REF}	DAC Reference Voltage	$I_O \leq$ 4 mA V_{CC} = 5.0 V V_{EE} = −5.2 V		2.122	2.152	2.177	V
	Reference Voltage Line Regulation	ΔV_{CC} = ±10% ΔV_{EE} = ±10%			±0.13	±0.4	% V_{REF}
	Reference Voltage Load Regulation	3.2 mA ≤ I_O ≤ 9.7 mA			±.06	±0.4	% V_{REF}
	Reference Voltage Temperature Coeffiecient				±15		ppm/°C
CLK_n	Clock Feedthrough (Note 8)	LC = GND CLK ≤ Max.			±4		pV-S

Notes: See notes following ECL Characteristics

DAC SPECIFICATIONS over **MILITARY** operating range (for APL Products, Group A, Subgroups 1, 2, 3, 7, 8, 9, 10, 11 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions		Min.	Typ.	Max.	Unit
	Resolution	Gray Scale (Note 6)		8	8	8	bits
	Linearity	Gray Scale (Note 7)			±0.15	±1	LSB
	Differential Linearity	Gray Scale (Note 7)			±.05	±1	LSB
V_{OC}	Output Compliance Voltage	$I_O \leq I_{MAX}$ (Note 9)		±1.5	±2.2		V
I_{ZS}	Zero Scale Current				.01	1	LSB
$I_{Max.}$	Output Maximum Current			40	60		mA
PSS IFS±	Power Supply Sensitivity, Full Scale Positive	V_{CC} = 5 V ±10% V_{EE} = −5.2 V ±10%			±0.1	±0.5	%/% Δ V
	Glitch Energy Feedthrough	LC = GND, CLK ≤ Max.			±10		pV-sec
	Output Capacitance				10		pF
I_{REF}	DAC Reference Current					3.4	mA
CLK	Clock Frequency	LC = GND, ECL Mode		D.C	250	160	MHz
		LC = V_{CC}, TTL Mode		D.C		83	
t_R	Risetime: 10 to 90% (Note 9)	R_{SET} = 1000Ω R_T = 37.5Ω			1.8	2.5	ns
t_F	Falltime: 90 to 10% (Note 9)	R_{SET} = 1000Ω R_T = 37.5Ω			1.55	2.3	ns
t_S	Setting Time (255 Level of Gray)	% Gray Scale (Notes 7 & 9)	Bits Accuracy				
		±0.2	8		9.5		
		±0.4	7		5.2		
		±0.8	6		3.2		
		±1.6	5		2.8		
		±3.2	4		2.4		ns
V_{OB}	Reference Black Output (Note 6)	R_{SET} = 1000Ω R_T = 37.5Ω		−728	−714	−700	mV
I_{FSTC}	Sync Output Current Temp Coefficient				±20		ppm/°C
V_{REF}	DAC Reference Voltage	$I_O \leq$ 4 mA V_{CC} = 5.0 V V_{EE} = −5.2 V		2.122	2.152	2.177	V
	Reference Voltage Line Regulation	ΔV_{CC} = ±10% ΔV_{EE} = ±10%			±0.13	±0.4	% V_{REF}
	Reference Voltage Load Regulation	3.2 mA ≤ I_O ≤ 9.7 mA			±.06	±0.4	% V_{REF}
	Reference Voltage Temperature Coeffiecient				±15		ppm/°C
CLK_n	Clock Feedthrough (Note 8)	LC = GND CLK ≤ Max.			±4		pV-S

Notes: See notes following ECL Characteristics

EXPLANATION OF DAC SPECIFICATIONS

Resolution:
Resolution refers to the number of discrete steps or levels which the DAC can provide, and is expressed as a number of bits. A DAC with n bits of resolution provides 2^n discrete analog levels. For the Am8151A the gray scale code between Reference Black and Reference White has 255 codes.

Linearity:
Linearity is the maximum deviation of an actual output from its ideal value. For the Am8151A, linearity is expressed as a fraction of an LSB.

Differential Linearity:
Differential Linearity is the measure of the difference between any two code values. For the Am8151A, differential linearity is expressed as a fraction of an LSB.

Output Compliance Voltage:
The DC voltage output for which the Am8151A will meet its Output Current, Linearity, and Differential Linearity specifications.

Zero Scale Current:
The output current of the Am8151A at Peak White output level.

Full Scale Current: (I_{FS}):
The output current at Sync output level. $I_{FS} \approx 13.22\ I_{REF}$.

Full Scale Current Temperature Coefficient (I_{FSTC}):
The ratio of the full scale output current to the temperature change causing it.

Power Supply Sensitivity ($P_{SS}\ I_{FS}$):
The change in the full scale output current for changes in one or more power supply voltages, expressed as a percentage of the power supply change.

Ref Black Output Level (V_{OB}):
The video voltage output of the Am8151A with $R_{SET} = 1000\ \Omega$, $R_T = 37.5\ \Omega$, and Reference Black as the output level.

Reference Current (I_{REF}):
The current into the I_{REF} pin.

Output Capacitance (C_O):
The package and DAC output capacitance of the Video pin on the Am8151A. For the ceramic DIP package C_O is 11 pF, while the plastic DIP and PLCC is 6 pF.

Output Delay:
Propagation delay from the CLK logic threshold to a 50% change in gray scale output level.

Output Settling Time:
Propagation delay from the 50% output change to within the specified percentage of final value.

Output Clock Feedthrough (CLK_n):
Synchronous video output noise. In the Am8151A, expressed as pV-S of energy.

Output Glitch Energy:
The mismatch in the energy of the undershoot and overshoot pulse for a 1 LSB transition at the Video output. In the Am8151A glitch energy is expressed in pV-S. It may also be expressed as LSB error for a given clock period. A half LSB glitch energy error at $f_{CLK} = 125$ MHz is 10.1 pV-S.

EXPLANATION OF REFERENCE VOLTAGE SPECIFICATIONS

Reference Voltage ($V_{REF} - V\ (I_{REF})$):
The voltage from the V_{REF} to the I_{REF} pin.

Line Regulation:
The change in the reference voltage produced by changes in one or more of the Am8151A power supply voltages expressed as a percentage of the reference voltage.

Load Regulation:
The change in the reference voltage produced with changes in the current in the V_{REF} pin expressed as a percentage of the reference voltage.

Reference Voltage Temperature Coefficient:
The change in the reference voltage divided by the temperature change producing it expressed in ppm/°C.

AC SWITCHING CHARACTERISTICS over COMMERCIAL operating range

No.	Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Unit
1	t_{CLK}	Clock Period (Note 9)	LC = GND, ECL Mode LC = V_{CC}, TTL Mode	5 12	4.1		ns ns
2	t_S	Address, OVERLAY, W/$\overline{B}$ Setup before Clock ↑ HSYNC, VSYNC, BLANK Setup before Clock ↑ Address, OVERLAY, W/$\overline{B}$, HSYNC, VSYNC, BLANK Setup before Clock ↑ (Note 9)	LC = GND LC = GND LC = V_{CC}	0.7 5.0 2.0	0.05 1.5 0.5		ns ns ns
3	t_H	Address, OVERLAY, W/$\overline{B}$ Hold after Clock ↑ HSYNC, VSYNC, BLANK Hold after Clock ↑ Address, OVERLAY, W/$\overline{B}$, HSYNC, VSYNC, BLANK after Clock ↑ (Note 9)	LC = GND LC = GND LC = V_{CC}	0.7 2.0 2.0	−0.05 −1.5 −0.5		ns ns ns
4	t_{PD}	Clock ↑ to 50% VIDEO change (Note 9)	LC = GND LC = V_{CC}	2.0 4.0	5.3	8 12	ns ns
5	t_{PD} ECL/TTL	Clock ↑ to Data Valid (Read) (Note 10)		5	13	30	ns
6	t_S	W/$\overline{R}$ Setup before $\overline{EN}$ ↓ (Note 9)		10	1.5		ns
7	t_H	W/$\overline{R}$ Hold after $\overline{EN}$ ↑ (Note 9)		10	1.5		ns
8	t_{PD}	$\overline{EN}$ ↓ to Data Active (Read) (Note 10)		5	11	25	ns
9	t_{PD}	$\overline{EN}$ ↑ to Data Three-State (Read) (Note 10)			7	20	ns
10	t_S	Address latched (Clock ↑) to $\overline{EN}$ ↓ Setup (Write) (Note 9)		10	4		ns
11	t_S	Data (and Address) Setup before $\overline{EN}$ ↑, Write Cycle Time (Write) (Note 9)		12	1.5		ns
12	t_H	Data Hold after $\overline{EN}$ ↑ (Note 9)		10	1.5		ns
13	t_{PD}	$\overline{EN}$ Low Pulse Width to Write (Note 9)			3		ns

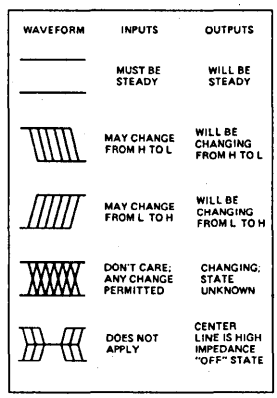
AC SWITCHING CHARACTERISTICS over MILITARY operating range (for APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted.)

No.	Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Unit
1	t_{CLK}	Clock Period (Note 9)	LC = GND, ECL Mode LC = V_{CC}, TTL Mode	6.25 12	4.1		ns ns
2	t_S	Address, OVERLAY, W/$\overline{B}$ Setup before Clock ↑ HSYNC, VSYNC, BLANK Setup before Clock ↑ Address, OVERLAY, W/$\overline{B}$, HSYNC, VSYNC, BLANK Setup before Clock ↑ (Note 9)	LC = GND LC = GND LC = V_{CC}	1.0 5.0 2.0	0.05 1.5 0.5		ns ns ns
3	t_H	Address, OVERLAY, W/$\overline{B}$ Hold after Clock ↑ HSYNC, VSYNC, BLANK Hold after Clock ↑ Address, OVERLAY, W/$\overline{B}$, HSYNC, VSYNC, BLANK after Clock ↑ (Note 9)	LC = GND LC = GND LC = V_{CC}	1.0 2.0 2.0	−0.05 −1.5 −0.5		ns ns ns
4	t_{PD}	Clock ↑ to 50% VIDEO change (Notes 9)	LC = GND LC = V_{CC}	2.0 4.0	5.3	8 12	ns ns
5	t_{PD} ECL/TTL	Clock ↑ to Data Valid (Read) (Note 10)		5	13	30	ns
6	t_S	W/$\overline{R}$ Setup before $\overline{EN}$ ↓ (Note 9)		10	1.5		ns
7	t_H	W/$\overline{R}$ Hold after $\overline{EN}$ ↑ (Note 9)		10	1.5		ns
8	t_{PD}	$\overline{EN}$ ↓ to Data Active (Read) (Note 10)		5	11	25	ns
9	t_{PD}	$\overline{EN}$ ↑ to Data Three-State (Read) (Note 10)			7	20	ns
10	t_S	Address latched (Clock ↑) to $\overline{EN}$ ↓ Setup (Write) (Note 9)		10	4		ns
11	t_S	Data (and Address) Setup before $\overline{EN}$ ↑, Write Cycle Time (Write) (Note 9)		12	1.5		ns
12	t_H	Data Hold after $\overline{EN}$ ↑ (Note 9)		10	1.5		ns
13	t_{PD}	$\overline{EN}$ Low Pulse Width to Write (Note 9)			3		ns

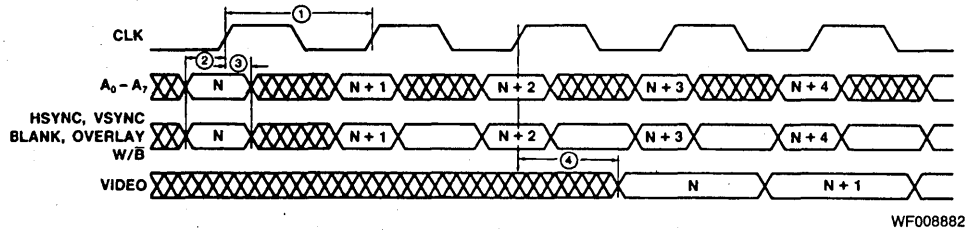
Notes: See notes following ECL Characteristics

SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS



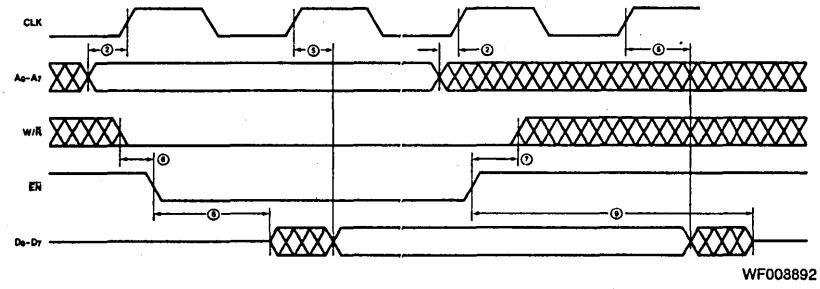
KS000010



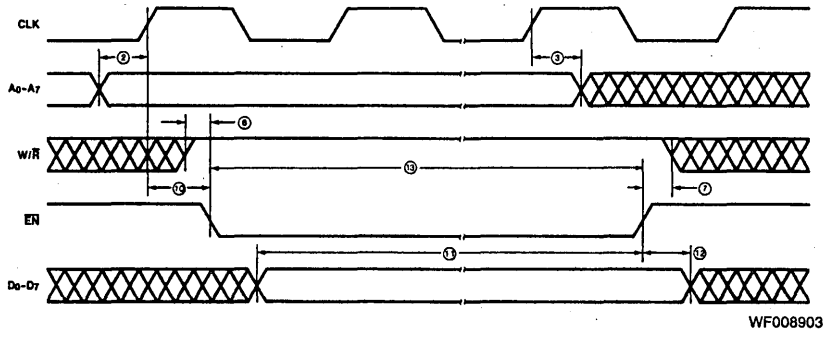
WF008882

Video Refresh Timing

SWITCHING WAVEFORMS (Cont'd.)

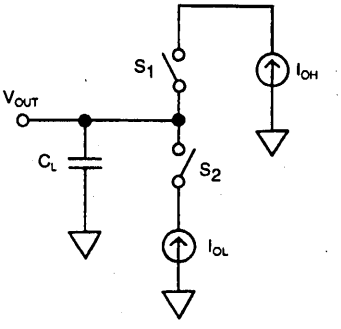


Read Timing



Write Timing

SWITCHING TEST CIRCUIT

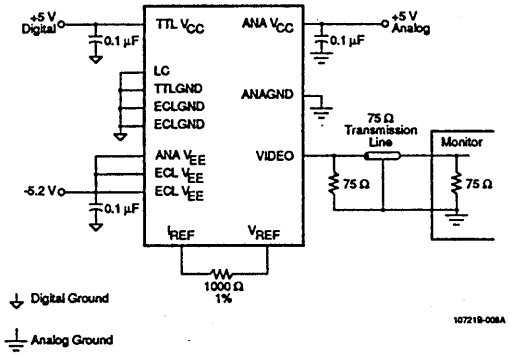


TC003132

A. Outputs

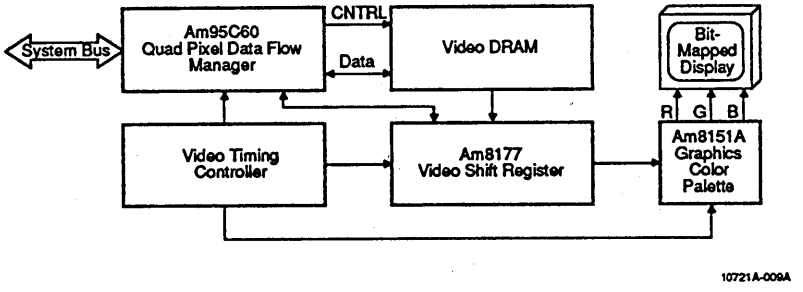
Notes: 1. C_L = 50 pF. The load capacitance includes scope probe, wiring, and stray capacitance without the device in the test fixture.
2. S_1 and S_2 are open during all DC and functional testing.
3. During AC testing, switches are set as follows:
   1) For $V_{OUT} > 1.5$ V, S_1 is closed and S_2 open
   2) For $V_{OUT} < 1.5$ V, S_1 is open and S_2 closed

TYPICAL CONNECTION DIAGRAM



CD011610

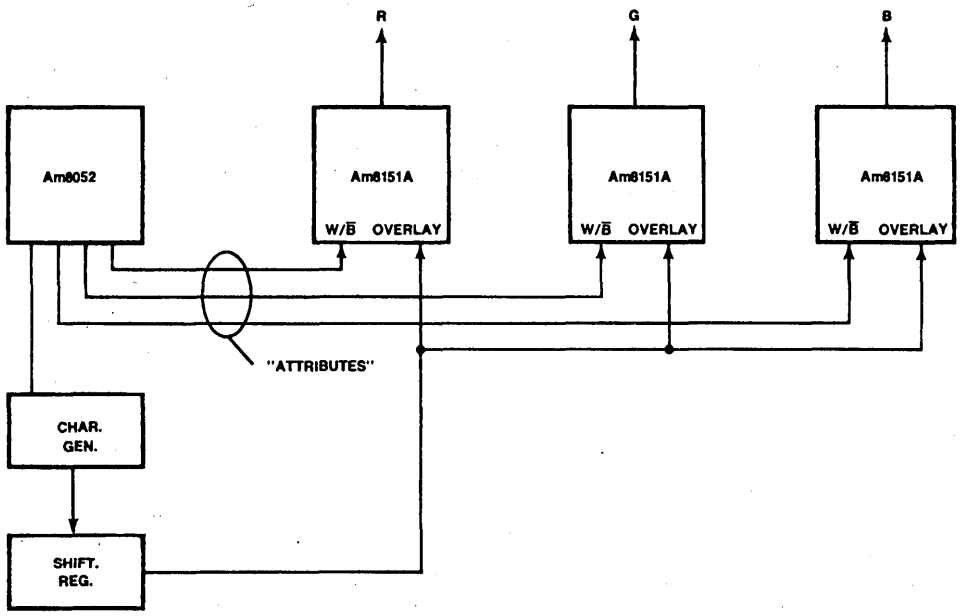
TYPICAL APPLICATION



BD008170

3

GENERATING 8 COLOR TEXT OVERLAY WITH W/$\overline{B}$ AND OVERLAY INPUTS



AF003721

SINGLE Am8151A TRUTH TABLE

Overlay	W/$\overline{B}$	Output
0	X	Graphics
1	1	Peak White
1	0	Ref. Black

TRUTH TABLE FOR THREE Am8151As

	W/$\overline{B}$			
Overlay	(R)	(G)	(B)	Output
0	X	X	X	Graphics
1	0	0	0	Black
1	0	0	1	Blue
1	0	1	0	Green
1	0	1	1	Cyan
1	1	0	0	Red
1	1	0	1	Magenta
1	1	1	0	Yellow
1	1	1	1	White

Preliminary

Am81C176

CMOS Color Palette

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Plug-in Replacement for Inmos G171 and G176
- VGA hardware and software compatible
- Clock rates up to 80 MHz
- Available in 28-pin DIP and 32-pin PLCC package
- 256 x 18 Color Look-Up Table (LUT)
- Triple 6-bit Digital-to-Analog Converters (DACs)
- RS-170A compatible RGB outputs
- External current reference
- Asynchronous MPU interface
- Single monolithic, high-performance CMOS
- Single +5 V power supply

GENERAL DESCRIPTION

The Am81C176 is a monolithic CMOS Color Palette and is hardware and software compatible with the VGA standard. Applications include high-resolution color graphics, CAD/CAM/CAE, and desktop publishing. The Am81C176 operates at speeds up to 80 MHz and can support monitors with resolutions up to 1024 x 768.

The Am81C176 has a 256 x 18 Look-Up Table and triple 6-bit DACs. It can simultaneously display 256 colors out of an available set of 256K colors.

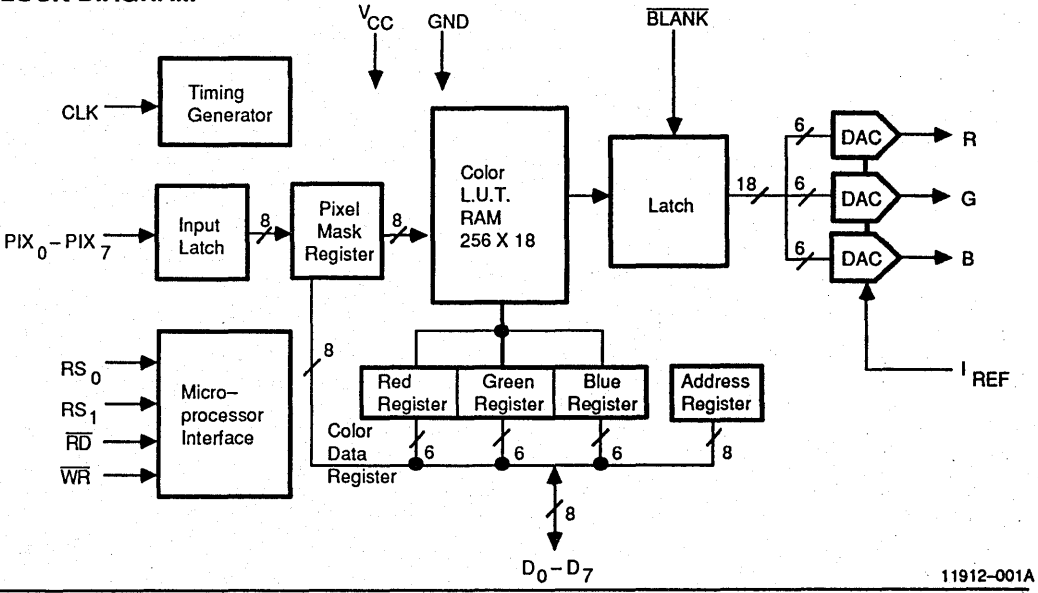
Because of a proprietary technique, read and write operations to the Color look-up table may occur during active video.

The Am81C176 generates RS-170A compatible outputs into doubly-terminated 75Ω loads, without external buffers.

The Am81C176 is fabricated using AMD's state-of-the-art 1.2µ CMOS process. The device is available in a 28-lead DIP and 32-lead PLCC package. It is pin- and functionally-compatible with the Inmos IMS G171 and IMS G176.

3

BLOCK DIAGRAM

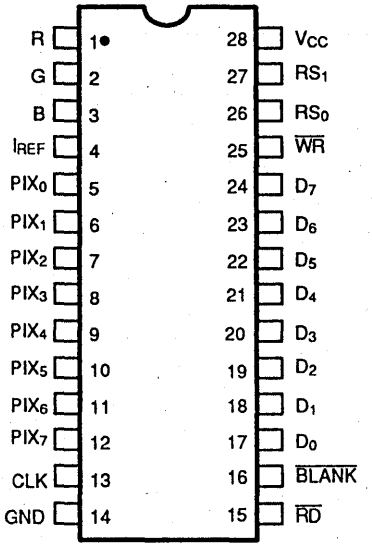


11912-001A

Publication # 11912 Rev. A Amendment /0
Issue Date: April 1989

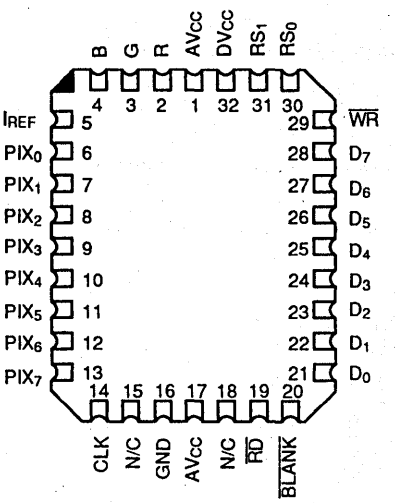
CONNECTION DIAGRAMS
Top View

28-PIN DIP



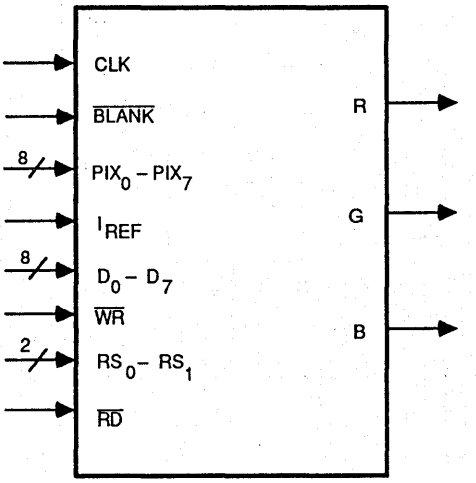
11912-002A

32-PIN PLCC



11912-003A

LOGIC SYMBOL

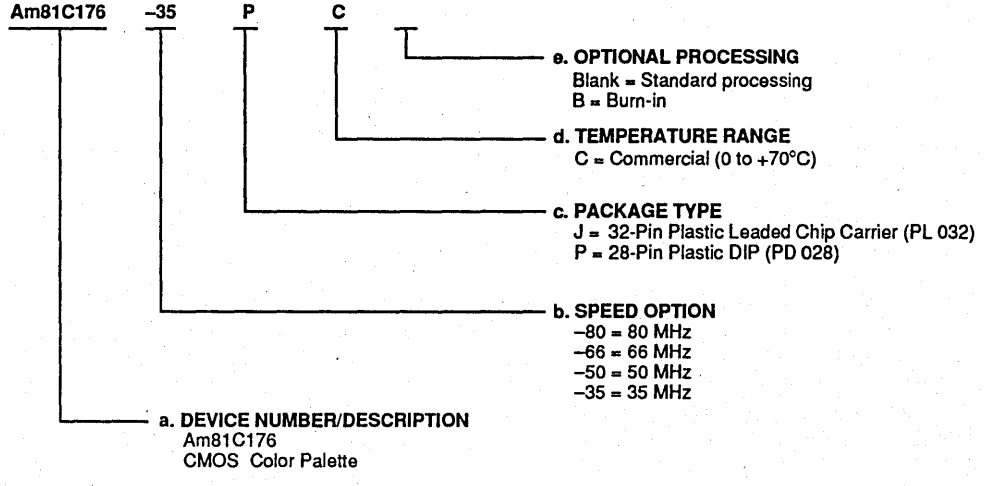


11912-004A

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Am81C176-80 Am81C176-66 Am81C176-50 Am81C176-35	JC, PCB, PC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Timing Section

CLK

Clock source pin (TTL compatible input)

This input is the pixel clock of the video system and is to be driven by a dedicated TTL buffer. The rising edge of CLK latches the $\overline{BLANK}$ and $PIX_0 - PIX_7$ inputs and also controls the flow of these signals through the pipeline stages of the Color Palette and DACs to the R, G, and B outputs.

$\overline{BLANK}$

Blank (TTL compatible input)

The $\overline{BLANK}$ input, when active, overrides the pixel data to force the R, G, and B outputs to their blank levels. This blank level is required during the monitor vertical and horizontal retrace times. It is latched on the rising edge of CLK. Typically, blank time is used to update the Color-Look-up Table through $D_0 - D_7$.

Bit Map Interface Section

$PIX_0 - PIX_7$

Color Pixel Data addresses (TTL compatible inputs)

These 8 inputs select which of the 256 entries in the Color Look-Up Table is to be used to provide pixel color information. These inputs run at the pixel rate of the system and are latched on the rising edge of CLK. PIX_0 is the least significant bit.

MPU Interface Section

$D_0 - D_7$

Data and address bus (TTL compatible bi-directional)

These 8 pins are used by the host microprocessor to write to (with $\overline{WR}$ low) and read from (with $\overline{RD}$ low) the internal registers (Pixel Mask Register, Pixel Address Register, and Color Data Register). D_0 is the least significant bit.

During write cycles, the rising edge of $\overline{WR}$ latches data from the $D_0 - D_7$ inputs into the register selected by the $RS_0 - RS_1$ inputs. During read cycles, $\overline{RD}$ drives the $D_0 - D_7$ lines from the register selected by $RS_0 - RS_1$. The end of a read cycle is determined by the rising edge of $\overline{RD}$.

When both $\overline{WR}$ and $\overline{RD}$ are a logical one, the $D_0 - D_7$ pins go into three-state.

$\overline{WR}$

Write Control Input (TTL compatible input)

$\overline{WR}$ is the control signal used for writing data into internal registers. $\overline{WR}$ must be a logical zero to write data to the internal registers. During Write operations, $RS_0 - RS_1$ are latched on the falling edge of $\overline{WR}$ and $D_0 - D_7$ are latched on the rising edge of $\overline{WR}$. When active, information on the external data bus is available to the $D_0 - D_7$ inputs.

$\overline{RD}$

Read Control Input (TTL compatible input)

$\overline{RD}$ must be a logical zero to read data from the internal registers. During Read operations, $RS_0 - RS_1$ are latched on the falling edge of $\overline{RD}$. When active, information on the internal data bus is available to the $D_0 - D_7$ pins.

$RS_0 - RS_1$

Register Select Inputs (TTL compatible inputs)

$RS_0 - RS_1$ allow the MPU to select any of the internal registers. These inputs determine the type of read or write operation being performed. See Table 1.

Analog Output Section

R

Red video output (Analog output)

Analog output of the red DAC. This output is capable of driving an RS-170A compatible doubly-terminated 75Ω cable.

G

Green video output (Analog output)

Analog output of the green DAC. This output is capable of driving an RS-170A compatible doubly-terminated 75Ω cable.

B

Blue video output (Analog output)

Analog output of the blue DAC. This output is capable of driving an RS-170A compatible doubly-terminated 75Ω cable.

I_{REF}

Current reference (Analog input).

I_{REF} is the reference current input. Through this pin the user provides the reference current for the DAC which, in turn, control the full-scale output currents.

$$I_{REF} = \frac{1}{2.1} * \frac{V_{white}}{R_{load} (= 37.5\Omega)}$$

Power Supply Section

V_{CC}

+5 volt supply.

DV_{CC}

+5 volt digital power supply.

AV_{CC}

+5 volt analog power supply.

GND

Ground

FUNCTIONAL DESCRIPTION

The Am81C176 CMOS color palette integrates all major functions required in the video section of a graphics system and supports pixel rates sufficient to drive monitors with resolutions up to 1024 x 768.

A programmable 256 x 18 Color Look-Up Table (LUT) maps pixel data from a bit-map memory into physical color, and three 6-bit Digital-to-Analog-Converters (DACs) convert the outputs of the Color Look-Up-Table (LUT) into RS170 compatible RGB analog format. Up to 8 bits per pixel are supported for a maximum of 256 simultaneous colors out of 256K available color combinations.

MPU Interface

The Am81C176 is designed to support a standard MPU bus interface with direct access to 256 Color Look-Up Table (LUT) locations and two control registers. The MPU interface is completely asynchronous with respect to pixel clock. However, data transfers between the LUT and Red Register, Green Register, and Blue Register (see block diagram) are internally synchronized to pixel clock. Double sampling techniques have been utilized in order to minimize metastability problems occurring when synchronizing an asynchronous event (such as $\overline{RD}$ or $\overline{WR}$) with a free running clock (such as CLK).

The Read and Write accesses to the LUT take one and two pixel clock cycles, respectively.

The nature of the MPU access is determined by the Register Select (RS_1, RS_0) inputs. RS_1 and RS_0 select among Address Register (LUT write), Address Register (LUT read), Color Data Register and Pixel Mask Register, as shown in Table 1.

Table 1. RS_0, RS_1 Decoding

RS_1	RS_0	Function
0	0	Address Register (LUT write)
0	1	Color Data Register
1	0	Pixel Mask Register
1	1	Address Register (LUT read)

A typical ***color data write cycle*** is initiated by setting the 8-bit Address Register (LUT write) with the address of the LUT into which data is to be written. Next the MPU performs three write cycles to the Color Data Register: one for red, one for green, one for blue intensity. At the end of the blue cycle the data is concatenated into an 18-bit word and written to the LUT location pointed to by the Address Register. The Address Register is then auto-incremented to point to the next location in LUT. This process may be repeated again as required. If the user needs to access consecutive LUT locations, the Address Register needs to be written to only at the beginning of the sequence. See Table 2.

A typical ***color data read cycle*** is initiated by setting the 8-bit Address Register (LUT read) with the address of the LUT to be read. At this point, 18 bits of color data are transferred from the LUT to the Red, Green and Blue portion of the Color Data Register (see block diagram) and the Address Register is auto-incremented to point to the next location in LUT. Next the MPU performs three read cycles to the Color Data Register: one for red, one for green, one for blue intensity. At the end of the blue cycle a new set of 18 bits is tranferred to the Red, Green and Blue portions of the Color Data Register, and the Address Register is again auto-incremented. This process may be repeated as required. If the user needs to access consecutive LUT locations the Address Register needs to be written to only at the beginning of the sequence. See table 2.

The 6-bit color data occupy the six least significant positions in the data bus. Bits D_6 and D_7 are ignored during write cycles and are set to 0 during read cycles. Bit D_0 is the least significant bit.

The Am81C176 uses one 8-bit ***Address Register*** to address the LUT as shown in Table 3. The Address Register resets to 0 after a blue read/write cycle to the LUT address 255. A user transparent modulo-3 counter (AR_b, AR_a) keeps track of the red, green and blue cycles and auto-increments at the end of each read/write access to the LUT. This counter is reset to zero after a write access to the Address Register, and is unchanged following a read access to the Address Register. Thus a write to the Address Register will abort any unfinished read or write sequence.

The Am81C176 uses one 8-bit ***Pixel Mask Register*** to modify the address of the LUT as provided by PIX_0 – PIX_7. The eight bits of this register are ANDed with PIX_0 – PIX_7, and the result used as the address to the LUT. This mechanism provides a quick way to alter the appearance of one or more colors on the display unit with just one MPU access, without the need for changing the bit-map memory or the LUT contents. The CPU addresses are not affected by this register.

Display Memory Interface

Pixel data PIX_0 – PIX_7 are latched on the rising edge of CLK and are used as address to the 256 locations of the LUT. The total pipeline delay from PIX_0 – PIX_7 and $\overline{BLANK}$ inputs, to R, G, B outputs is four clock cycles.

Video Generation

During each clock cycle, a 18-bit word from the LUT is presented to three DACs: 6 bits for red, 6 for green and 6 for blue. The three DACs convert the digital color memory output into RGB RS-170A analog format.

The $\overline{BLANK}$ input is latched on the rising edge of CLK. It is routed to the three DACs after a delay of four clock periods, identical to the delay incurred by the video stream.

$\overline{BLANK}$, when active, forces a zero to the input to the DACs, overriding the current LUT output.

The three analog outputs of the Am81C176 are each capable of driving a doubly terminated 75Ω coaxial cable.

Table 2. Read/Write Access to the Am81C176

$\overline{RD}$	$\overline{WR}$	RS_1	RS_0	AR_b	AR_a		Function
1	0	0	0	X	X	Write Address Register (LUT Write)	AR(7:0)←D(7:0); ARb:ARa←00.
1	0	0	1	0	0	Write Color Data Register(Red)	RREG(5:0)←D(5:0); ARb:ARa←01.
1	0	0	1	0	1	Write Color Data Register(Green)	GREG(5:0)←D(5:0); ARb:ARa←10;
1	0	0	1	1	0	WriteColor Data Register(Blue)	BREG(5:0)←D(5:0); ARb:ARa←00;
						Write Color Look-Up-Table	R(5:0)←RREG; G(5:0)←GREG; B(5:0)←BREG; INC. AR(7:0).
1	0	1	1	X	X	Write Address Register (LUT Read)	AR(7:0)←D(7:0); ARb:ARa←00;
						Read Color Look-Up Table	RREG←R(5:0); GREG←G(5:0); BREG←B(5:0); INC. AR(7:0).
0	1	0	1	0	0	Read Color Data Register (Red)	D(5:0)←RREG(5:0); D(7:6)←0; ARb:ARa←01.
0	1	0	1	0	1	Read Color Data Register (Green)	D(5:0)←GREG(5:0); D(7:6)←0; ARb:ARa←10.
0	1	0	1	1	0	Read Color Data Register (Blue)	D(5:0)←BREG(5:0); D(7:6)←0; ARb:ARa←00.
0	1	0	0	X	X	Read Address Register	D(7:0)←AR(7:0).
1	0	1	0	X	X	Write Pixel Mask Register	PMREG(7:0)←D(7:0).
0	1	1	0	X	X	Read Pixel Mask Register	D(7:0)←PMREG(7:0).

Note: Refer to timing diagrams for edge information on $\overline{RD}$ and $\overline{WR}$.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	–65 to +150°C
Ambient Temperature	
Under Bias	–55 to +125°C
Junction Temperature	+175°C
Supply Voltage to Ground	
Potential Continuous	–0.5 to 7.0 V
DC Voltage Applied to Outputs	
for HIGH Output State	–0.5 to V_{CC} +0.5 V
DC Input Voltage	–0.5 to V_{CC} +0.5 V

Stresses above those listed under "Absolute Maximum Ratings" may cause device failure. Functionality at or above these limits is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices	
Ambient Temperature (T_A)	0 to +70°C
Supply Voltage (V_{CC})	
for 80 MHz devices	+4.75 to +5.25 V
for 35, 50, 66 MHz device	+4.50 to +5.50 V
I_{REF} Current	–7 to –9 mA
Output Load	37.5Ω

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS (over operating range)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Unit
Digital Inputs						
V_{IH}	Input High Voltage		2.0		V_{CC}+0.5	V
V_{IL}	Input Low Voltage		GND-0.5		0.8	V
I_{IH}	Input High Current	V_{in}= V_{CC}			1	µA
I_{IL}	Input Low Current	V_{in}= GND			–1	µA
C_{IN}	Input Capacitance	f=1 MHz, V_{in}= 2.4 V			7	pF
Digital Outputs						
V_{OH}	Output High Voltage	I_{OH} = –5 mA	2.4			V
V_{OL}	Output Low Voltage	I_{OL} = +5 mA			.4	V
I_{OZ}	Three-State Current	GND <= V_{in} <= V_{CC}			50	µA
C_{out}	Output Capacitance				7	pF
Analog Outputs						
	Resolution (each DAC)		6	6	6	Bits
INL	Integral Linearity Error				±1/2	LSB
DNL	Differential Linearity Error				±1/2	LSB
	Full Scale Error				±5	%Gray
	Monotonicity			Guaranteed		
	Coding	Binary				Binary
	Output Voltage				1.5	V
	Output Current				21	mA
	DAC-to-DAC Matching				2	%
	Glitch Energy			120		pV-sec

PRELIMINARY

SWITCHING CHARACTERISTICS

Parameter Number	Parameter Description	Min/Typ Max					Unit
	Clock Rate		80	66	50	35	MHz
1	$RS_0 - RS_1$ Setup Time	Min	10	10	10	15	ns
2	$RS_0 - RS_1$ Hold Time	Min	10	10	10	15	ns
3	$\overline{RD}$ Asserted to Data Bus Driven	Min	5	5	5	5	ns
4	$\overline{RD}$ Asserted to Data Valid	Max	40	40	40	40	ns
5	$\overline{RD}$ Negated to Data Bus 3-stated	Max	20	20	20	20	ns
6	Output Hold Time	Min	5	5	5	5	ns
7	$\overline{RD}$ Pulse Width Low	Min	50	50	50	50	ns
8	$\overline{WR}$ Pulse Width Low	Min	50	50	50	50	ns
9	Write Data Setup Time	Min	10	10	10	15	ns
10	Write Data Hold Time	Min	10	10	10	15	ns
11	Clock Cycle Time	Min	12.5	15.2	20	28	ns
12	Clock Pulse Width High Time	Min	4	5	6	7	ns
13	Clock Pulse Width Low Time	Min	4	5	6	9	ns
14	Pixel Setup Time	Min	3	3	3	4	ns
15	Pixel Hold Time	Min	3	3	3	4	ns
16	Analog Output Delay	Max	30	30	30	30	ns
17	Analog Output RiseTime (Note 1)	Max	3	6	8	8	ns
18	Analog Output Settling Time (Note 1)	Max	13	15.3	20	28	ns
19	Read after write address register (read mode)	Min	5* t_{CYC}	5* t_{CYC}	5* t_{CYC}	5* t_{CYC}	ns
20	Successive read interval	Min	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	ns
21	Read after color read	Min	5* t_{CYC}	5* t_{CYC}	5* t_{CYC}	5* t_{CYC}	ns
22	Write after color read	Min	5* t_{CYC}	5* t_{CYC}	5* t_{CYC}	5* t_{CYC}	ns
23	Successive write interval	Min	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	ns
24	Read after color write	Min	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	ns
25	Write after color write	Min	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	ns
26	Read followed by write interval	Min	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	ns
27	Write followed by read interval	Min	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	4* t_{CYC}	ns
	Analog Output Skew	Max	2	2	2	2	ns
	Pipeline Delay	Typ	4	4	4	4	clocks
	V_{CC} Supply Current (Note 3)	Typ	150	150	150	150	mA
	V_{CC} Supply Current (Note 3)	Max	180	180	180	180	mA

PRELIMINARY

Notes:

1. Clock and data feedthrough are not included
2. Load = 37.5Ω + 30 pF with I_{REF} = –8.88 mA
3. Measured at maximum f_{CLK};
   I_{CC}(Max.): V_{CC} = 5.25 V, T_A = 0°C
   I_{CC}(Typ.): V_{CC} = 5.0 V, T_A = + 25°C

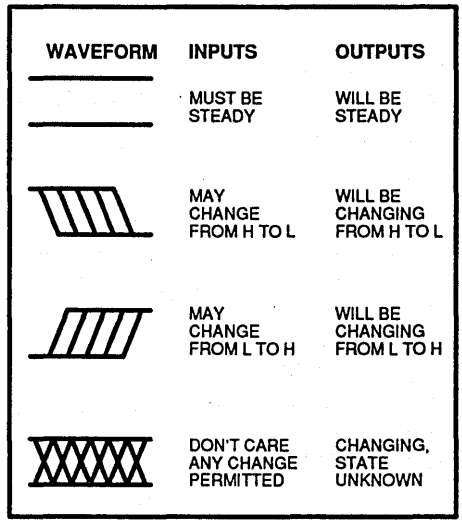
Test Conditions:

TTL Input Level: 0 to 3 V with t_R, t_F (10-90%) ≤ 3 ns

Analog Output Load ≤ 10 pF; $D_0 - D_7$ Output Load ≤ 50 pF

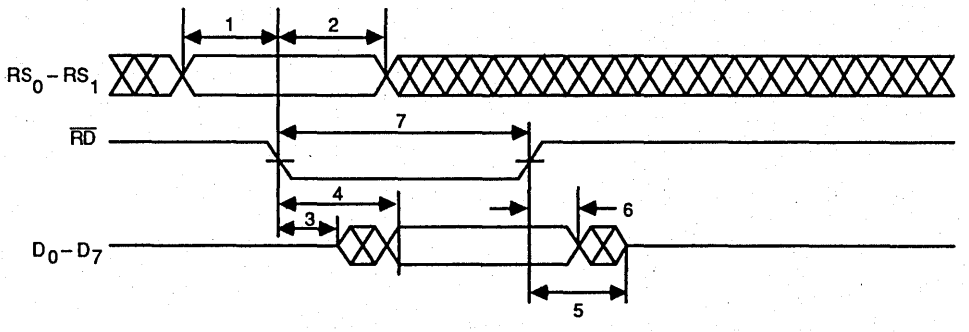
SWITCHING WAVEFORMS

Key To Switching Waveforms



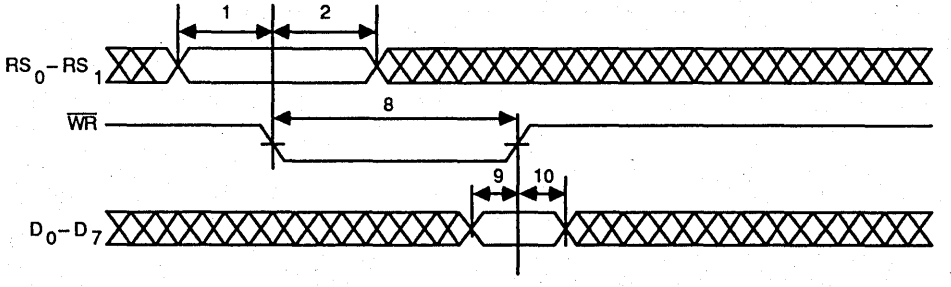
KS000010

SWITCHING WAVEFORMS (continued)



11912-005A

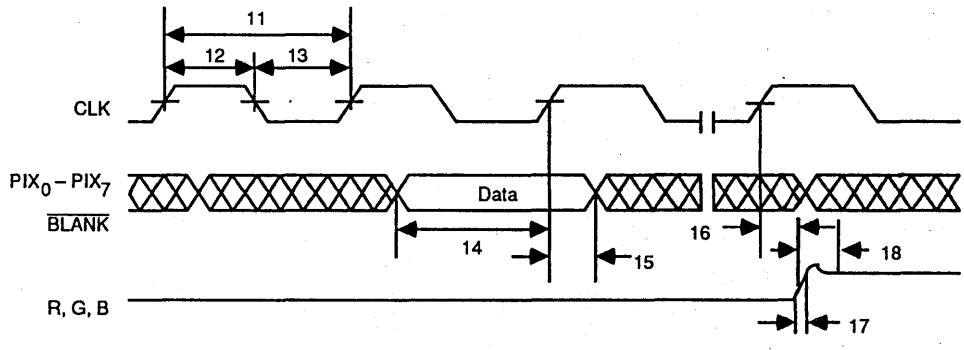
MPU Read Cycle



11912-006A

MPU Write Cycle

SWITCHING WAVEFORMS (continued)



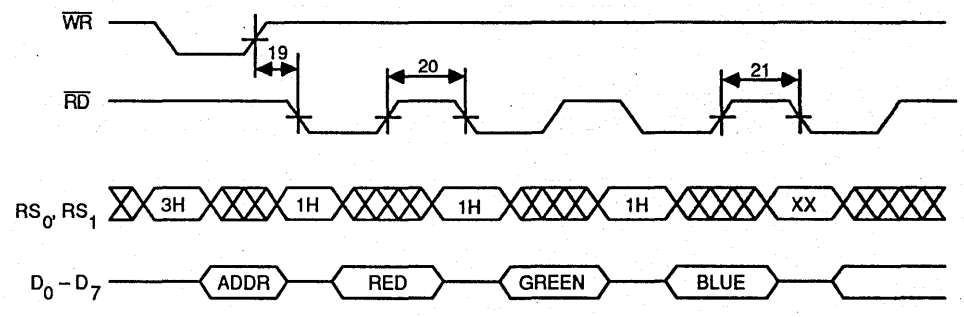
Note 1: Output delay measured from the 50% point of the rising edge of CLK to the 50 % point of the full scale transition.

Note 2: Settling time measured from the 50% point of the full scale transition to the output remaining within ± 1 LSB.

Note 3: Output rise/fall time measured between the 10% and 90% points of full scale transition.

11912-007A

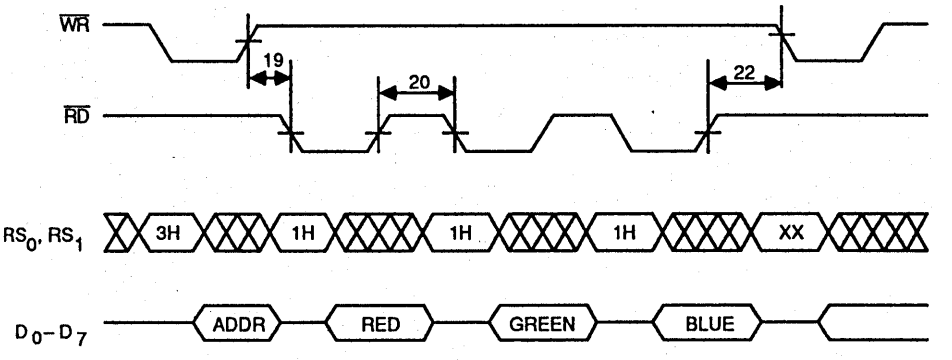
Video Input /Output



11912-008A

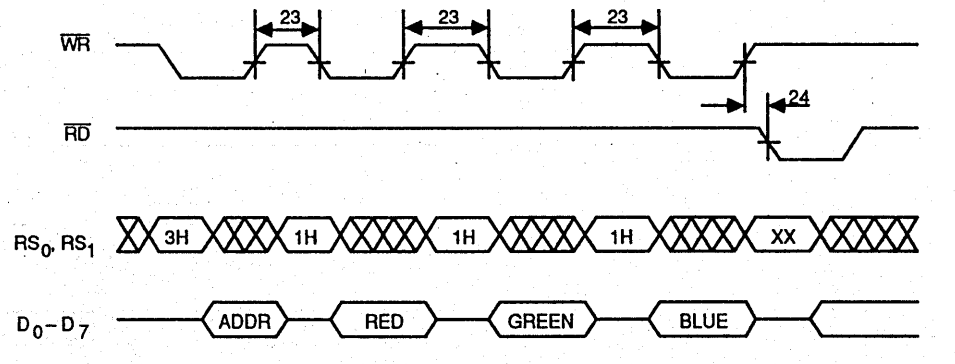
Color Value Read Followed by Any Read

SWITCHING WAVEFORMS (continued)



11912-009A

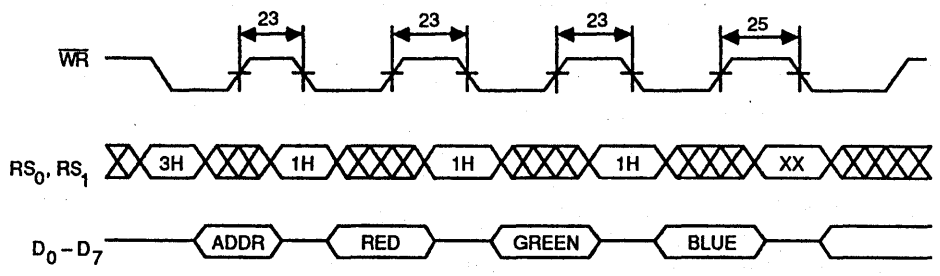
Color Value Read Followed by Any Write



11912-010A

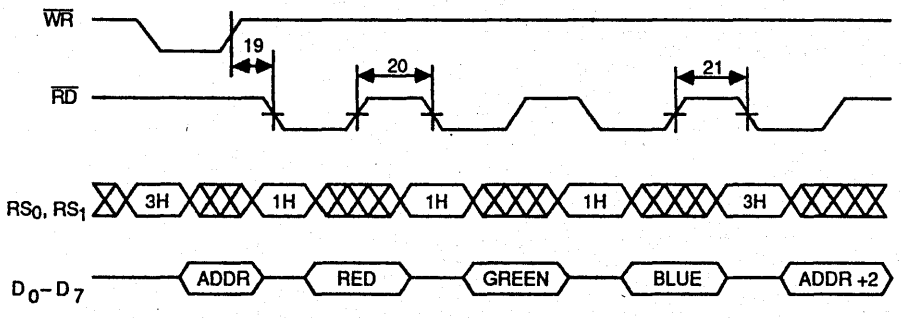
Color Value Write Followed by Any Read

SWITCHING WAVEFORMS (continued)



11912-011A

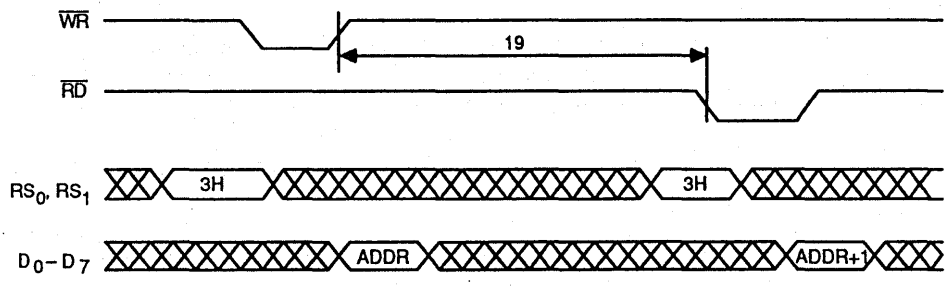
Color Value Write Followed by Any Write



11912-012A

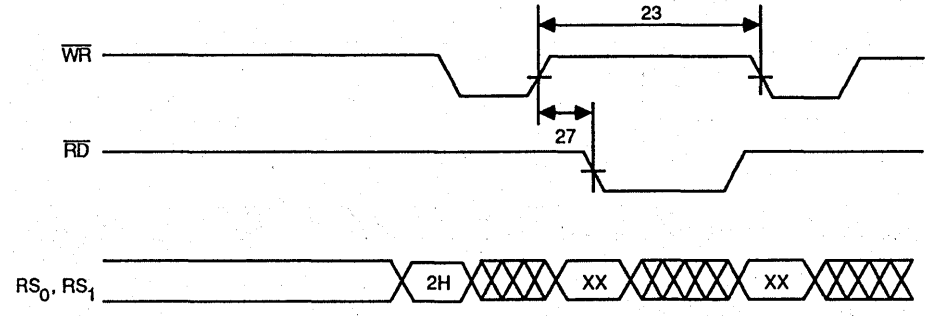
Read Color Value then Read the Address Register

SWITCHING WAVEFORMS (continued)



11912-013A

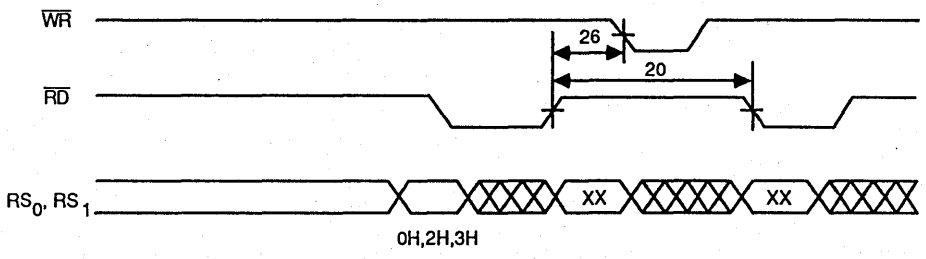
Write and then Read Back the Address Register



11912-014A

Write to Mask Register Followed by Read or Write

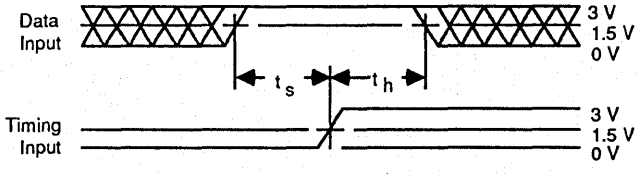
SWITCHING WAVEFORMS (continued)



11912-015A

Read from Mask or Address Register Followed by Read or Write

SWITCHING TEST WAVEFORM



Notes: 1. Diagram shown for HIGH data only. Output transition may be opposite sense.
2. Cross-hatched area is don't care condition.

11912-016A

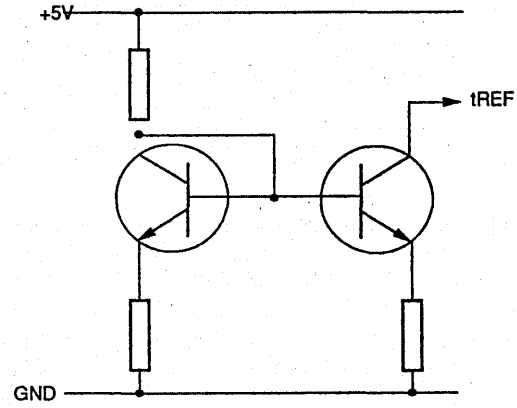
APPENDIX A: APPLICATION NOTE FOR THE Am81C176

The design of a system using the Am81C176 should be based on the guidelines used for designing high precision mixed analog and digital circuits. Some of these rules are outlined here. Users may, of course, choose to design circuits considerably different from that shown here.

Separate power planes should be used for the analog and digital power pins to reduce the noise on the analog output due to the switching at the digital inputs. A high frequency capacitor of around 0.1 µf should be placed close to the package between V_{CC} and GND. A large tantalum capacitor of about 22 µF should also be placed in parallel to the 0.1 µF capacitor. This same arrangement should be used on the DIP package as well. An inductor used in series with the power supply acts as a low-pass filter and improves the Am81C176 supply even further. Care should be taken to see that the analog plane does not cross the digital plane. The pixel data lines should be kept as far from the digital lines accessible by the MPU interface.

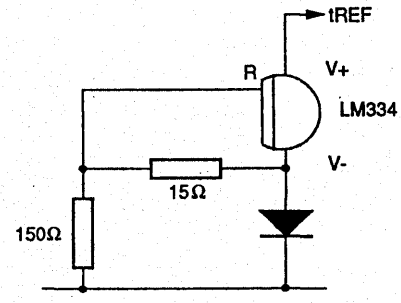
Use of a current mirror such as the one shown in Figure A1 is recommended to ensure a constant current source for the DACs. The current sources may need to be temperature compensated externally using a diode bias. It may be necessary to add coupling capacitors (47 µF in parallel with 0.1 µF) between I_{REF} and V_{CC} to absorb power supply variations not absorbed by the current source. The two capacitors help track the variations in power supply at the low and the high frequency end.

The connection between the DACs and the monitor acts as a transmission lie The two ends of the line need to be terminated to provide proper impedance matching and thereby avoid any reflections that might result otherwise. Also, analog output protection is provided by the diode as shown in Figure A2. The diode basically acts to prevent excessive negative or positive voltage swings which could result from electrostatic discharges and things of that nature.



11912-017A

Figure A1.



11912-018A

Figure A2.

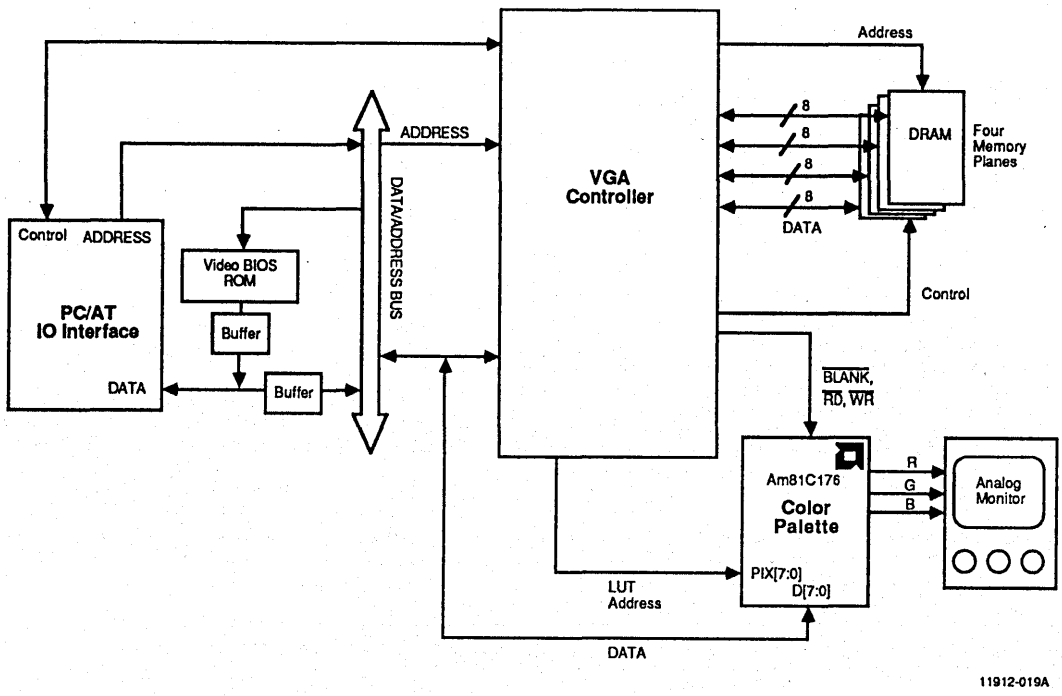


Figure A3. Application Example: VGA system using a VGA Controller and an Am81C176

Am81C451/458

CMOS Color Palette

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- Plug-in Replacement for Bt451 (Bt458)
- Available in 80-, 110-, 125-, 165-MHz versions
- Typical Power Dissipation: 1.0 W
- Available in PGA and PLCC packages
- Multiplexed TTL Pixel Ports
- Triple 4-Bit (8-Bit) Digital-to-Analog Converters (DACs)
- 256 x 12(24) Dual-Port Color Palette RAM
- 4 x 12(24) Dual-Port Overlay RAM
- RS-343A-Compatible RGB Outputs
- Read and Blink Masks for each bit-plane
- Standard MPU Interface
- Single +5-V Power Supply
- Full military range specifications

GENERAL DESCRIPTION

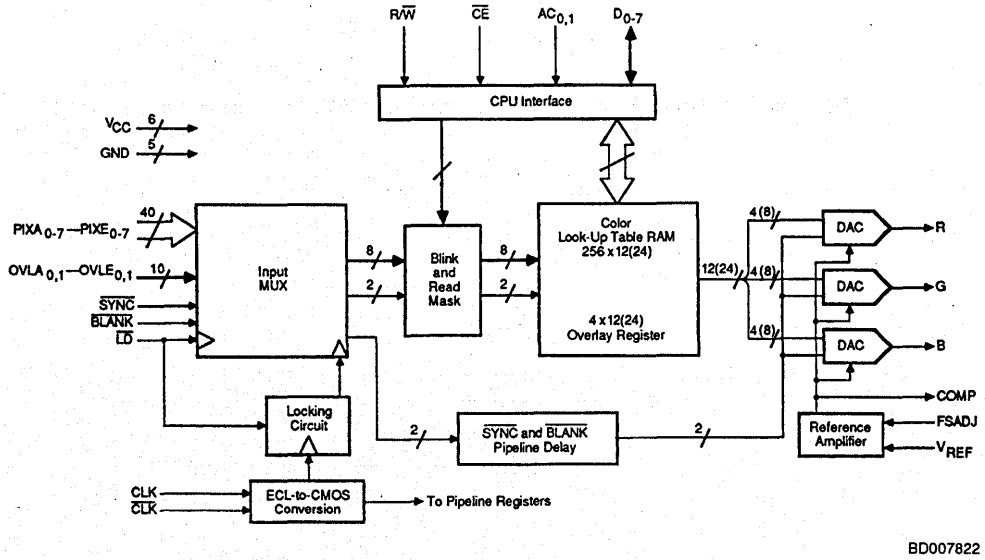
The Am81C451/458 CMOS Color Palette drives all three guns of a standard RS-343A color monitor. It is designed specifically for the high-resolution color graphics market for applications such as image processing, CAE/CAD/CAM, solid modeling, and animation. The Am81C451/458 operates at speeds sufficient to support monitor resolutions up to 1600 x 1280 pixels, and can simultaneously display 259 colors out of an available set of 4K colors for the Am81C451 and 16.8 million colors for the Am81C458.

The Am81C451/458 includes an input buffer, an input multiplexer, a 256 x 12(24) Look-Up Table, a 4 x 12(24) Overlay Table, and three 4-bit (8-bit) RS-343A-compatible DACs. It is available in versions with pixel rates as high as 165 MHz. It also contains programmable bit-plane Read and Blink Masks. Proprietary DAC decoding techniques minimize glitch energy and skew.

The Am81C451/458 minimizes the need for high-speed ECL signals on the PC board since there are only two inputs (CLK, $\overline{CLK}$) that need to operate at pixel rate. Multiple pixel ports and internal multiplexing enables TTL-compatible interfacing to the Display Memory while maintaining the high pixel data rates on-chip.

The Am81C451/458 is fabricated using AMD's state-of-the-art 1.2-micron CMOS process. The device is available in an 84-lead PGA package as well as a lower cost 84-pin PLCC package.

BLOCK DIAGRAM

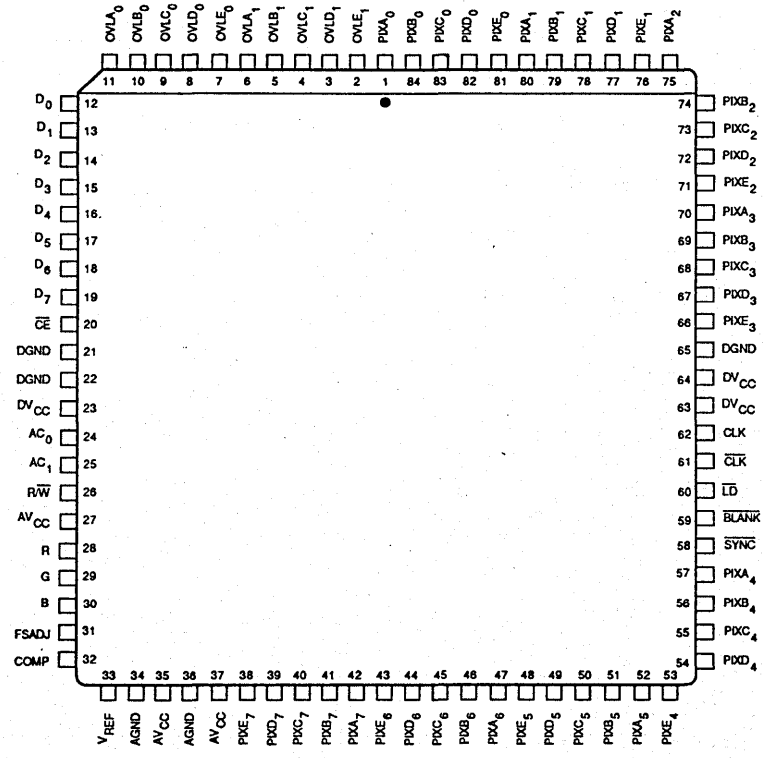


BD007822

Publication # 10451 | Rev. B | Amendment /0
Issue Date: October 1988

CONNECTION DIAGRAMS

PLCC
(Top View)



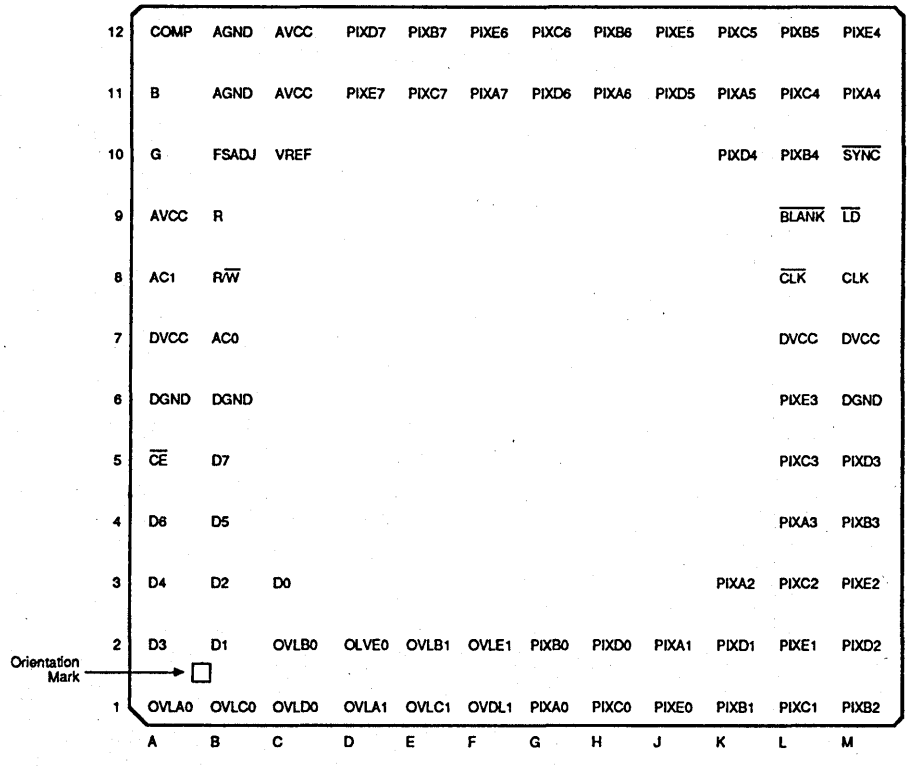
10451A-001A

CD011570

Note: Pin 1 is marked for orientation.

CONNECTION DIAGRAMS (Cont'd.)

PGA*
Top View
(Pins Facing Down)



10451A-002A
CD011580

Please see PGA Pin Designations for pinout sorted by both Pin Names and Pin Numbers.

PGA PIN DESIGNATIONS

(Sorted by Pin Name)

Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
B-7	AC_0	B-6	DGND	K-3	$PIXA_2$	E-11	$PIXC_7$
A-8	AC_1	M-6	DGND	L-4	$PIXA_3$	H-2	$PIXD_0$
B-12	AGND	A-6	DGND	M-11	$PIXA_4$	K-2	$PIXD_1$
B-11	AGND	L-7	DV_{CC}	K-11	$PIXA_5$	M-2	$PIXD_2$
C-12	AV_{CC}	A-7	DV_{CC}	H-11	$PIXA_6$	M-5	$PIXD_3$
C-11	AV_{CC}	M-7	DV_{CC}	F-11	$PIXA_7$	K-10	$PIXD_4$
A-9	AV_{CC}	B-10	FSADJ	G-2	$PIXB_0$	J-11	$PIXD_5$
A-11	B	A-10	G	K-1	$PIXB_1$	G-11	$PIXD_6$
L-9	$\overline{BLANK}$	M-9	$\overline{LD}$	M-1	$PIXB_2$	D-12	$PIXD_7$
A-5	$\overline{CE}$	A-1	$OVLA_0$	M-4	$PIXB_3$	J-1	$PIXE_0$
L-8	$\overline{CLK}$	D-1	$OVLA_1$	L-10	$PIXB_4$	L-2	$PIXE_1$
M-8	CLK	C-2	$OVLB_0$	L-12	$PIXB_5$	M-3	$PIXE_2$
A-12	COMP	E-2	$OVLB_1$	H-12	$PIXB_6$	L-6	$PIXE_3$
C-3	D_0	B-1	$OVLC_0$	E-1	$PIXB_7$	M-12	$PIXE_4$
B-2	D_1	E-1	$OVLC_1$	H-1	$PIXC_0$	J-12	$PIXE_5$
B-3	D_2	C-1	$OVLD_0$	L-1	$PIXC_1$	F-12	$PIXE_6$
A-2	D_3	F-1	$OVLD_1$	L-3	$PIXC_2$	D-11	$PIXE_7$
A-3	D_4	D-2	$OVLE_0$	L-5	$PIXC_3$	B-9	R
B-4	D_5	F-2	$OVLE_1$	L-11	$PIXC_4$	B-8	$R/\overline{W}$
A-4	D_6	G-1	$PIXA_0$	K-12	$PIXC_5$	M-10	$\overline{SYNC}$
B-5	D_7	J-2	$PIXA_1$	G-12	$PIXC_6$	C-10	V_{REF}

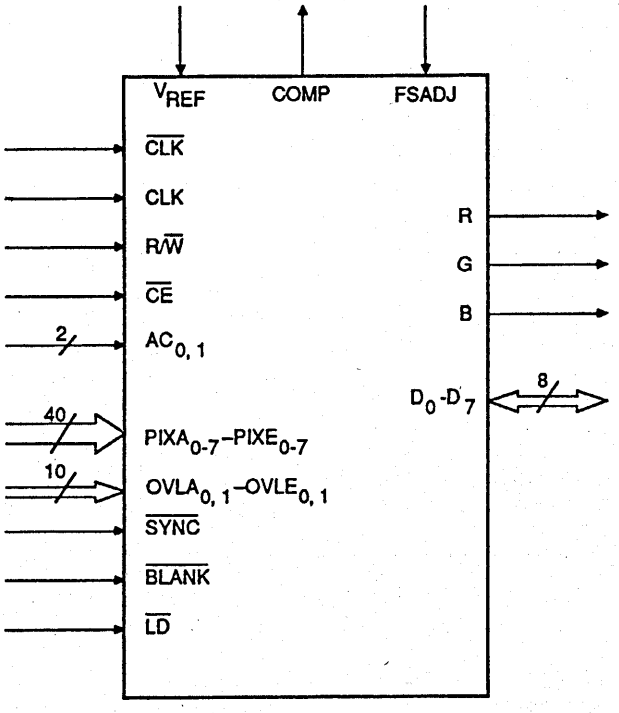
3

PGA PIN DESIGNATIONS (Cont'd.)

(Sorted by Pin Number)

Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
A-1	$OVLA_0$	B-10	FSADJ	G-1	$PIXA_0$	L-4	$PIXA_3$
A-2	D_3	B-11	AGND	G-2	$PIXB_0$	L-5	$PIXC_3$
A-3	D_4	B-12	AGND	G-11	$PIXD_6$	L-6	$PIXE_3$
A-4	D_6	C-1	$OVLD_0$	G-12	$PIXC_6$	L-7	DV_{CC}
A-5	$\overline{CE}$	C-2	$OVLB_0$	H-1	$PIXC_0$	L-8	$\overline{CLK}$
A-6	DGND	C-3	D_0	H-2	$PIXD_0$	L-9	$\overline{BLANK}$
A-7	DV_{CC}	C-10	V_{REF}	H-11	$PIXA_6$	L-10	$PIXB_4$
A-8	AC_1	C-11	AV_{CC}	H-12	$PIXB_6$	L-11	$PIXC_4$
A-9	AV_{CC}	C-12	AV_{CC}	J-1	$PIXE_0$	L-12	$PIXB_5$
A-10	G	D-1	$OVLA_1$	J-2	$PIXA_1$	M-1	$PIXB_2$
A-11	B	D-2	$OVLE_0$	J-11	$PIXD_5$	M-2	$PIXD_2$
A-12	COMP	D-11	$PIXE_7$	J-12	$PIXE_5$	M-3	$PIXE_2$
B-1	$OVLC_0$	D-12	$PIXD_7$	K-1	$PIXB_1$	M-4	$PIXB_3$
B-2	D_1	E-1	$OVLC_1$	K-2	$PIXD_1$	M-5	$PIXD_3$
B-3	D_2	E-2	$OVLB_1$	K-3	$PIXA_2$	M-6	DGND
B-4	D_5	E-11	$PIXC_7$	K-10	$PIXD_4$	M-7	DV_{CC}
B-5	D_7	E-12	$PIXB_7$	K-11	$PIXA_5$	M-8	CLK
B-6	DGND	F-1	$OVLD_1$	K-12	$PIXC_5$	M-9	$\overline{LD}$
B-7	AC_0	F-2	$OVLE_1$	L-1	$PIXC_1$	M-10	$\overline{SYNC}$
B-8	$R/\overline{W}$	F-11	$PIXA_7$	L-2	$PIXE_1$	M-11	$PIXA_4$
B-9	R	F-12	$PIXE_6$	L-3	$PIXC_2$	M-12	$PIXE_4$

LOGIC SYMBOL



10451A-003A

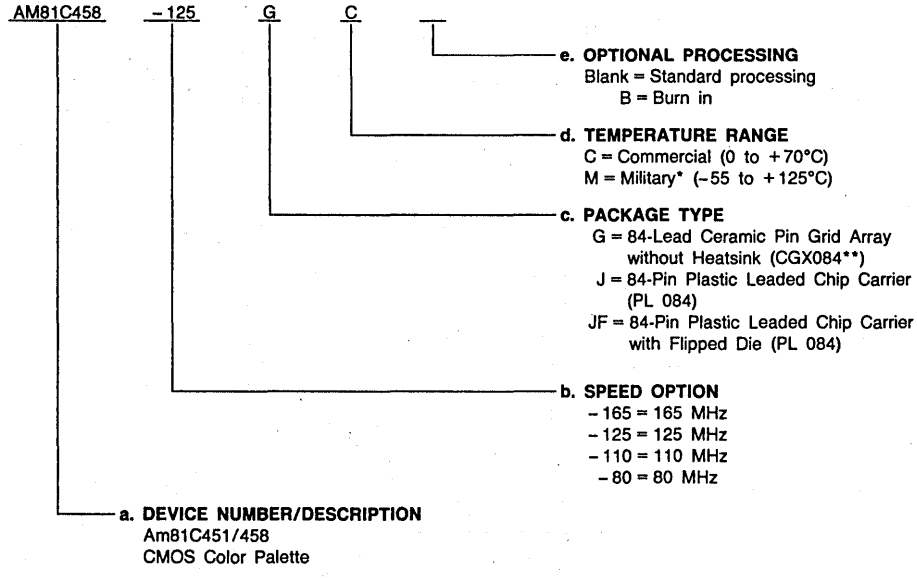
LS003271

3

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM81C451-165	GC, JC, GM, GMB
AM81C451-125	
AM81C451-110	
AM81C451-80	
AM81C458-165	
AM81C458-125	
AM81C458-110	
AM81C458-80	

*Military or Limited Military temperature range products are "NPL" (Non-Compliant Products List) or Non-MIL-STD-883C Compliant products only.

**Preliminary; Package in development.

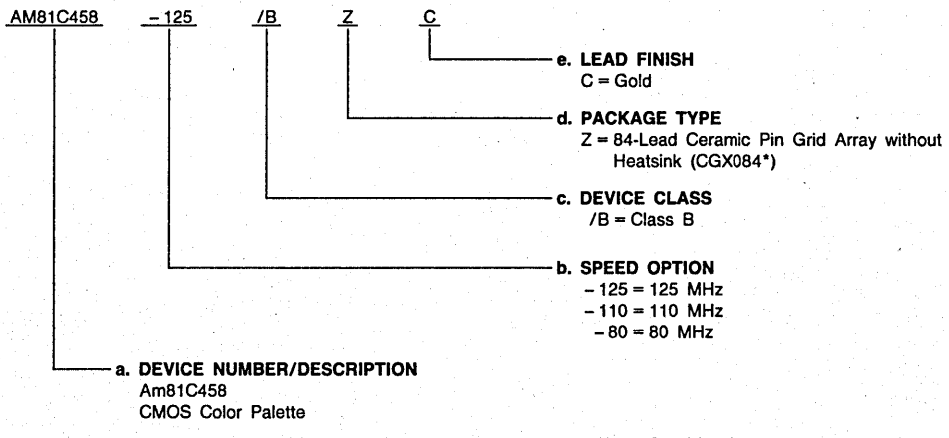
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defence applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM81C458-125	/BZC
AM81C458-110	
AM81C458-80	

*Preliminary; Package in development.

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups
1, 2, 3, 7, 8, 9, 10, 11.

3

PIN DESCRIPTION

Timing Section

$\overline{BLANK}$ $\overline{Blank}$ (TTL-Compatible Input)
When active, the $\overline{BLANK}$ input overrides the color pixel and overlay data to force the Red, Green, and Blue video outputs to their blank levels. This blank level is required during the monitor's vertical and horizontal retrace times. It is latched on the rising edge of $\overline{LD}$.

CLK, $\overline{CLK}$ Clock Source Pins (Pseudo-ECL-Compatible Inputs)
These differential clock inputs operate at the pixel clock rate of the system. They are driven by ECL logic configured for single (+5 V) supply operation.

$\overline{LD}$ $\overline{Load\ Clock}$ (TTL-Compatible Input)
Data present on the $PIXA_{0-7}$ through $PIXE_{0-7}$ and $OVLA_{0,1}$ through $OVLE_{0,1}$, $\overline{SYNC}$, and $\overline{BLANK}$ inputs are clocked into the part on the rising edge of $\overline{LD}$. The input rate for this pin may be either one-fourth or one-fifth of the clock frequency, depending on how the part is programmed. See Display Memory Interface section under Functional Description.

$\overline{SYNC}$ $\overline{Sync}$ (TTL-Compatible Input)
When active, the $\overline{SYNC}$ input switches off a current source on the Green video output. It is latched on the rising edge of $\overline{LD}$. Because $\overline{SYNC}$ does not override any other inputs or control pins, it should be asserted only during blanking intervals.

Data Path Section

$OVLA_{0,1}$ through $OVLE_{0,1}$ Color Overlay Data Address (TTL-Compatible Inputs)
These ten inputs are organized as five 2-bit addresses. Each 2-bit address selects which of the four Overlay Registers is used to provide color information. These ten inputs are latched into the input buffer on the rising edge of $\overline{LD}$. Either four or all five ports may be used for selecting Overlay Registers. Unused inputs should be grounded. These inputs are used with bit CMD6 of the Command Register as follows (assuming no Read and no Blink Masking):

$OVLX_1$	$OVLX_0$	CMD6 = 1	CMD6 = 0
0	0	Color Palette RAM	Overlay Color 0
0	1	Overlay Color 1	Overlay Color 1
1	0	Overlay Color 2	Overlay Color 2
1	1	Overlay Color 3	Overlay Color 3

$PIXA_{0-7}$ through $PIXE_{0-7}$ Color Pixel Data Addresses (TTL-Compatible Inputs)
These 40 inputs are organized as five 8-bit addresses. Each 8-bit address selects which of the 256 entries in the Color Look-Up Table is to be used to provide pixel color information. These 40 inputs are latched into the input buffer on the rising edge of $\overline{LD}$. Either four or all five ports may be used for selecting color information. Unused inputs should be grounded.

Analog Section

COMP Compensation Capacitor Connection (Analog Input)
A 0.1-μF ceramic capacitor is connected between this pin and AV_{CC}.

FSADJ Full-Scale Adjust (Analog Input)
The magnitude of the full-scale video signal is controlled by a resistor connected between FSADJ and AGND. The typical value for this resistor for RS-343A into 37.5 ohms is 523 ohms.

R Red Video Output (Analog Output)
This is the analog output of the Red DAC. This output is capable of driving an RS-343A-compatible doubly terminated 75-ohm cable.

G Green Video Output (Analog Output)
This is the analog output of the Green DAC. This output is capable of driving an RS-343A-compatible doubly terminated 75-ohm cable. The SYNC current source is connected to this output.

B Blue Video Output (Analog Output)
This is the analog output of the Blue DAC. This output is capable of driving an RS-343A-compatible doubly terminated 75-ohm cable.

V_{REF} Voltage Reference (Analog Input)
An external voltage reference circuit must be used to supply this input with a 1.235-V (typical) reference.

MPU Interface

$AC_{0,1}$ Address Control (TTL-Compatible Inputs)
AC_0 and AC_1 allow the MPU to address any location in the Color Look-Up Table or any of the internal control registers. They are latched on the falling edge of $\overline{CE}$. See Table 1.

$\overline{CE}$ $\overline{Chip\ Select}$ (TTL-Compatible Input)
This signal enables the MPU interface. Data on D_{0-7} is internally latched on the rising edge of $\overline{CE}$ during Write operations.

D_{0-7} Data and Address Bus (TTL-Compatible Input/Output)
These eight pins are used to load and read back the internal control registers and the Color Look-Up Table. D_0 is the least-significant bit.

R/$\overline{W}$ Read/$\overline{Write}$ (TTL-Compatible Input)
R/$\overline{W}$ is latched on the falling edge of $\overline{CE}$. A "logical one" indicates a Look-Up Table or Register Read-Back operation. A "logical zero" indicates a Write operation.

Power Supply

AGND, DGND Analog, Digital Ground

AV_{CC}, DV_{CC} Analog, Digital +5-Volt Supply

FUNCTIONAL DESCRIPTION

The Am81C451/458 CMOS Color Palette integrates all the major functions required in the back-end of a video system, and supports pixel rates sufficient for most medium- to high-resolution monitors.

Four or five pixels are input in parallel from Display Memory and are serialized internally. A programmable Look-Up Table maps the serial pixel stream (address) into a physical color (data), and finally, three DACs convert the digital outputs of the look-up table into RS-343A-compatible RGB analog format.

Microprocessor Interface

A standard 8-bit MPU interface allows easy communication between the Am81C451/458 and most common MPUs. The $\overline{CE}$ and R/$\overline{W}$ inputs control MPU access timing, as shown in Figure 1. The AC_1 and AC_0 inputs select the access type as detailed in Table 1.

TABLE 1. AC_1, AC_0 DECODING

AC_1	AC_0	Access Type
0	0	Address Register
0	1	Look-Up Table
1	0	Control Registers
1	1	Overlay Registers

Fast access to the Look-Up Table and to the Overlay Registers is achieved by means of two internal counters: an 8-bit Address Register (AR0 – AR7), which generates addresses for the Color Memory locations and the Control Registers, and a Modulo 3 Counter (ARa,ARb) that controls which byte of the 24-bit Color Memory word is accessed. Tables 2 and 3 illustrate the operation of these two counters.

TABLE 2. ADDRESS REGISTER OPERATION

AR7-AR0	AC_1	AC_0	Location/Register Addressed by MPU
$00-$FF	0	1	Color Look-up Table Location #00-#FF
$00	1	1	Overlay Register 0
$01	1	1	Overlay Register 1
$02	1	1	Overlay Register 2
$03	1	1	Overlay Register 3
$04	1	0	Read Mask Register
$05	1	0	Blink Mask Register
$06	1	0	Command Register
$07	1	0	Test Register

TABLE 3. MODULO 3 COUNTER OPERATION

ARb	ARa	Color Byte Being Accessed
0	0	Red
0	1	Green
1	0	Blue

The Address Register, directly accessible by the MPU, auto-increments at the end of each third (Blue) access having $AC_0 = 1$. This feature avoids the rewriting of the Address Register with consecutive values, saving MPU time and Bus bandwidth for transfers to or from consecutive Color Memory locations.

The Modulo 3 Counter, not accessible by the MPU, increments at the end of each MPU access with $AC_0 = 1$ (color operations), and is reset to 0 at the end of each MPU access with $AC_0 = 0$ (Control Register operations).

Table 4 illustrates the Read/Write access to the Am81C451/458 palette.

TABLE 4. READ/WRITE ACCESS TO THE Am81C458

R/$\overline{W}$	AC_1	AC_0	ARb	ARa	Function
0	0	0	x	x	Write Address Register; AR7-AR0 ← D_{7-0}; ARb, ARa ← 0.
0	0	1	0	0	Write Red Color; RREG ← D_{7-0}; Incr. ARb, ARa.
0	0	1	0	1	Write Green Color; GREG ← D_{7-0}; Incr. ARb, ARa.
0	0	1	1	0	Write Blue Color; BREG ← D_{7-0}; ARb, ARa ← 0
					Write Color Look-Up Table; R7-R0 ← RREG; G7-G0 ← GREG; B7-B0 ← BREG; Incr. AR7-AR0.
0	1	0	x	x	Write Control Register; Reg (AR7-AR0) ← D_{7-0}; ARb, ARa ← 0.
0	1	1	0	0	Write Red Color; RREG ← D_{7-0}; Incr. ARb, ARa.
0	1	1	0	1	Write Green Color; GREG ← D_{7-0}; Incr. ARb, ARa.
0	1	1	1	0	Write Blue Color; BREG ← D_{7-0}; ARb, ARa ← 0;
					Write Overlay Register; R7-R0 ← RREG; G7-G0 ← GREG; B7-B0 ← BREG; Incr. AR7-AR0.
1	0	0	x	x	Read Address Register; D_{7-0} ← AR7-AR0; ARb, ARa ← 0.
1	0	1	0	0	Read Color LUT Red; D_{7-0} ← R7-R0; Incr. ARb, ARa.
1	0	1	0	1	Read Color LUT Green; D_{7-0} ← G7-G0; Incr. ARb, ARa.
1	0	1	1	0	Read Color LUT Blue; D_{7-0} ← B7-B0; ARb, ARa ← 0; Incr. AR7-AR0.
1	1	0	x	x	Read Control Register; D_{7-0} ← Reg (AR7-AR0); ARb, Ara ← 0.
1	1	1	0	0	Read Overlay Red; D_{7-0} ← R7-R0; Incr. ARb, ARa.
1	1	1	0	1	Read Overlay Green; D_{7-0} ← G7-G0; Incr. ARb, ARa.
1	1	1	1	0	Read Overlay Blue; D_{7-0} ← B7-B0; ARb, ARa ← 0; Incr. AR7-AR0.

Key:
← = "gets the value of"
D_{7-0} = MPU Data Bus
R7-R0 = Color Memory Red Byte
G7-G0 = Color Memory Green Byte
B7-B0 = Color Memory Blue Byte
RREG = Red Byte Register
GREG = Green Byte Register
BREG = Blue Byte Register
Reg (AR7-AR0) = Register pointed to by Address Register

Note that for the Am81C451 only the most significant data lines (D_7–D_4) are used while accessing the color lookup table or overlay registers. During a write cycle bits D_3–D_0 are ignored and during a read cycle bits D_3–D_0 are forced to logical zero.

If the pixel or overlay inputs address same entry in the color look-up table that is being written to by the MPU during blue write, the possibility exists that one or more pixels may be corrupted. Only one pixel may be corrupted if the MPU data is valid during the entire $\overline{CE}$ active time.

Display Memory Interface

The Am81C451/458 allows pixel data to be transferred from Display Memory at TTL-type data rates while presenting RGB information to the CRT at much higher rates through the use of internal latches and multiplexers.

Forty pixel data inputs ($PIXA_{0-7}$ through $PIXE_{0-7}$), ten overlay data inputs ($OVLA_{0,1}$ through $OVLE_{0,1}$), and two video inputs ($\overline{SYNC}$ and $\overline{BLANK}$), are loaded on the rising edge of $\overline{LD}$. An input MUX performs a 4:1 or a 5:1 serialization based on the FORMAT bit (CMD7) of the Command Register. During each clock cycle the Am81C451/458 outputs video information based first on the $PIXA_{0-7}$ inputs, then the $PIXB_{0-7}$ inputs, and so on until the $PIXD_{0-7}$ inputs (CMD7 = 0) or $PIXE_{0-7}$ inputs (CMD7 = 1), at which time the cycle repeats.

No phase relationship is imposed on the $\overline{LD}$ and CLK inputs; the only requirement is that the $\overline{LD}$ frequency be one-fourth or one-fifth of pixel clock (CLK, $\overline{CLK}$ depending on the CMD7 bit of the Command Register). This is obtained by virtue of an on-chip Locking Circuit that guarantees stable inputs to the Resync Register during positive transitions of the internal load signal (see Figure 1).

Note that the pixel data, overlay data, and $\overline{SYNC}$ and $\overline{BLANK}$ are loaded with the same $\overline{LD}$ clock to maintain synchronization with video data inside the Am81C451/458.

Color Selection

During each clock cycle, 10 bits of data are transferred from the input MUX and processed by the Read Mask, Blink Mask, and Command Register (see Figure 2). The processed data then selects an entry in the color palette RAM or an Overlay Register to provide color information.

The Read Mask is used to selectively enable or disable bit-planes from being presented to the color palette RAM; the Blink Mask is used to selectively enable or disable blinking on a bit-plane. CMD4 and CMD5 in the Command Register determine the blink-rate duty cycle. The counter that generates the internal blink clock is incremented during vertical retrace intervals. Such intervals are detected when the $\overline{BLANK}$ input is LOW for a period of at least 256 $\overline{LD}$ cycles.

Overlay Color Selection

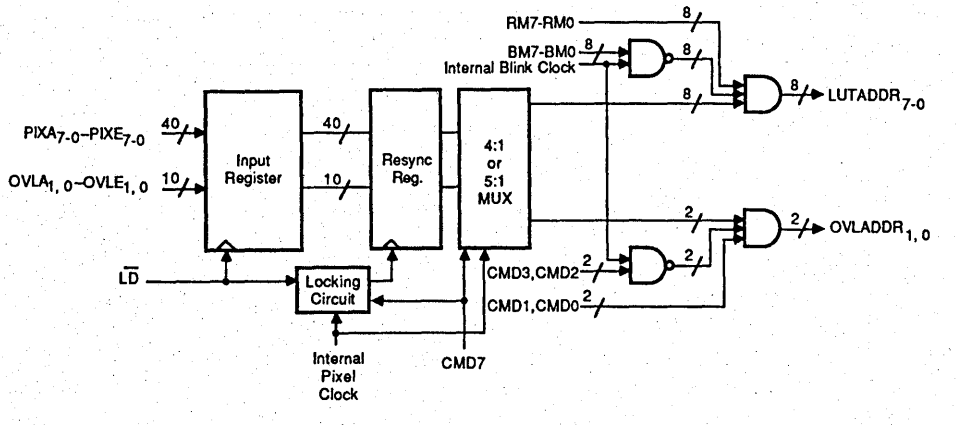
Four different overlay colors are available through four Overlay Registers. CMD1 and CMD0 in the Command Register act as Read Masks, and CMD2 and CMD3 in the Command Register act as Blink Masks on the overlay inputs (see Figure 1).

When $OVLADDR_1 = 0$ and $OVLADDR_0 = 0$ (see Figure 1), CMD6 of the Command Register selects between Overlay Register 0 and Look-Up Table output (determined by $LUTADDR_{7-0}$). See Table 5.

TABLE 5. OVERLAY COLOR SELECTION

$OVLADDR_1$	$OVLADDR_0$	CMD6 = 0	CMD6 = 1
0	0	Overlay Register 0	LUT
0	1	Overlay Register 1	Overlay Register 1
1	0	Overlay Register 2	Overlay Register 2
1	1	Overlay Register 3	Overlay Register 3

Key: LUT = Look-Up Table content addressed by $LUTADDR_{7-0}$
CMD6 = Bit 6 of Command Register (RAM Enable)



Note: CMDx bits are programmed in the Command Register
BMx bits are programmed in the Blink Mask Register
RMx bits are programmed in the Read Mask Register

10451B-006A
BD008070

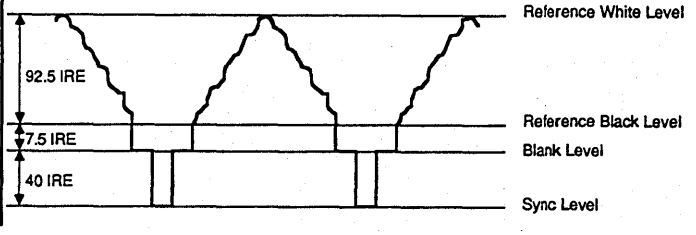
Figure 1. Input MUX and Masking Stage

Video Generation

During each clock cycle, 12(24) bits of information from either the Look-Up Table or an Overlay Register are presented to the three 4-bit (8-bit) DACs. These DACs convert the Color Memory digital output into RGB RS-343A analog format.

The $\overline{SYNC}$ and $\overline{BLANK}$ inputs are routed to the DACs after a delay equal to the pipeline delay incurred by the video stream to produce the relative Blank and Sync levels. $\overline{BLANK}$ is routed to all three DACs while $\overline{SYNC}$ is routed only to the Green DAC. See Figure 2 for DAC current and voltage levels.

RED		GREEN	
mA	mV	mA	mV
19.04	714	26.67	1000
1.44	54	9.07	340
0.00	0	7.63	286
-	-	0.00	0



10451C-007A

DF006680

Note: 75-Ω doubly terminated load. Sync on Green DAC only. When Sync is used, the Green DAC current output is 7.63 mA higher than the corresponding Red or Blue current ouputs. When Sync is not used, Green, Red, and Blue outputs are identical.

Figure 2. DAC Current and Voltage Levels

User-Accessible Registers

In addition to the address register, there are four user-accessible registers: Command, Read Mask, Blink Mask, and Test. These registers should be initialized after power-up.

Command Register (CMD7 – CMD0)

This is an 8-bit register located at address #6, and is described below.

TABLE 6. COMMAND REGISTER DEFINITION

Position	Name	Description
CMD7	**FORMAT** 0 = 4:1 Multiplexing 1 = 5:1 Multiplexing	CMD7 specifies whether 4:1 or 5:1 multiplexing should take place on the pixel and overlay data. If CMD7 is set to logical '0' (4:1 multiplexing), $\overline{LD}$ should be one-fourth the frequency of CLK; the $PIXE_{0-7}$ and $OVLE_{0,1}$ inputs are ignored and should be connected to GND. If CMD7 is set to logical '1' (5:1 multiplexing), $\overline{LD}$ should be one-fifth the frequency of CLK.
CMD6	**RAM Enable** 0 = Use Overlay Register 0 1 = Use LUT	When the processed overlay inputs $OVLADDR_1$ and $OVLADDR_0$ equal (0, 0) this bit selects between the Color Look-Up Table output and Overlay Register 0.
CMD5, CMD4	**Blink Rate Select** 00 = 25/75 (16/48) 01 = 50/50 (16/16) 10 = 50/50 (32/32) 11 = 50/50 (64/64)	These bits specify the blink rate and duty cycle of the internal blink clock. Numbers in parentheses specify the rate and duty cycle of the internal blink clock (logic 1/logic 0) expressed in vertical retrace intervals. See Figure 1.
CMD3	**OVL_1 Blink Mask** 0 = Disable Blinking 1 = Enable Blinking	When this bit is set to a logical 1, the $OVLA_1$ Through $OVLE_1$ inputs are allowed to toggle at the selected blinking blink rate between the input value and logical 0 before being applied to the LUT/Overlay decode logic. When this bit is set to logical 0 it does not affect the $OVLA_1$ through $OVLE_1$ inputs.
CMD2	**OVL_0 Blink Mask** 0 = Disable Blinking 1 = Enable Blinking	When this bit is set to a logical 1, the $OVLA_0$ through $OVLE_0$ inputs are allowed to toggle at the selected blink rate between the input value and logical 0 before being applied to the LUT/Overlay decode logic. When this bit is set to logical 0 it does not affect the $OVLA_0$ through $OVLE_0$ inputs.
CMD1	**OVL_1 Read Mask** 0 = Disable Mask 1 = Enable Mask	When this bit is set to logical 0, the $OVLA_1$ through $OVLE_1$ inputs are forced to logical 0 before being applied to the LUT/Overlay decode logic. When this bit is set to logical 1 it does not affect the $OVLA_1$ through $OVLE_1$ inputs.
CMD0	**OVL_0 Read Mask** 0 = Disable Mask 1 = Enable Mask	When this bit is set to logical 0, the $OVLA_0$ through $OVLE_0$ inputs are forced to logical 0 before being applied to the LUT/Overlay decode logic. When this bit is set to logical 1 it does not affect the $OVLA_0$ through $OVLE_0$ inputs.

Read Mask Register (RM7 – RM0)

This is an 8-bit register located at address #4. It selectively enables (logical 1) or disables (logical 0) pixel bit-planes from addressing the Color Look-Up Table. Bit RM0 will mask inputs $PIXA_0$ through $PIXE_0$; bit RM1 will mask inputs $PIXA_1$ through $PIXE_1$; and so on.

Overlay plane masking is controlled by bits CMD1 and CMD0 of the Command Register. See Figure 1.

Blink Mask Register (BM7 – BM0)

This is an 8-bit register located at address #5. It selectively enables (logical 1) or disables (logical 0) pixel bit-planes from blinking. Bit BM0 will mask inputs $PIXA_0$ through $PIXE_0$; bit BM1 will mask inputs $PIXA_1$ through $PIXE_1$; and so on. When enabled, that particular bit-plane will toggle between its original value and logical 0 at the selected rate and duty cycle.

Overlay plane blinking is controlled by bits CMD3 and CMD2 of the Command Register. See Figure 1.

Test Register (T7 – T0)

This is an 8-bit register located at address #7. It is used to read back data presented to the DACs from the Color Memory (either pixel or overlay data). The four most-significant bits, T7 – T4, contain color information while the four least-significant bits, T3 – T0, contain control information, as shown in Table 7. T7 – T4 are defined only when exactly one of the T2 – T0 bits is a "logical one". Note that for the Am81C451, bit T3 is forced to logical zero.

When unused, this register should be initialized to Hex 00.

TABLE 7. TEST REGISTER DEFINITION

Test Register Bits	Function
T7-T4	Color Nibble
T3	1 = Select LOW Nibble 0 = Select HIGH Nibble
T2	1 = Enable Blue Data 0 = Disable Blue Data
T1	1 = Enable Green Data 0 = Disable Green Data
T0	1 = Enable Red Data 0 = Disable Red Data

As an example, in order to read the least-significant 4 bits of data presented to the Green DAC, the user must first write Hex 0A to the Test Register to enable the LOW nibble and green data. Subsequently, the user reads the Test Register, keeping the pixel inputs constant; bits D7 – D4 of the MPU interface Bus will contain the desired color data while bits D3 – D0 will contain Hex A.

When reading the Test Register, the data presented to the DAC inputs must be held stable during this period either by slowing the pixel clock or by holding the pixel and overlay inputs constant.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature
  Under Bias ..... −55 to +125°C
  Junction Temperature ..... +175°C
Supply Voltage to Ground Potential
  Continuous ..... −0.5 to +7.0 V
DC Voltage Applied to Outputs for
  HIGH Output State ..... −0.5 V to V_{CC} Max.
DC Input Voltage GND ..... −0.3 to V_{CC} + 0.3 V
DC Input Current ..... −10 to +10 mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
  Ambient Temperature (T_A) ..... 0 to +70°C
  Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V
Military (M) Devices
  Case Temperature (T_C) ..... −55 to +125°C
  Supply Voltage (V_{CC}) ..... +4.50 to +5.50 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified (for APL products, Group A Subgroups 1, 2, 3, 7, & 8 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions		Min.	Typ.	Max.	Unit
DIGITAL CLOCK INPUTS (CLK, $\overline{CLK}$)							
V_{CKIH}	Input HIGH Voltage			V_{CC} −1.0		V_{CC} +0.5	V
V_{CKIL}	Input LOW Voltage			GND −0.5		V_{CC} −1.6	V
I_{CKIH}	Input HIGH Current					1	µA
I_{CKIL}	Input LOW Current					−1	µA
C_{CKIN}	Input Capacitance	f = 1 MHz, V_{IN} = 4.0 V				10	pF
DIGITAL INPUTS (Except CLK, $\overline{CLK}$)							
V_{IH}	Input HIGH Voltage			2.0		V_{CC} +0.5	V
V_{IL}	Input LOW Voltage			GND −0.5		0.8	V
I_{IH}	Input HIGH Current	V_{IN} = 2.4 V				1	µA
I_{IL}	Input LOW Current	V_{IN} = 0.4 V				−1	µA
C_{IN}	Input Capacitance	f = 1 MHz, V_{IN} = 2.4 V				10	pF
DIGITAL OUTPUTS (D_{0-7})							
V_{OH}	Output HIGH Voltage	I_{OH} = 800 µA		2.4			V
V_{OL}	Output LOW Voltage	I_{OL} = 6.4 mA				0.4	V
I_{OZ}	Three-State Current					10	µA
C_{OUT}	Output Capacitance				10		pF
ANALOG OUTPUTS (R_{FS} = 523 Ω, V_{REF} = 1.235 V)							
	Resolution (Each DAC)			4 (8)		4 (8)	Bits
LIN_i LIN_d	Accuracy (Each DAC): Integral Linearity Differential Linearity LSB Output Current		COM'L		69.1	±1/8 (±1) ±1/16 (±1)	LSB
			MIL			+1/8 (±2) +1/16 (±1)	
	LSB Output Current				1175 (69.1)		µA
	Gray Scale Error	% Gray Scale				±5	%
	Monotonicity					Guaranteed	
	Coding	Binary					
	Output Current: White Level Relative to Blank Level White Level Relative to Black Level Black Level Relative to Blank Level Blank Level on R, B Blank Level on G Sync Level on G	R_{FS} = 523 Ω, V_{REF} = 1.235 V		17.69 16.74 0.95 0 6.29 0	19.05 17.62 1.44 5 7.62 5	20.40 18.50 1.90 50 8.96 50	mA mA mA µA mA µA
	DAC-to-DAC Matching				2	5	%
V_{OC}	Output Compliance			−1.0		+1.2	V
R_{OUT}	Output Impedance				50		kΩ
C_{OUT}	Output Capacitance	f = 1 MHz, I_{OUT} = 0 mA			13	20	pF
I_{REF}	Volt Reference Input Current				10		µA
PSRR	Power Supply Rejection Ratio	C_{COMP} = 0.1 µF f = 1 KHz			0.5		%/%Δ V_{CC}

PRELIMINARY

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified (for APL Products, Group A, Subgroups 9, 10, 11 unless otherwise noted)

No.	Parameter Symbol	Parameter Description	165 MHz			125 MHz			110 MHz			80 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
DIGITAL CPU INTERFACE TIMING															
1	t_S	R/$\overline{W}$, AC_0, AC_1 Setup Time	0			0			0			0			ns
2	t_H	R/$\overline{W}$, AC_0, AC_1 Hold Time	15			15			15			15			ns
3	t_{CYC}	CPU Cycle Time	100			100			110			125			ns
4	t_W	$\overline{CE}$ LOW Active Time	70			75			85			100			ns
5	t_W	$\overline{CE}$ HIGH Time	25			25			25			25			ns
6	t_{PD}	$\overline{CE}$ Edge to Data Bus Driven (C_L = 40 pF)	10			10			10			10			ns
7	t_{PD}	$\overline{CE}$ Edge to Data Valid (C_L = 40 pF)			70			75			85			100	ns
8	t_{PD}	$\overline{CE}$ Edge to Data Three-Stated (C_L = 40 pF)			15			15			15			15	ns
9	t_S	Write Data Setup Time	35			35			40			50			ns
10	t_H	Write Data Hold Time	0			0			0			0			ns
DIGITAL VIDEO PATH TIMING															
11	f_{CK}	Video Clock Frequency			165			125			110			80	MHz
12	f_{LD}	$\overline{LD}$ Rate			41.25			31.25			27.5			20	MHz
13	t_{CYC}	$\overline{LD}$ Cycle Time	24			32			36.36			50			ns
14	t_W	$\overline{LD}$ Pulse Width LOW	9			13			15			20			ns
15	t_W	$\overline{LD}$ Pulse Width HIGH	9			13			15			20			ns
16	t_S	PIX_x, OVL_x, $\overline{SYNC}$ and $\overline{BLANK}$ Setup Time	2			3			3			4			ns
17	t_H	PIX_x, OVL_x, $\overline{SYNC}$ and $\overline{BLANK}$ Hold Time	2			2			2			2			ns
18	t_{CYC}	Clock Cycle Time	6			8			9.09			12.5			ns
19	t_W	Clock Pulse Width LOW	2.4			3.2			4.0			5.0			ns
20	t_W	Clock Pulse Width HIGH	2.4			3.2			4.0			5.0			ns
21	t_R, t_F	Clock Rise/Fall Time (20%-80%)			1.6			1.6			1.6			1.6	ns
22		Pipeline Delay	6		10	6		10	6		10	6		10	Clks

Notes: See notes following end of table continued on next page.

PRELIMINARY

SWITCHING CHARACTERISTICS over operating ranges (Cont'd.)

No.	Parameter Symbol	Parameter Description		165 MHz			125 MHz			110 MHz			80 MHz			Unit
				Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
ANALOG VIDEO DAC TIMING																
23		Analog Output Delay			20			20			20			20		ns
24	t_R, t_F	Analog Output Rise and Fall Time (Note 1)			2			2			2			2		ns
25	t_S	Analog Output Settling Time (Notes 1, 2)				6			8			8			12	ns
26		Analog Output Skew			0	2		0	2		0	2		0	2	ns
27		Glitch Impulse Energy (Note 2)			50			50			50			50		PV-sec
28		Clock and Data Feedthrough (Note 3)			TBD			TBD			TBD			TBD		dB
DYNAMIC POWER DISSIPATION																
		I_{CC} (Supply Current)	COM'L		270	370		250	340		240	330		230	295	mA
			MIL	–	–	–		250	425		240	410		230	370	

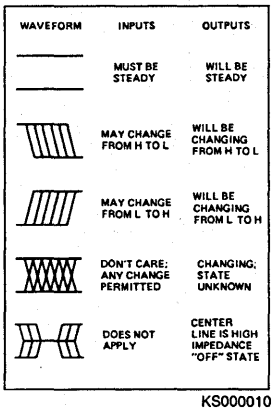
Notes: 1. Clock and data feedthrough is not included.
2. Includes clock and data feedthrough, –3–dB bandwidth = 2 x clock frequency.
3. Measurement performed on TTL digital inputs with 74HC logic level and with 1-kohm resistor to GND.

Test Conditions: TTL Input Level: 0 to 3 V with t_R, t_F(10-90%) $\leq$ 3 ns;
ECL Input Level: (V_{CC} - 0.8 V) to (V_{CC} - 1.8 V) with t_F, t_R (20-80%) $\leq$ 2 ns;
R_{SET} = 523 ohms, V_{REF} = 1.235 V;
Analog Output Load $\leq$ 10 pF; D_{0-7} Output Load $\leq$ 40 pF.

PRELIMINARY

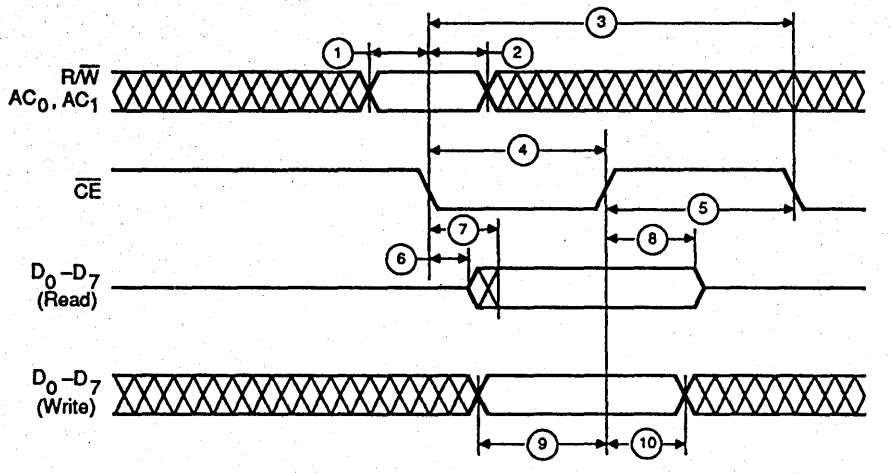
SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS



KS000010

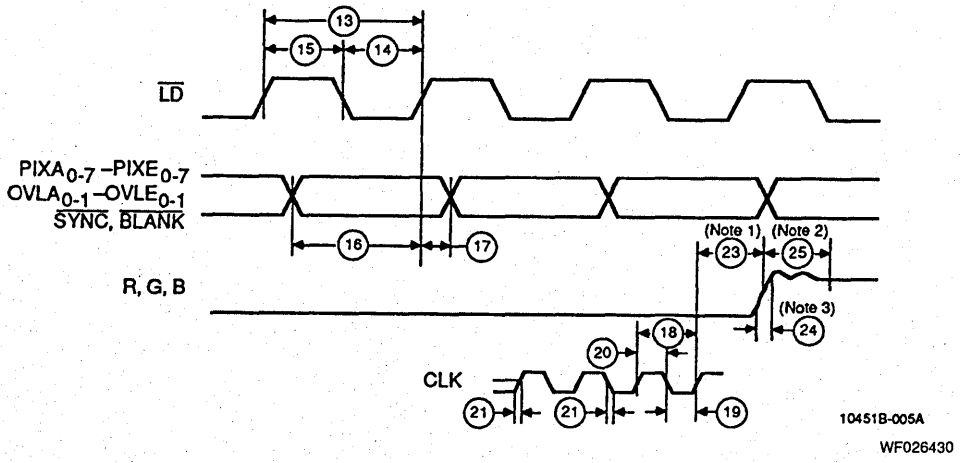
SWITCHING WAVEFORMS (Cont'd.)



10451D-004A

WF026630

CPU Read/Write Timing



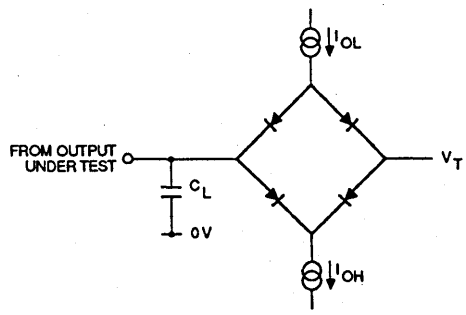
10451B-005A

WF026430

Notes: 1. Output delay time measured from 50% point of the rising clock edge to 50% point of full-scale level.
2. Output settling time measured from 50% point of full-scale level to output settling within ±1 LSB.
3. Output rise/fall time measured between 10% and 90% points of full-scale level.

Video Input/Output Timing

SWITCHING TEST CIRCUIT



AF004810

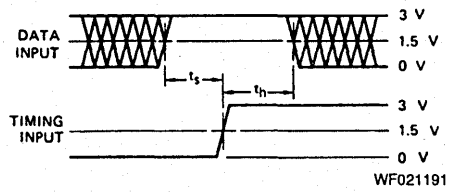
Notes: 1. C_L = 50 pF (includes scope probe, wiring, and stray capacitance without device in test fixture).
2. V_T = 1.5 V.

Notes on Testing

Incoming test procedures on this device should be carefully planned, taking into account the complexity and power levels of the part. The following notes may be useful.

1. Insure the part is adequately decoupled at the test head. Large changes in V_{CC} current as the device switches may cause erroneous function failures due to V_{CC} changes.
2. Do not leave inputs floating during any tests, as they may start to oscillate at high frequency.
3. Do not attempt to perform threshold tests at high speed. Following an input transition, ground current may change by as much as 400mA in 5-8ns. Inductance in the ground cable may allow the ground pin at the device to rise by 100's of millivolts momentarily.
4. Use extreme care in defining input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins and they may not actually reach V_{IL} or V_{IH} until the noise has settled. AMD recommends using $V_{IL} \leq 0$ V and $V_{IH} \geq 3$ V for AC tests.
5. To simplify failure analysis, programs should be designed to perform DC, Function, and AC tests as three distinct groups of tests.

SWITCHING TEST WAVEFORM



WF021191

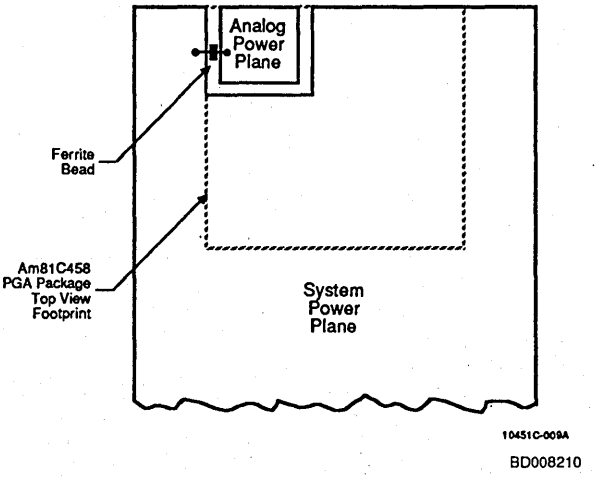
Notes: 1. Diagram show for HIGH data only. Output transition may be opposite sense.
2. Cross-hatched are is don't care condition.

Setup and Hold Times

APPENDIX A — APPLICATION NOTE FOR THE Am81C458

The design of a system using the Am81C458 should be guided by considerations similar to those used for designing precision high-speed mixed analog and digital systems. The following rules and examples are given for orientation purposes. Users may choose to design circuits, differently from the examples given here.

Power pins should be decoupled from power lines of the rest of the system. The circuit board layout should have a dedicated analog power plane. The power plane should be connected to the main power plane by wires running through ferrite beads. The analog plane should be small enough so that no digital signal passes under it (see Figures A1 and A2).



BD008210

Figure A1. Example of Power Planes Layout

There should be only one ground plane for all digital and analog ground pins, which is the same ground plane of the rest of the board. Tantalum capacitors, in parallel with a 0.1-μF ceramic capacitor, would be placed between each side of the ferrite beads and the ground plane. If too much ripple exists on the supply lines, the use of a dedicated linear regulator only for the Am81C451/458 is recommended.

The two groups of digital V_{CC} (DV_{CC}) pins should be decoupled from each other by connecting a 0.1-μF capacitor and a 0.01- μF capacitor in parallel between them and the closest group of digital ground pins (DGND).

A 0.1-μF ceramic capacitor, in parallel with a 0.001-μF chip capacitor, should be connected between each group of analog power pins (AV_{CC}) and the group of analog ground pins (AGND).

The COMP pin should also be decoupled from the power pins and the rest of the system. A 0.1-μF and a 0.01-μF chip capacitor should be connected, in parallel, between the COMP pin and the analog power plane.

Digital lines concerning the CPU interface should be kept far from pixel data lines. Pixel clock lines should be kept far from all other digital inputs. Analog outputs should be kept far from any other input. No digital line should run under the analog plane. The $\overline{CE}$ line should be as short as possible to minimize any noise picked up from other sources.

Connection with the monitor should be done through a doubly terminated 75-ohm coaxial cable. To minimize reflections, terminating resistors on the color palette side should be placed as close as possible to the R, G, B outputs.

The signals produced by the Am81C451/458, including Sync, are all positive (outgoing) currents, which when passing through the terminating resistors produce positive voltages. Since most monitors are AC-coupled, DC restoration with the proper DC levels is done inside the monitor. If a negative-going Sync (of −0.286 V) is required, DC-level shifting can be done outside the palette, prior to entering the transmission cable. Two possible circuits that produce the level shifting are shown here.

The first circuit (Figure A3) shows the 75-ohm terminating resistor relative to the green DAC, connected between the G output and a voltage source of −0.572 V. This resistor, in series with the other 75-ohm terminator inside the monitor, constitutes a voltage divider which forces the voltage on the line to be offset by half of 0.572 V (i.e., 0.286 V).

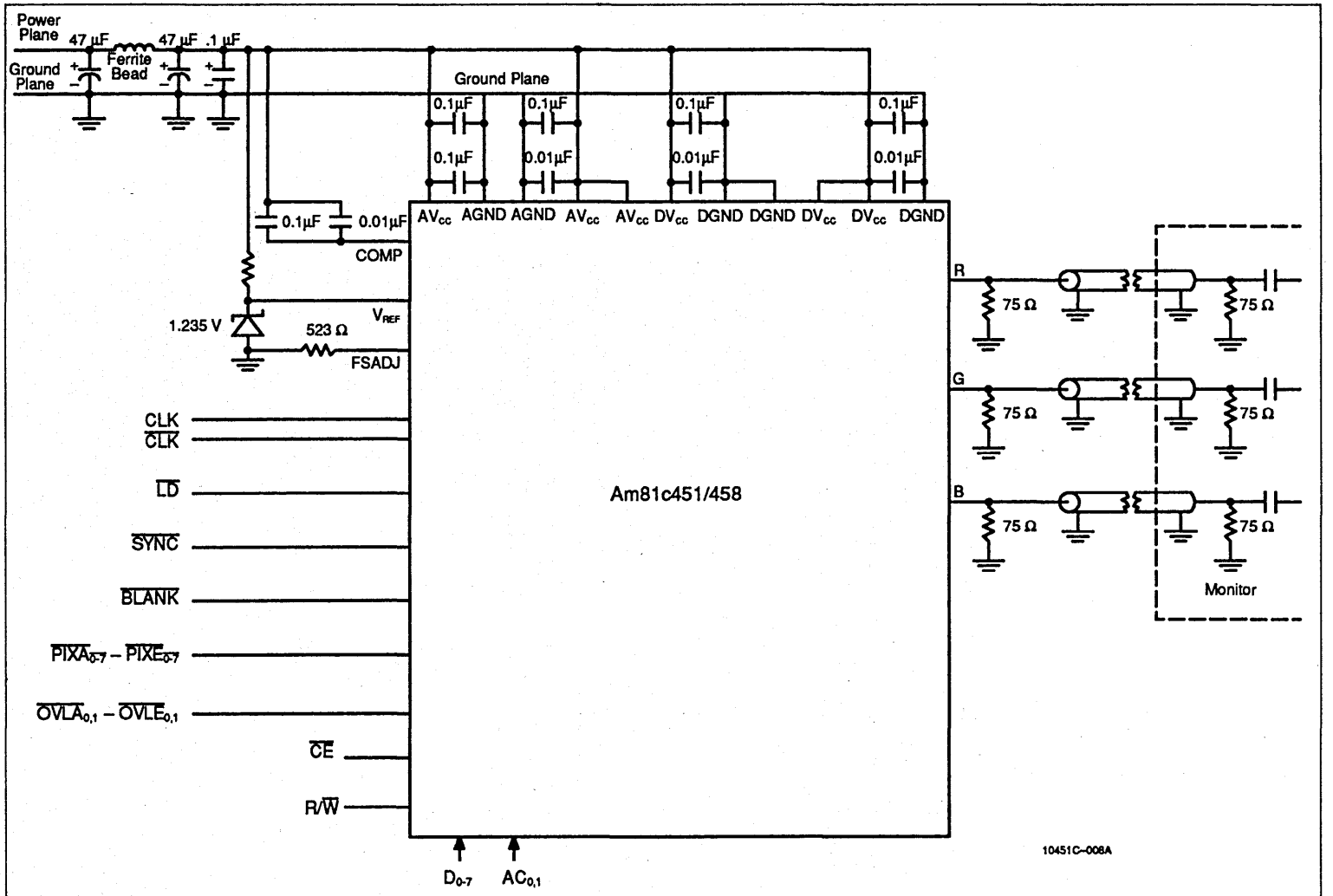


Figure A2. Am81C451/458 Connection Diagram (AC Coupling of the Monitor)

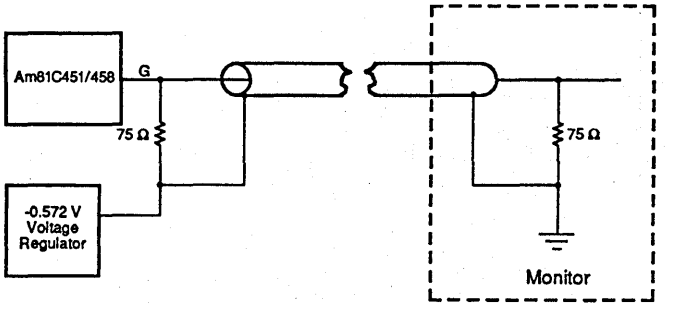


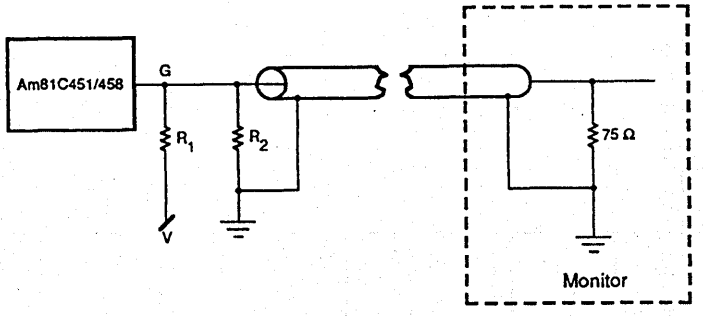
Figure A3. DC-Level Shifting Using a Voltage Regulator

The second circuit (Figure A4), useful if an exact 0.572-V voltage source is not available, shows two resistors (R_1 and R_2) instead of one, constituting the terminator at the transmitting side. R_1 is connected between the green DAC output and ground, while R_2 is connected between the green DAC output and a voltage source more negative than 0.572 V. R_1 and R_2 are such that in parallel they constitute 75 ohms, while their ratio is such that the voltage drop caused by the negative voltage source across R_1 is 0.572 V. This relationship is described by the following formulas:

$$R_2 = \frac{V \times 75}{-0.572}$$

$$R_1 = \frac{V \times 75}{V - 0.572} \quad \text{or} \quad R_1 = \frac{R_2 \times 75}{R_2 - 75}$$

The variable voltage drop across the parallel of R_1 and R_2 caused by the DAC currents adds to that caused by the external voltage source.



V	R_1	R_2
-5 V	655 Ω 1%	85 Ω 1%
-12 V	1573 Ω 1%	79 Ω 1%

Figure A4. DC-Level Shifting Using Two Resistors in Parallel

TYPICAL APPLICATION

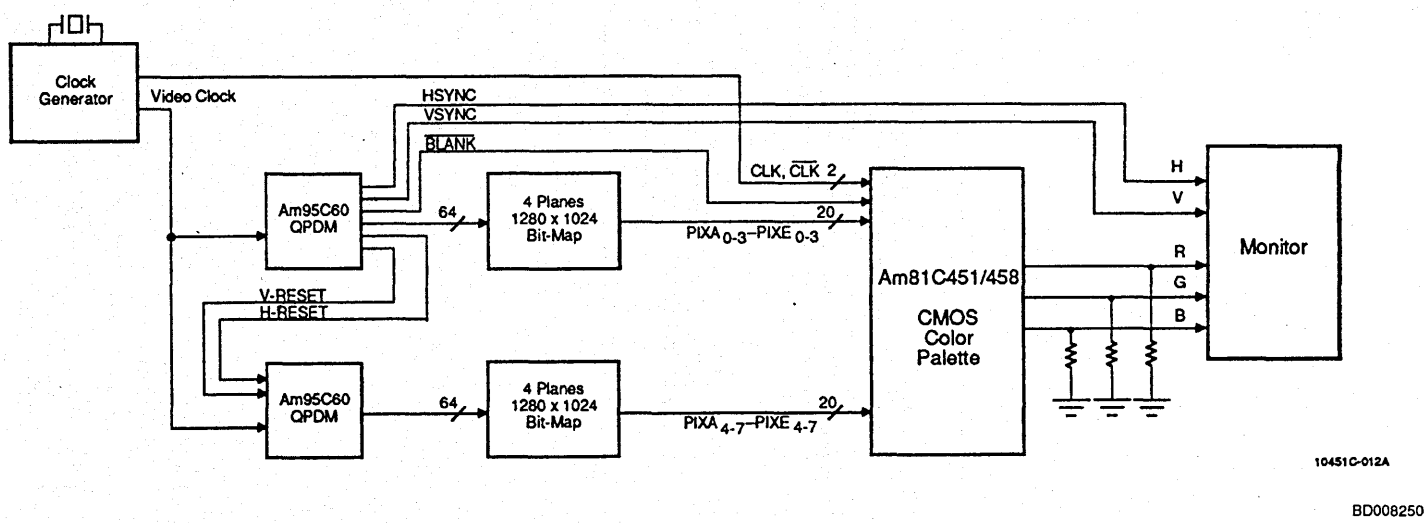


Figure A5. Application Example: Graphic System Using Two Am95C60 QPDMs and an Am81C451/458 Color Palette

Advance Information

Am81C453

CMOS Color Palette

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Plug-in replacement for Bt453
- Macintosh II compatible
- Available in 40- and 66-MHz versions
- Available in 44-pin PLCC package
- 256 x 24 color Look-Up Table (LUT)
- Triple 8-bit DACs
- RS343A/RS-170A-compatible RGB outputs
- 3 x 24 dual-port overlay RAM
- Standard MPU interface
- +5-V monolithic, high-performance CMOS

GENERAL DESCRIPTION

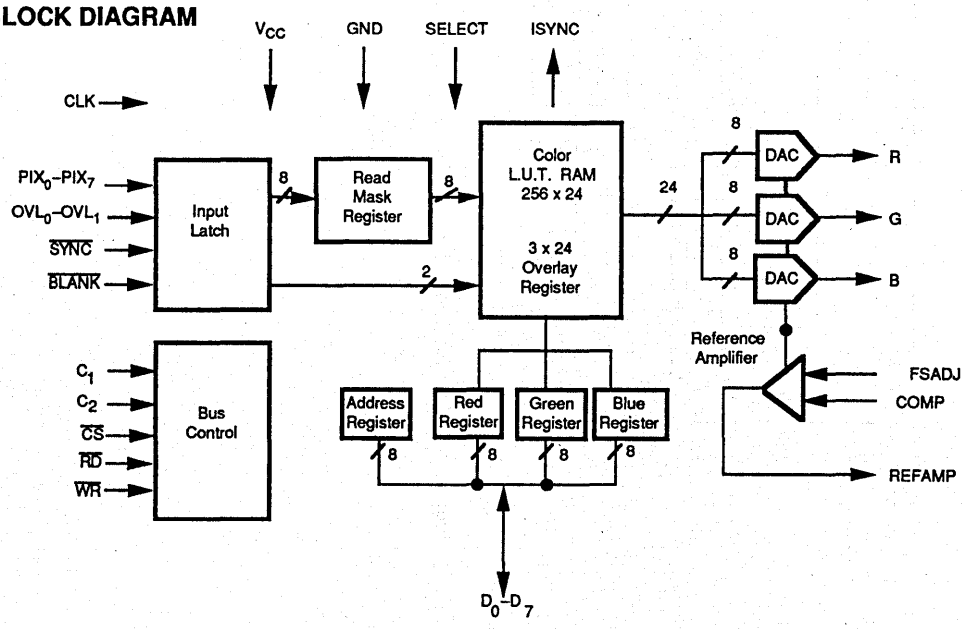
The Am81C453 CMOS Color Palette has been designed specifically for the Macintosh II market. Applications include high-resolution color graphics, CAD/CAM/CAE, image processing, and desktop publishing. The Am81C453 operates at speeds sufficient to support screen resolutions up to 1024 x 768.

The Am81C453 has an input latch, a 256 x 24 Look-Up Table, a 3 x 24 Overlay Table, and triple 8-bit video DACs. It can simultaneously display 259 colors out of an available set of 16.8 million colors. Proprietary DAC decoding techniques minimize glitch energy and skew. It is available in versions with pixel rates as high as 66 MHz.

The Am81C453 generates RS-343A-compatible outputs into doubly terminated 75-ohm loads and RS-170-compatible output into a singly terminated 75-ohm load, without external buffers. Overlaying cursors, text, grids, etc. can be implemented using the three overlay registers.

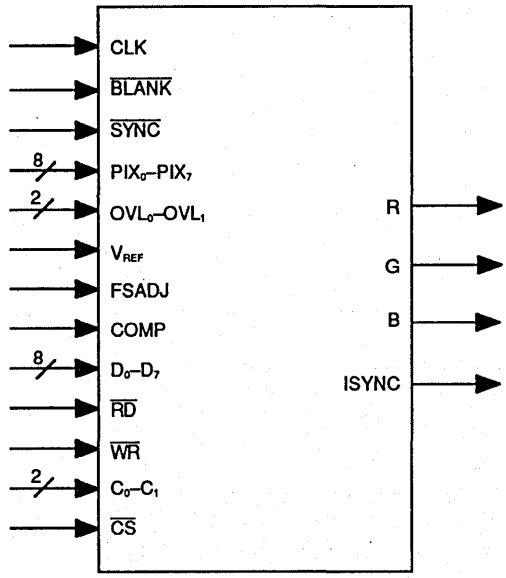
The Am81C453 is fabricated using AMD's state-of-the-art 1.2 µ CMOS process. The device is available in a 44-pin PLCC package. The Am81C453 is pin and functionally compatible with the Bt453.

BLOCK DIAGRAM

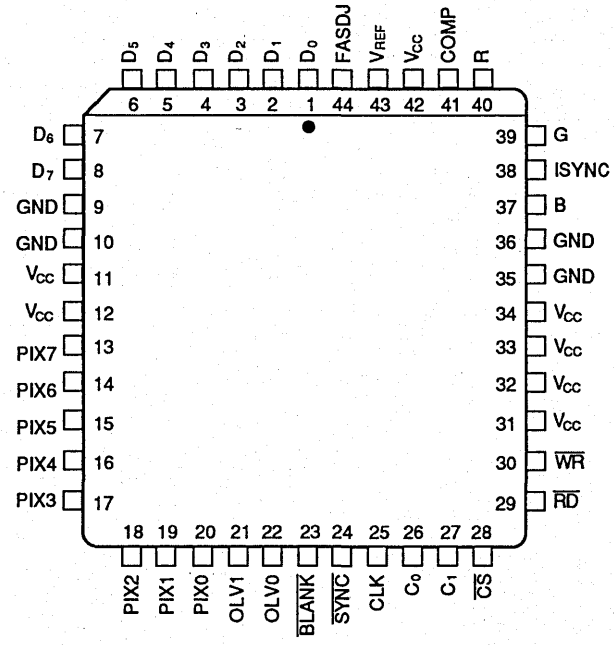


Publication #11916 Rev. A Amendment 0
Issue Date: September 1989

LOGIC SYMBOL



CONNECTION DIAGRAM
Top View

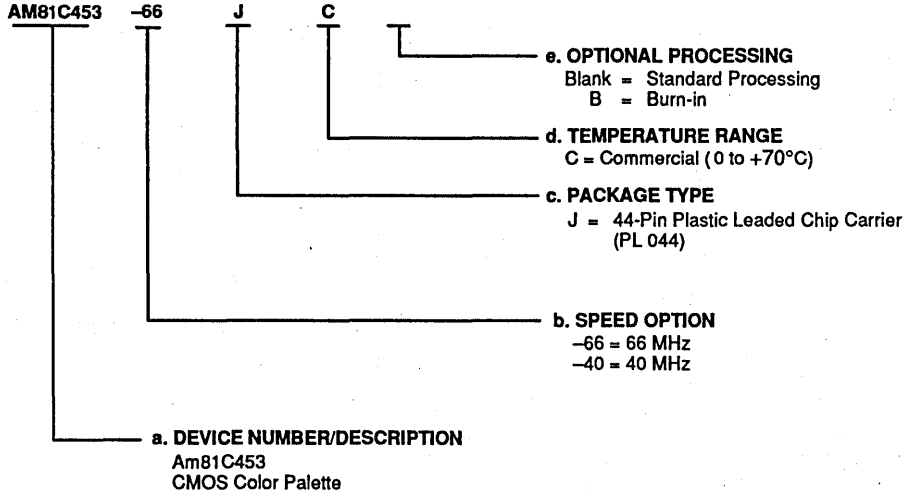


11447A-002A

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM81C453-66	JC
AM81C453-40	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Preliminary

Am81C471/478

CMOS Color Palette

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Plug-in replacements for Bt471 and Bt478
- VGA hardware and software compatible
- Available in 35, 50, 66, 80 MHz versions
- Available in 44-pin PLCC package
- 256 x 18(24) Color Palette RAM
- 15 x 18(24) Overlay RAM
- Triple 6-bit (8-bit) DACs
- Sync on all three outputs
- RS-343A/RS-170 compatible RGB outputs
- External current or voltage reference
- Standard MPU interface
- Single +5 V power supply

GENERAL DESCRIPTION

The Am81C471/478 CMOS Color Palette has been designed specifically for the VGA market. Applications include high-resolution color graphics, CAD/CAM/CAE, and desktop publishing. The Am81C471 and Am81C478 operate at speeds sufficient to support monitor sizes of 1024 X 768 pixels.

The Am81C471/478 has a 256 x 18(24) look-up table and 15 X 18(24) overlay regsiters and as such can simultaneously display 271 colors out of an available set of 65K (16.8 million) colors. The Am81C471 has triple 6-bit video DACs; the Am81C478 may be used in either 6-bit or 8-bit mode.

The Am81C471/478 also include input buffers and programmable bit-plane Read Masks. They are available in versions with pixel rates as high as 80 MHz. Proprietary DAC decoding techniques minimize glitch energy and skew.

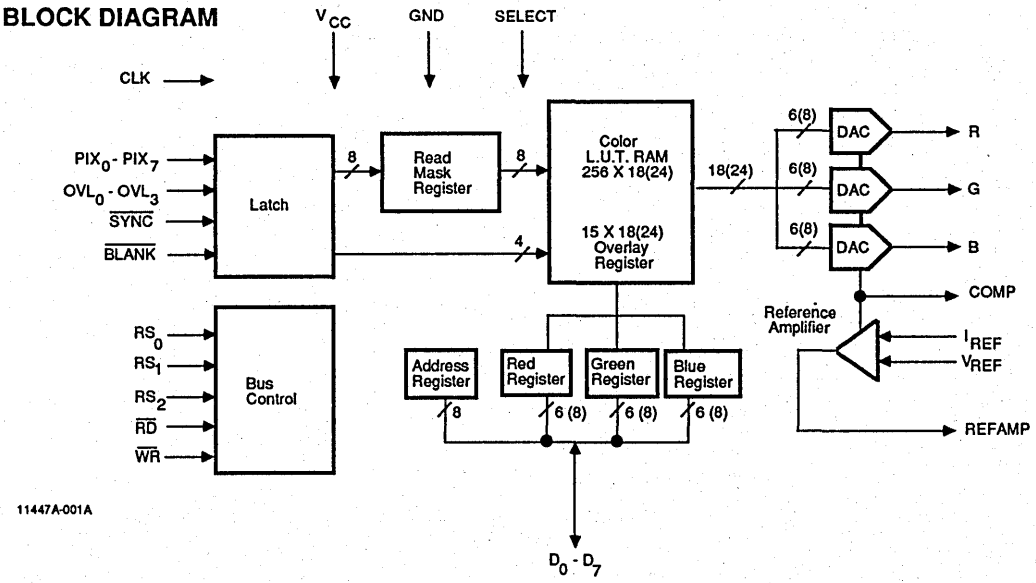
Both the Am81C471 and Am81C478 include programmable pedestals (0 or 7.5 IRE) and can be used with an external voltage or current reference. EGA emulation, overlaying cursors, text, grids, etc. can be implemented using the 15 overlay registers.

The Am81C471/478 generate RS-343A compatible outputs into doubly-terminated 75 Ω loads and RS-170 compatible output into a singly-terminated 75 Ω load, without external buffers.

The Am81C471 and Am81C478 are fabricated using AMD's state-of-the-art 1.2µ CMOS process. The devices are available in a 44-lead PLCC package.

3

BLOCK DIAGRAM



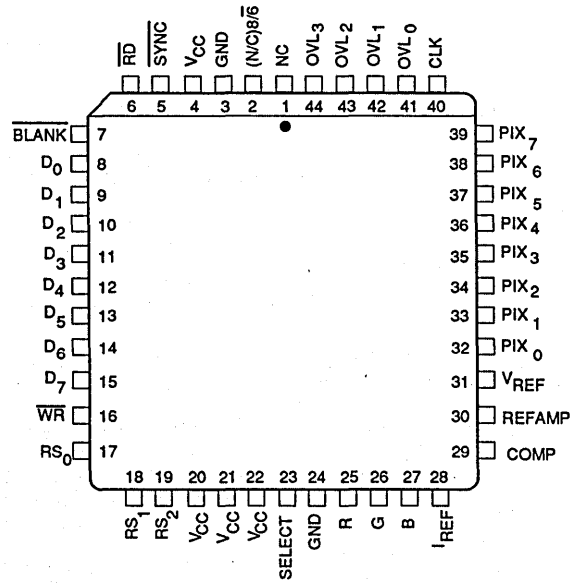
Publication #	Rev.	Amendment
11447	A	/0

Issue Date: October 1988

CONNECTION DIAGRAM

PLCC

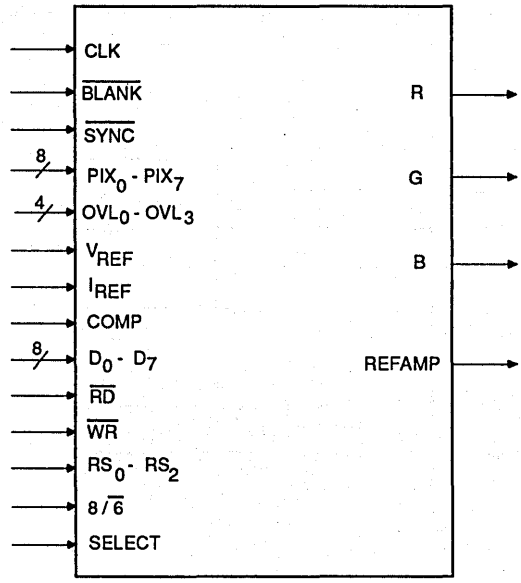
(Top View)



11447A-002A

Note: Pin 1 is marked for orientation.

LOGIC DIAGRAM



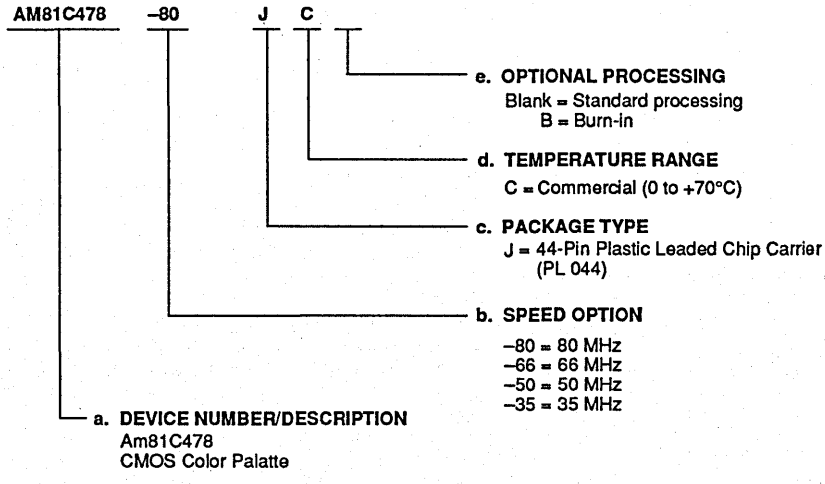
1147A-003A

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM81C471-80	JC, JCB
AM81C471-66	
AM81C471-50	
AM81C471-35	
AM81C478-80	
AM81C478-66	
AM81C478-50	
AM81C478-35	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

* Military or Limited Military temperature range products are "NPL" (Non-Compliant Products List) or Non-MIL-STD-883C Compliant products only.

3

PIN DESCRIPTION

Timing Section

CLK
Clock source pin (TTL compatible input)
This input operates at the pixel clock rate of the system. It is to be driven by a dedicated TTL buffer. The rising edge of CLK latches the $\overline{SYNC}$, $\overline{BLANK}$, PIX_0–PIX_7 and OVL_0–OVL_3 inputs.

$\overline{BLANK}$
Blank (TTL compatible input)
The $\overline{BLANK}$ input, when active, overrides the color pixel and overlay data to force the R,G, and B video outputs to their blank levels. This blank level is required during the monitor's vertical and horizontal retrace times. It is latched on the rising edge of CLK.

$\overline{SYNC}$
Sync (TTL compatible input)
The $\overline{SYNC}$ input, when active, switches off a current source on the R, G, and B video outputs. It is latched on the rising edge of CLK. Because $\overline{SYNC}$ does not override any other input or control pin, it should be asserted only during blanking intervals.

Bit Map Interface Section

PIX_0–PIX_7
Color Pixel Data addresses (TTL compatible Inputs)
These 8 inputs select which of the 256 entries in the Color Palette look-up table is to be used to provide pixel color information. They are latched into the input buffer on the rising edge of CLK. PIX_0 is the least significant bit. Unused inputs should be grounded.

OVL_0–OVL_3
Overlay Data address (TTL compatible Inputs)
These 4 inputs select which of the 15 overlay registers is to be used to provide color information. They are latched into the input buffer on the rising edge of CLK. The PIX_0–PIX_7 inputs are ignored when the overlay palette is accessed. OVL_0 is the least significant bit. Unused inputs should be grounded.

MPU Interface Section

D_0–D_7
Data and address bus (TTL compatible bi-directional)
These 8 pins are used to load and read back the Color look-up table and the internal control registers. D_0 is the least significant bit.

$\overline{RD}$
Read Control Input (TTL compatible input)
$\overline{RD}$ must be a logical zero to read data from the Color look-up table or any of the registers. During Read operations, RS_0–RS_2 are latched on the falling edge of $\overline{RD}$.

$\overline{WR}$
Write Control Input (TTL compatible input)
$\overline{WR}$ must be a logical zero to write data to the Color look-up table or any of the registers. During Write operations, RS_0–RS_2 are latched on the falling edge of $\overline{WR}$.

RS_0–RS_2
Register Select Inputs (TTL compatible Inputs)
RS_0–RS_2 allow the MPU to address any location in the Color look-up tables or any of the internal control registers. These inputs determine the type of read or write operation being performed.

$8/\overline{6}$
8-bit/6-bit Select Input (TTL compatible Input).
This pin is not used on the Am81C471 and should not be connected. For the Am81C478, this pin determines whether the MPU is reading or writing 8-bits (logical one) or 6-bits (logical zero) of information during each cycle. In 8-bit operation, D_7 is the most significant bit. In 6-bit operation D_5 is the most significant bit; D_6 and D_7 are ignored during write cycles and are logical zero during read cycles.

Analog Output Section

R
Red video output (Analog output)
Analog output of the red DAC. This output is capable of driving an RS-343A compatible doubly-terminated 75 Ω cable and an RS-170 compatible singly terminated 75 Ω cable.

G
Green video output (Analog output)
Analog output of the green DAC. This output is capable of driving an RS-343A compatible doubly-terminated 75 Ω cable and an RS-170 compatible singly terminated 75 Ω cable.

B
Blue video output (Analog output)
Analog output of the blue DAC. This output is capable of driving an RS-343A compatible doubly-terminated 75 Ω cable and an RS-170 compatible singly terminated 75 Ω cable.

V_{REF}
Voltage reference (Analog input).
An external voltage reference circuit such as the one shown in Figure A1 must supply this input with a 1.235 volt (typical) reference.

I_{REF}
Current reference (Analog input).
Voltage Mode:
When an external voltage reference is used the R_{SET} resistor connecting this pin and GND controls the magnitude of the full scale video signal according to the following relationship:

$$R_{SET}\ (ohms) = K \cdot 1{,}000 \cdot V_{REF}\ (V) / I_{out}\ (mA)$$

The table shown below defines K along with R_{SET} values for doubly-terminated 75 Ω loads.

Mode	Pedestal	K	R_{SET} (Ω)
6-bit	7.5 IRE	3.025	147
8-bit	7.5 IRE	3.200	148
6-bit	0 IRE	3.000	139
8-bit	0 IRE	3.175	140

Current Mode:
When an external current reference is used, the relationship between I_{REF} and the full scale output current on each video output is given by:

$$I_{REF}\ (mA) = I_{out}\ (mA) / K$$

REFAMP
Reference Amplifier Output (Analog output)
When an external voltage reference is used, this pin should be connected to COMP. When an external current reference is used this pin should be left floating.

COMP
Compensation capacitor connection (Analog input)
A 0.1 µF ceramic capacitor is connected between this pin and V_{CC}. This pin should be connected to REFAMP when an external voltage reference is used. When an external current reference is used this pin should be connected to I_{REF}.

Power Supply Section

V_{CC}
Analog +5 volt supply

SELECT
Select control input (TTL-compatible input)
This pin selects either a 0 IRE (SELECT = GND) or 7.5 IRE (SELECT = V_{CC}) blanking pedestal.

3

FUNCTIONAL DESCRIPTION

The Am81C471/478 CMOS Color Palette integrates all major functions required in the backend of a video system and supports pixels rates sufficient to drive most low-to-medium resolution monitors (including 1024 X 768).

A programmable look-up table maps the pixel data from a frame buffer into physical color and three Digital-to-Analog-Converters (DACs) convert the digital outputs of the look-up table into RS-343A compatible RGB analog outputs.

Microprocessor Interface

The Am81C471/478 is designed to support a standard microprocessor bus interface with direct access to 256 look-up table (LUT) RAM locations, 15 overlay registers, and two control registers. The microprocessor interface is asynchronous with respect to pixel clock. However, data transfers between the LUT RAM or overlay registers and the Red Register, Green Register, and Blue Register as shown in the block diagram are synchronized to pixel clock. Such read and write operations take one and two pixel clock cycles, respectively.

The nature of the microprocessor operation is determined by examining the RS_0–RS_2 inputs. The RS_0–RS_2 select among the address register, the 256 LUT RAM locations, 15 overlay registers, or the Read Mask register, as shown in Table 1.

Table 1. RS_0–RS_2 Decoding

RS_2	RS_1	RS_0	Function
0	0	0	Write Mode RAM Look-Up-Table Address Register
0	1	1	Read Mode RAM Look-Up-Table Address Register
0	0	1	Color Palette RAM Look-Up-Table
0	1	0	Pixel Read Mask Register
1	0	0	Write Mode Overlay Address Register
1	1	1	Read Mode Overlay Address Register
1	0	1	Overlay Registers
1	1	0	Reserved

A typical color data write cycle is initiated by setting the 8-bit Address Register with the desired address of the LUT RAM (Overlay Registers) with the proper setting of RS_0–RS_2. Next, the microprocessor performs three write cycles to the color palette: one for Red, one for Green, and one for Blue data with the RS_0–RS_2 selecting either the LUT RAM or RS_0 of the the Overlay Registers. At the end of the blue cycle the data is concatenated into a 24-bit word (18 bits for the Am81C471 or for the Am81C478 with the $8/\overline{6}$ input set LOW) and subsequently written to the LUT RAM location (Overlay Register) pointed to by the Address Register. The Address Register is then auto-incremented to point to the next LUT RAM location in the LUT RAM (to the next Overlay Register). This process may be repeated again as required by the microprocessor. See Table 2.

A typical color data read cycle is initiated by setting the 8-bit Address Register with the desired address of the LUT RAM (Overlay Registers) with the proper setting of RS_0–RS_2. Next, the microprocessor performs three read cycles to the color palette: one for Red, one for Green, and one for Blue data with the RS_0–RS_2 selecting either the LUT RAM or one of the Overlay Registers. At the end of the blue cycle the Address Register is auto-incremented to point to the next LUT RAM location in the LUT RAM (to the next Overlay Register). This process may be repeated again as required by the microprocessor. See Table 2.

For the Am81C471 (and for the Am81C478 with the $8/\overline{6}$ pin set LOW), the 6-bit color data occupies the six least significant positions of the data bus. Bits D_6 and D_7 are ignored during write cycles and are set to 0 during read cycles. Bit D_0 is the least significant bit (LSB).

The Am81C471/478 uses one 8-bit Address Register to address both the LUT RAM and the Overlay Registers as shown in Table 3. During access to the Overlay Registers the upper four bits of the Address Registers are ignored. The Address Register resets to 0 after a blue read/write cycle to the LUT RAM address 255. A user-transparent modulo-3 counter (ARb, ARa) keeps track of the read, green, and blue cycles and auto-increments at the end of each read/write access to the LUT RAM or Overlay Registers. This counter is reset to zero after a write access to the Address Register and is unchanged following a read access to the Address Register.

The Am81C471/478 uses the 8-bit Pixel Read Mask Register to modify the address of the LUT RAM as provided by PIX_0–PIX_7. The eight bits of this register are ANDed with PIX_0–PIX_7 and the result used as the address to the LUTRAM. This masking mechanism provides a quick way to alter the appearance of one or more colors on the display unit with just one microprocessor access.

Display Memory Interface

The color inputs, PIX_0–PIX_7 and OVL_0–OVL_3 are latched on the rising edge of CLK and are used as address to the 256 locations of the LUT RAM and the 15 Overlay Registers, respectively.

The total pipeline delay from the digital inputs PIX_0–PIX_7, OVL_0–OVL_3, $\overline{SYNC}$, and $\overline{BLANK}$ to the analog R, G, and B outputs is four clock cycles.

Video Generation

During each clock cycle, a 24-bit word (18-bits for the Am81C471 or for the Am81C478 with the $8/\overline{6}$ input set LOW) from either the LUT RAM or the Overlay Registers is presented to the three DACs. The three DACs convert the digital color memory output into RGB RS-343A analog format.

The $\overline{SYNC}$ and $\overline{BLANK}$ inputs are latched on the rising edge of CLK. These two inputs are responsible for adding weighted currents to RGB analog outputs as shown in Figures 1 and 2.

Tables 4 and 5 show how the $\overline{SYNC}$ and $\overline{BLANK}$ inputs change analog output levels.

The $\overline{SYNC}$ and $\overline{BLANK}$ inputs are routed to the three DACs after a delay equal to four clock periods, identical to the delay incurred by the video stream.

To specify which blanking pedestal is to be applied, the SELECT input is used; if SELECT is set to logical zero, the blanking pedestal is 0.0 IRE and if SELECT is set to logical one, the blanking pedestal is 7.5 IRE.

The three analog outputs of the Am81C471/478 are each capable of driving a doubly terminated 75 Ω coaxial cable.

Table 2. Read/Write Access to the Am81C471/478

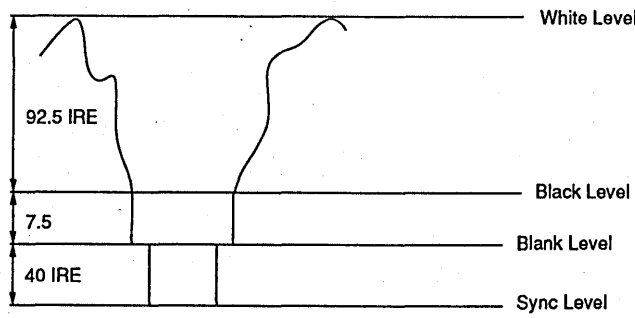
$\overline{RD}$	$\overline{WR}$	RS_2	RS_1	RS_0	ARb	ARa	FUNCTION
1	0	0	0	0	X	X	Write Address Register; (LUT Write) AR(7:0) ← D(7:0); ARb, ARa ← 0
1	0	0	0	1	0	0	Write Red Color; RREG(7:0) ← D(7:0); INC. ARb, ARa
1	0	0	0	1	0	1	Write Green Color; GREG(7:0) ← D(7:0); INC. ARb, ARa
1	0	0	0	1	1	0	Write Blue Color; BREG(7:0) ← D(7:0); ARb, ARa ← 0;
							Write Color Look-Up-Table; R(7:0) ← RREG; G(7:0) ← GREG; B(7:0) ← BREG; INC AR(7:0)
1	0	1	0	0	X	X	Write Address Register; (Overlay Write) AR(7:0) ← D(7:0); ARb, ARa ← 0
1	0	1	0	1	0	0	Write Red Color; RREG(7:0) ← D(7:0); INC. ARb, ARa
1	0	1	0	1	0	1	Write Green Color; GREG(7:0) ← D(7:0); INC. ARb, ARa
1	0	1	0	1	1	0	Write Blue Color; BREG(7:0) ← D(7:0); ARb, ARa ← 0;
							Write Overlay Register; ; R(7:0) ← RREG; G(7:0) ← GREG; B(7:0) ← BREG; INC AR(7:0)
1	0	0	1	1	X	X	Write Address Register; (LUT Read) AR(7:0) ← D(7:0); ARb, ARa ← 0; RREG ← R(7:0); GREG ← G (7:0); BREG ← B(7:0)
0	1	0	0	1	0	0	Read LUT Red; D (7:0) ← RREG(7:0); INC. ARb, ARa
0	1	0	0	1	0	1	Read LUT Green; D(7:0) ← GREG(7:0); INC. ARb, ARa
0	1	0	0	1	1	0	Read LUT Blue; D(7:0) ← BREG(7:0); ARb, ARa ← 0; INC AR(7:0)
1	0	1	1	1	X	X	Write Address Register; (Overlay Read) AR(7:0) ← D(7:0); ARb, ARa ← 0; RREG ← R(7:0); GREG ← G(7:0); BREG ← B(7:0)
0	1	1	0	1	0	0	Read Overlay Red; D(7:0) ← RREG(7:0); INC. ARb, ARa
0	1	1	0	1	0	1	Read Overlay Green ; D(7:0) ← GREG(7:0); INC. ARb, ARa
0	1	1	0	1	1	0	Read Overlay Blue ; D(7:0) ← BREG(7:0); ARb, ARa ← 0; INC AR(7:0)

Note: Refer to timing diagrams for edge information on $\overline{RD}$ and $\overline{WR}$.

Table 3. Address Register Operation

AR(7:0)	RS_2	RS_1	RS_0	Function
$00-$FF	0	0	1	Color Look-Up-Table location $00-$FF
XXXX-0000	1	0	1	Reserved
XXXX-0001	1	0	1	Overlay Register 1
XXXX-0010	1	0	1	Overlay Register 2
XXXX-0011	1	0	1	Overlay Register 3
:	:	:	:	:
XXXX-1111	1	0	1	Overlay Register 15

V	mA
1.000	26.67
0.340	9.05
0.286	7.62
0.000	0.00



Note: 75 Ω doubly-terminated load, SELECT = V_{CC}. External voltage or current reference adjusted for 26.67 mA full scale output. RS-343A levels and tolerances assumed on all levels.

11447A-004A

Figure 1. Composite Video Output Signals (SELECT = V_{CC})

Table 4. Video Output Truth Table (SELECT = V_{CC})

Description	$\overline{SYNC}$	$\overline{BLANK}$	I_{out} (mA)	DAC Input Data
WHITE	1	1	26.67	$FF
DATA	1	1	data + 9.05	data
DATA–SYNC	0	1	data + 1.44	data
BLACK	1	1	9.05	$00
BLACK-SYNC	0	1	1.44	$00
BLANK	1	0	7.62	$xx
SYNC	0	0	0	$xx

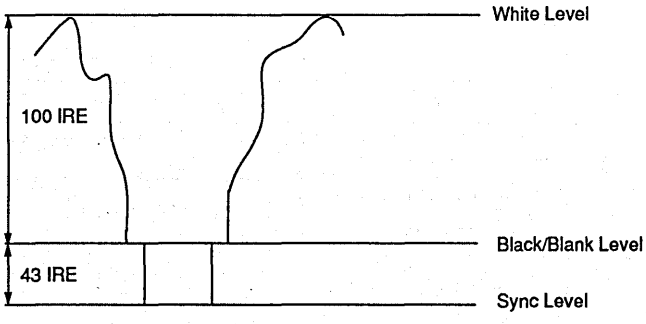
Note: Typical will full scale G = 26.67 mA. SELECT = V_{CC}. External voltage or current reference adjusted for 26.67 mA full scale output.

Table 5. Video Output Truth Table (SELECT = GND)

Description	$\overline{SYNC}$	$\overline{BLANK}$	I_{out} (mA)	DAC Input Data
WHITE	1	1	26.67	$FF
DATA	1	1	data + 8.05	data
DATA–SYNC	0	1	data	data
BLACK	1	1	8.05	$00
BLACK-SYNC	0	1	0	$00
BLANK	1	0	8.05	$xx
SYNC	0	0	0	$xx

Note: Typical will full scale G = 26.67 mA. SELECT = GND. External voltage or current reference adjusted for 26.67 mA full scale output.

V	mA
1.000	26.67
0.302	8.05
0.000	0.00



Note: 75 Ω doubly-terminated load, SELECT = GND. External voltage or current reference adjusted for 26.67 mA full scale output.

11447A-005A

Figure 2. Composite Video Output Signals (SELECT = GND)

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65 to +150°C
Ambient Temperature
  Under Bias ..... –55 to +125°C
  Junction Temperature ..... +175°C
Supply Voltage to Ground Potential
  Continuous ..... –0.5 to 7.0 V
DC Voltage Applied to ..... GND–0.5 V to V_{CC} Max+0.5 V
  Outputs for HIGH
  Output State
DC Input Voltage GND ..... –0.3 to V_{CC} +0.3 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
  Ambient Temperature (T_A) ..... 0 to +70°C
  Supply Voltage (V_{CC})
  for 80 MHz devices ..... +4.75 to +5.25 V
  for 35, 50, 66 MHz devices ..... +4.50 to +5.50 V
Reference Voltage ..... +1.14 to +1.26 V
  (Voltage Reference Configuration)
I_{REF} Current ..... –3 to –10 mA
  (Current Reference Configuration)
Output Load ..... 37.5 Ω

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS (Over operating range)

Parm. Num	Parm. Symbol	Parameter Description	Test Conditions	Min	Typ	Max	Unit
Digital Inputs							
	V_{IH}	Input High Voltage		2.0		V_{CC}+0.5	V
	V_{IL}	Input Low Voltage		GND–0.5		0.8	V
	I_{IH}	Input High Current	V_{in}=2.4 V			1	µA
	I_{IL}	Input Low Current	V_{in}=0.4 V			–1	µA
	C_{IN}	Input Capacitance	f=1 MHz, V_{in}= 2.4 V			7	pF
Digital Outputs							
	V_{OH}	Output High Voltage	I_{OH} = –400 µA	2.4			V
	V_{OL}	Output Low Voltage	I_{OL} = 3.2 mA			.4	V
	I_{OZ}	3-State Current				50	µA
	C_{out}	Output Capacitance				7	pF
Analog Outputs							
		Resolution (each DAC)		6 (8)	6 (8)	6 (8)	Bits
		Accuracy (each DAC)					
	LIN_i	Integral Linearity Error				±1/4(1)	LSB
	LIN_d	Differential Linearity Error				±1/4(1)	LSB
		Gray Scale Error				±5	%Gray
		Monotonicity			Guaranteed		
		Coding	Binary				Binary
		Output Current					
		White Level Relative to Blank		17.69	19.05	20.4	mA
		White Level Relative to Black		16.74	17.62	18.5	mA
		Black Level Relative to Blank	SELECT = V_{CC}	0.95	1.44	1.9	mA
			SELECT = GND	0	5	50	µA
		Blank Level		6.29	7.62	8.96	mA
		Sync Level		0	5	50	µA
		LSB Size					
		Am81C478 (8/$\overline{6}$=Logical one)			69.1		µA
		Am81C471			279.68		µA
		DAC-to-DAC Matching			2	5	%
	R_{out}	Output Impedance			10		KΩ
	C_{out}	Output Capacitance	f=1 MHz, I_{out}=0 mA			30	pF
	I_{VREF}	Voltage Reference Input Current			100		µA
	PSSR	Power Supply Rejection Ratio	COMP=0.1 pF, f=1 KHz			0.5	%/%V_{CC}

PRELIMINARY

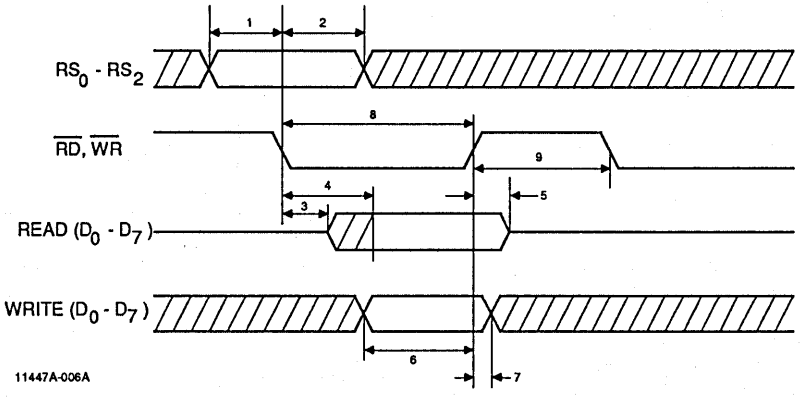
SWITCHING CHARACTERISTICS (Over operating range)

Parm. Num.	Parm. Symbol	Parameter Description	80 MHz Min	80 MHz Typ	80 MHz Max	66 MHz Min	66 MHz Typ	66 MHz Max	50 MHz Min	50 MHz Typ	50 MHz Max	35 MHz Min	35 MHz Typ	35 MHz Max	Units
		Clock Rate			80			66			50			35	MHz
1	t_S	RS_0–RS_2 Setup Time	10			10			10			15			ns
2	t_H	RS_0–RS_2 Hold Time	10			10			10			15			ns
3	t_P	$\overline{RD}$ Asserted to Data Bus Driven	5			5			5			5			ns
4	t_P	$\overline{RD}$ Asserted to Data Valid			40			40			40			40	ns
5	t_P	$\overline{RD}$ Negated to Data Bus 3-stated			20			20			20			20	ns
6	t_S	Write Data Setup Time	10			10			10			15			ns
7	t_H	Write Data Hold Time	10			10			10			15			ns
8	t_W	$\overline{RD}$, $\overline{WR}$ Pulse Width Low	50			50			50			50			ns
9	t_W	$\overline{RD}$, $\overline{WR}$ Pulse Width High	4 X t_{cyc}			4 X t_{cyc}			4 X t_{cyc}			4 X t_{cyc}			ns
10	t_S	Pixel and Overlay Setup Time	3			3			3			4			ns
11	t_H	Pixel and Overlay Hold Time	3			3			3			4			ns
12	t_{CYC}	Clock Cycle Time	12.5			15.2			20			28			ns
13	t_W	Clock Pulse Width High Time	4			5			6			7			ns
14	t_W	Clock Pulse Width Low Time	4			5			6			9			ns
15	t_P	Analog Output Delay			30			30			30			30	ns
16	t_R, t_F	Analog Output Rise/Fall Time (Note 1)		3			3			3			3		ns
17	t_S	Analog Output Settling Time (Note 1)		13			16			20			28		ns
		Clock and Data Feedthrough		–30			–30			–30			–30		dB
		Glitch Impulse (Note 2)		75			75			75			75		pV–sec
		DAC to DAC Crosstalk		–23			–23			–23			–23		dB
		Analog Output Skew			2			2			2			2	ns
		Pipeline Delay		4			4			4			4		clocks
	I_{CC}	V_{CC} Supply Current (Note 3)		160	200		160	200		160	200		160	200	mA

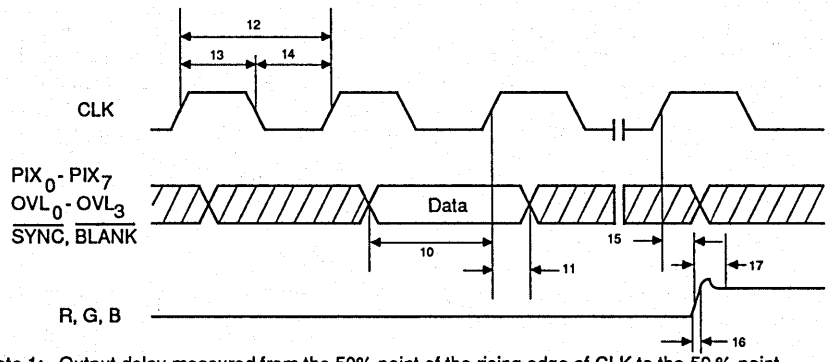
PRELIMINARY

Notes:
1. Clock and data feedthrough are not included
2. Included clock and data feedthrough, –3 dB bandwidth = 2 x clock frequency
3. Measured at maximum f_{CLK};
   I_{CC}(Max.): V_{CC} = 5.25 V, T_A = 0°C
   I_{CC}(Typ.): V_{CC} = 5.0 V, T_A + 25°C

Test Conditions:
TTL Input Level: 0 to 3 V with t_R, t_F (10–90%) ≤ 3 ns
R_{set}=148 Ω (Am81C478); R_{set}=147 Ω (Am81C471)
V_{ref}=1.235 V, Select = V_{CC}, 8/6=Logical '1'
Analog Output Load ≤ 10 pF, D_0–D_7 Output Load ≤ 50 pF



MPU Read / Write Timing



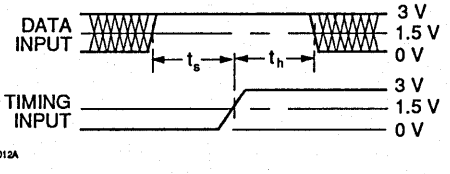
Note 1: Output delay measured from the 50% point of the rising edge of CLK to the 50 % point of the full scale transition

Note 2: Settling time measured from the 50% point of the full scale transition to the output remaining within ± LSB (Am81C478) or ± 1/4 LSB (Am81C471)

Note 3: Output rise/fall time measured between the 10% and 90% points of full scale transition

11447A-007A

Video Input / Output Timing



Notes: 1. Diagram shown for HIGH data only. Output transition may be opposite sense.
2. Cross-hatched areas are don't care condition.

Switching Test Waveform

APPENDIX A: APPLICATION NOTE FOR THE AM81C471/478

The design of a system using the Am81C471/478 should be guided by considerations similar to those used for designing precision high-speed mixed analog and digital systems. The following rules and examples are given for orientation purposes. Users may choose to design circuits, which may differ considerably from the examples given here.

The Am81C471/478 can be configured in either current or voltage reference modes. The voltage reference scheme, as shown in Figure A1, is preferred over the current source scheme because of the following features: superior power supply rejection, temperature compensation, lower power, fewer components and overall simplicity of design. The current reference mode, as shown in Figure A2, requires more components to provide adequate temperature compensation, and power supply rejection.

Power pins should be decoupled from the power lines in the rest of the system. The Am81C471/478 should have its own power plane; this plane should provide power to all its power pins and voltage/current reference circuits. This plane should be connected to the main power plane by a wire running through ferrite beads as shown in Figures A1 and A2.

If further noise reduction is required on the analog outputs then a separate analog power plane and digital power plane should be used and connected to pins (21,22) and (4,20) respectively. These two power planes should be connected to the main power plane by wires running through ferrite beads. The analog plane should not be crossed by any digital signal. Only one ground plane should be used, directly connected to the rest of the system. Tantalum capacitors, in parallel with a 0.1 µF ceramic capacitor, should be placed between each side of the ferrite beads and the ground plane. If too much ripple exists on the lines, the use of a dedicated linear regulator is recommended.

The COMP pin should be decoupled from the power pins and the rest of the system. A 0.1 µF ceramic capacitor should be connected in parallel between this pin and the analog power pins (21,22).

Digital lines accessible by the MPU interface should be kept far from pixel data lines. Pixel clock lines should be kept far from all other digital inputs. Analog outputs should be kept far from any other input. No digital line should run under the analog plane.

Connection to the monitor should be done through a doubly-terminated 75 Ω coaxial cable. To minimize reflections, terminating resistors on the color palette side should be placed as close as possible to the R, G, and B outputs.

The signals produced by the Am81C471/478 are all positive currents, which when passing through the terminating resistors produce positive voltages. Since most monitors are AC-coupled, DC restoration with the proper DC level is done inside the monitor. If a negative-going sync (–0.286V) is required, DC level shifting can be done outside the palette, prior to entering the transmission cable. The circuit that produces this level shifting is shown in Figure A3. This circuit shows two resistors, R_1 and R_2, as the termination at the transmitting side. R_1 is connected between R (G, or B) outputs and ground, while R_2, is connected between R (G, or B) outputs and a voltage source more negative than –0.572V. R_1 and R_2 are such that in parallel they equal 75 Ω, while their ratio is such that the voltage drop caused by the negative voltage source across R_1 is 0.572 V. The relationship is described by the following formulae:

$$R_1 \parallel R_2 = 75\ \Omega\ ;\ R_1 = V \cdot 75/(V+0.572);\ R_2 = V \cdot 75/(-0.572)$$

Figure A4 shows a VGA system in which a VGA graphics controller is paired with Am81C471/478 in a typical PC board design.

The Am81C471 has 6-bit µP interface which is compatible with existing VGA software. To accomodate 8-bit software the Am81C478 can be used. By driving the $8/\overline{6}$ input pin from the µP interface the software can automatically change the DAC resolution (bits per channel) according to its own requirements.

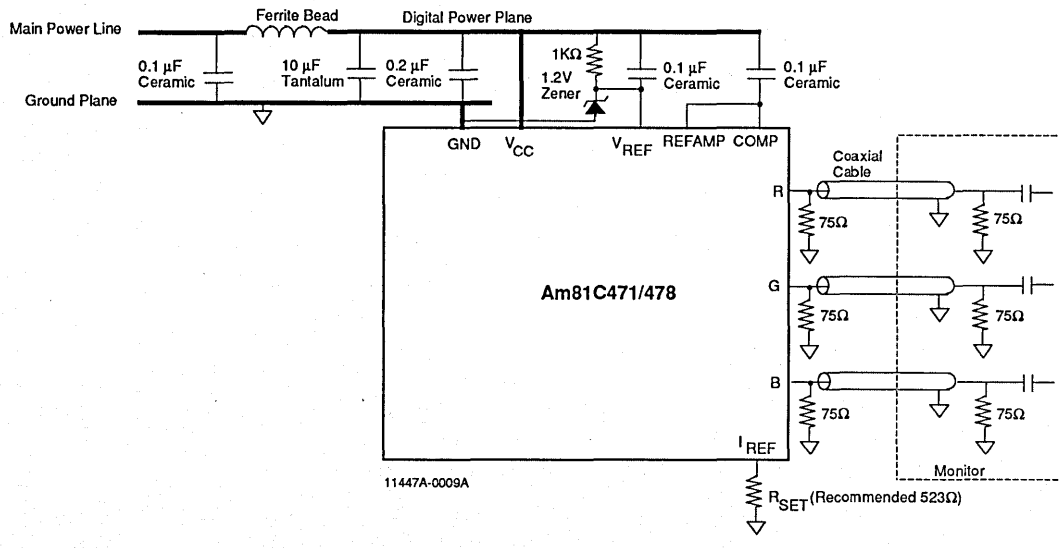


Figure A1. Am81C471/478 Voltage Reference Connection Diagram (AC Coupling of the Monitor)

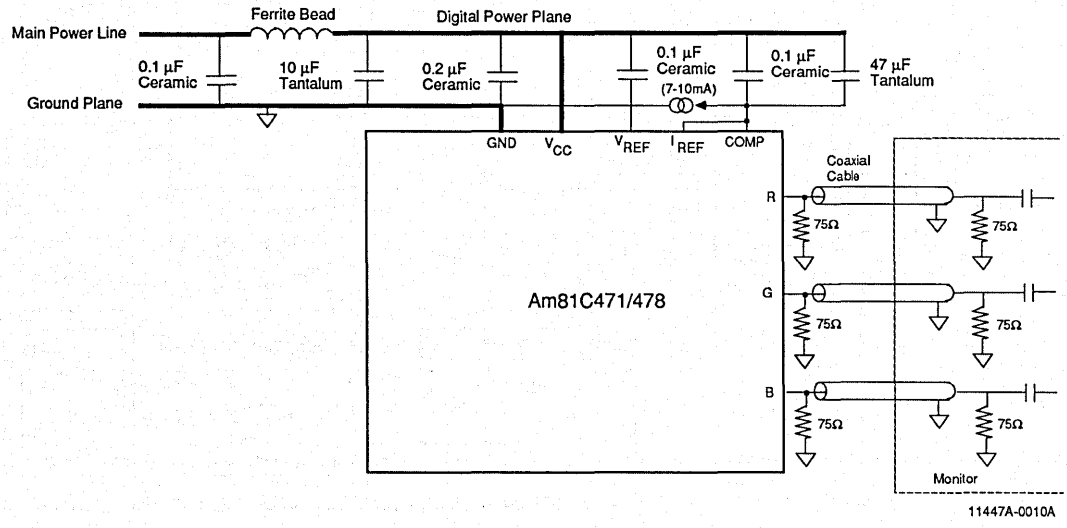
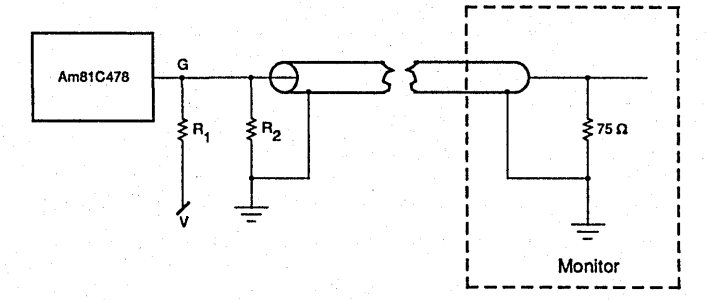


Figure A2. Am81C471/478 Current Reference connection Diagram (AC Coupling of the Monitor)



V	R_1	R_2
-5 V	655 Ω 1%	85 Ω 1%
-12 V	1573 Ω 1%	79 Ω 1%

Figure A3. DC-Level Shifting Using Two Resistors in Parallel

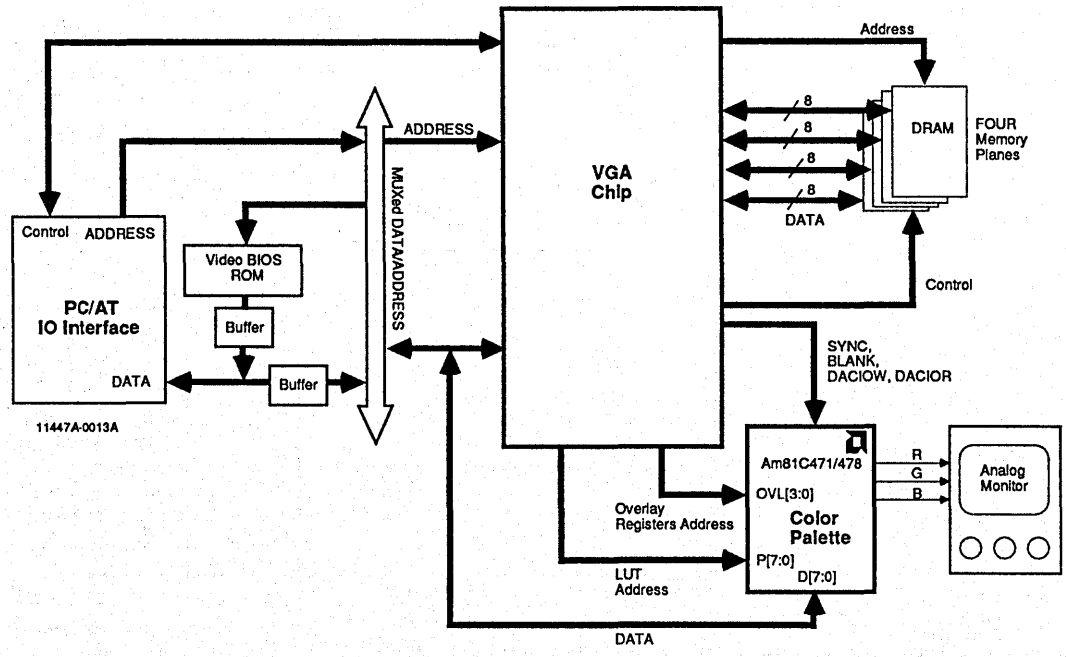


Figure A4. Application Example: VGA system using a VGA chip and an Am81C471/478

Advance Information

Am81EC176

CMOS Color Palette

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

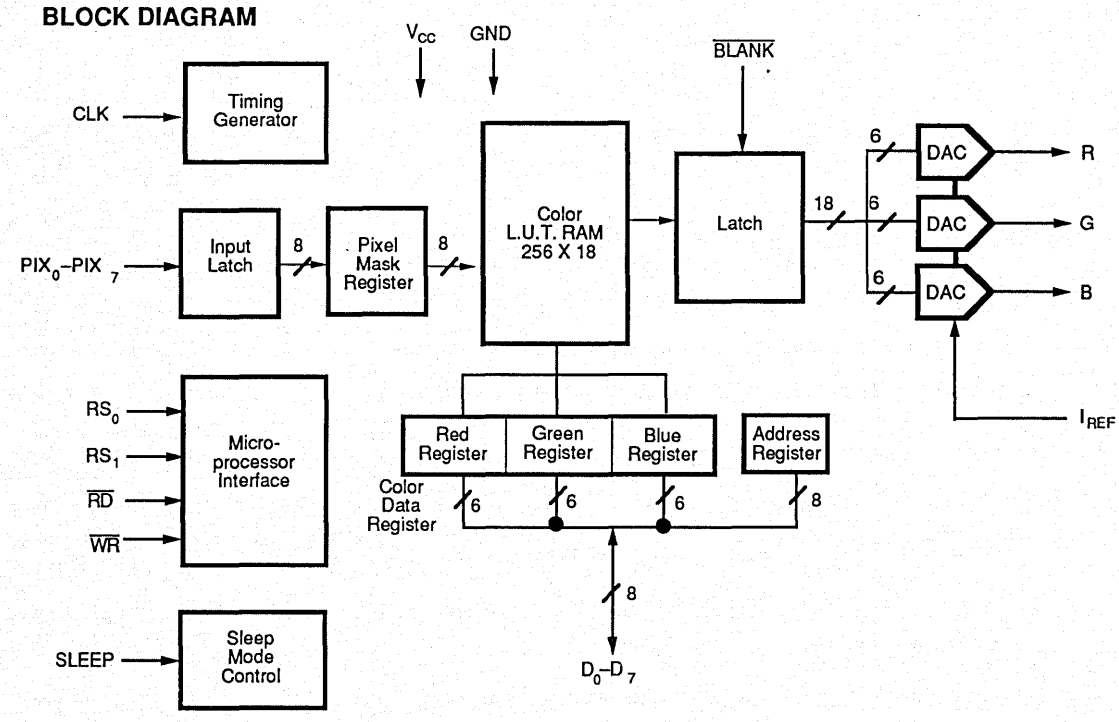
- VGA hardware and software compatible with low-power Sleep mode
- Consumes approximately 2 mA in Sleep mode
- Compatible with the Am81C176
- Clock rates up to 80 MHz
- Available in 32-pin PLCC package
- 256 x 18 color Look-Up Table (LUT)
- Triple 6-bit DACs
- RS-170A-compatible RGB outputs
- External current reference
- Asynchronous MPU interface
- Single monolithic, high-performance CMOS
- Single +5-V power supply

GENERAL DESCRIPTION

The Am81EC176 has been designed specifically for laptop personal computer manufacturers offering VGA compatibility. This part is hardware and software compatible with the Am81C176. The Am81EC176 operates at speeds up to 80 MHz and can support monitors with resolutions up to 1024 X 768.

The low-power option is enabled by pulling the SLEEP pin High. Enabling the Sleep feature reduces power consumption of the Am81C176 by about 98%.

BLOCK DIAGRAM



Publication #12436 Rev. A Amendment 0
Issue Date: September 1989

GENERAL DESCRIPTION (continued)

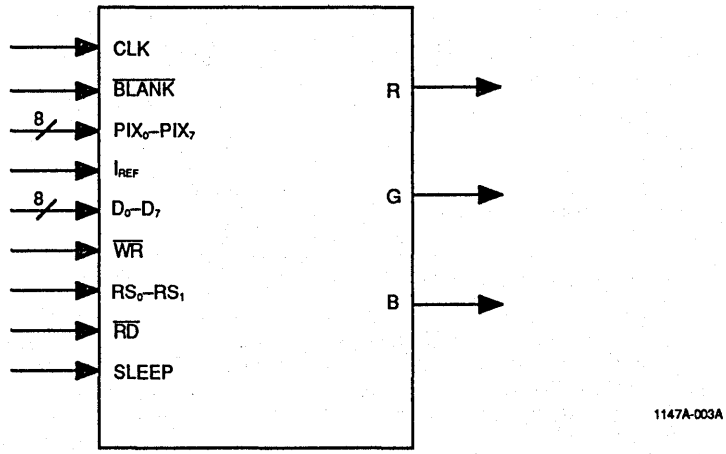
The Am81EC176 has a 256 x 18 Look-Up Table and triple 6-bit DACs. It can simultaneously display 256 colors out of an available set of 256K colors.

Because of a proprietary technique, Read and Write operations to the color Look-Up Table may occur during active video.

The Am81EC176 generates RS-170A-compatible outputs into doubly terminated 75-ohm loads, without external buffers.

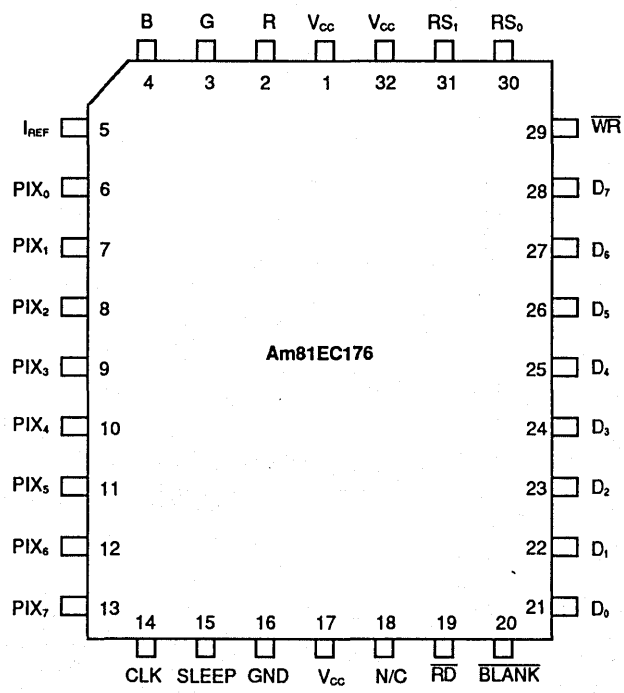
The Am81EC176 is fabricated using AMD's state-of-the-art 1.2-µ CMOS process. The device is available in a 32-lead PLCC package.

LOGIC SYMBOL



CONNECTION DIAGRAM

PL 032

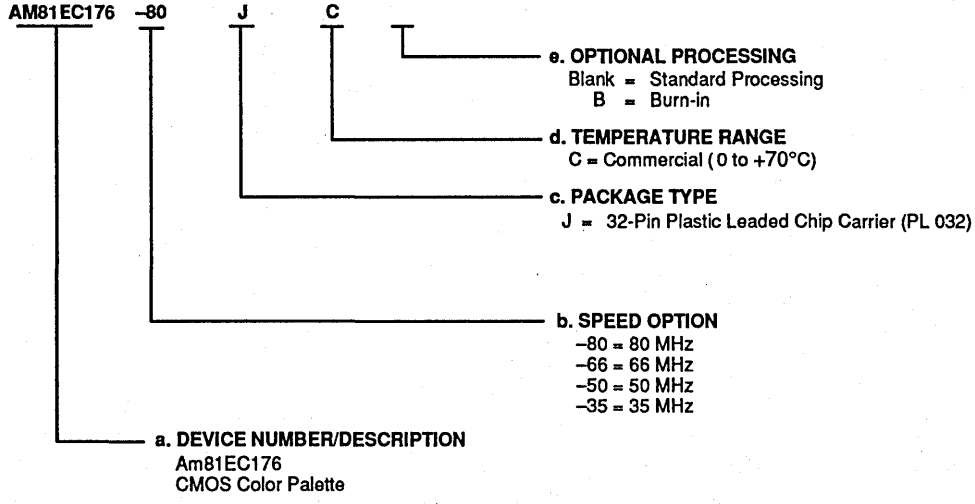


3

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM81EC176-80	JC, JCB
AM81EC176-66	
AM81EC176-50	
AM81EC176-35	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Advance Information

Am81EC471/478

CMOS Color Palette

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- VGA-compatible color palette with low-power Sleep mode
- Consumes only 2 mA power in Sleep mode
- Sleep mode enabled by hardware or software method
- Compatible with Am81C471 (Am81C478)
- Available in 35-, 50-, 66-, and 80-MHz versions
- Available in 44-pin PLCC package
- 256 × 18(24) color palette RAM
- 15 × 18(24) overlay RAM
- Triple 6-bit (8-bit) DACs
- Sync on all three outputs
- R-343A/RS-170-compatible RGB outputs
- External current or voltage reference
- Standard MPU interface
- Single +5-V power supply

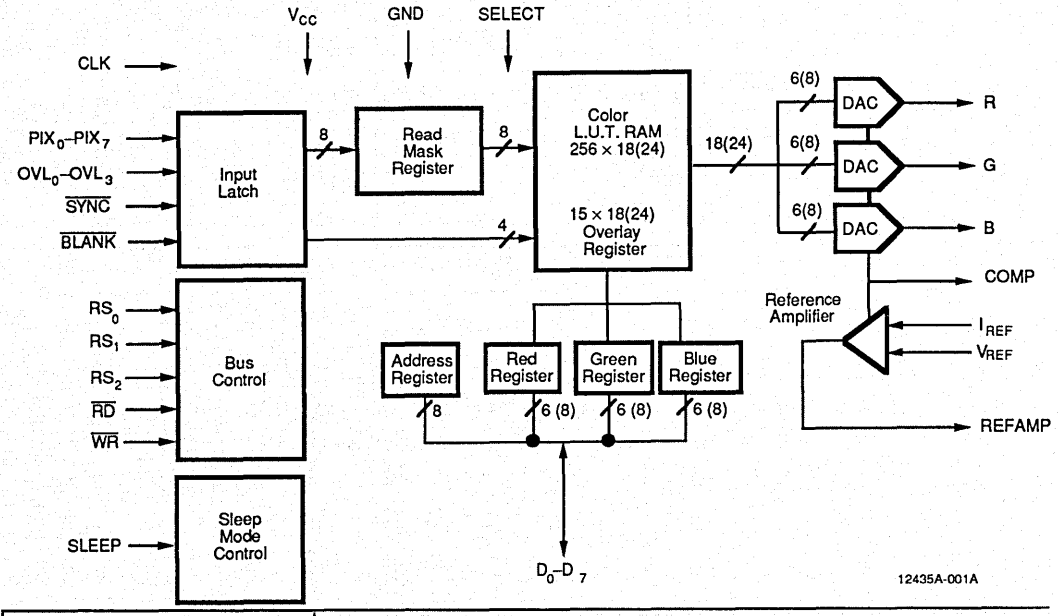
GENERAL DESCRIPTION

The Am81EC471/478 CMOS Color Palette has been designed specifically for manufacturers of laptop personal computers that will offer the VGA feature. The part is software- and hardware-compatible with the Am81C471/478 and includes a low-power (sleep) feature. The Am81EC471 and Am81EC478 operate at speeds sufficient to support screen resolutions up to 1024 × 768 pixels.

The low-power option can be enabled using two methods. The first method (software) method requires only a write to an internal register. The second method (hardware) requires pulling the SLEEP pin high. Enabling the sleep feature reduces the power consumption of the Am81EC471/478 by about 98%.

3

BLOCK DIAGRAM



Publication #12435 Rev. A Amendment /0
Issue Date: September 1989

GENERAL DESCRIPTION (continued)

The Am81EC471/478 has a 256 × 18(24) Look-Up Table and 15 × 18(24) Overlay Table and as such can simultaneously display 271 colors out of an available set of 256K (16.8 million) colors. The Am81EC471 has triple 6-bit video DACs; the Am81EC478 may be used in either 6-bit or 8-bit mode.

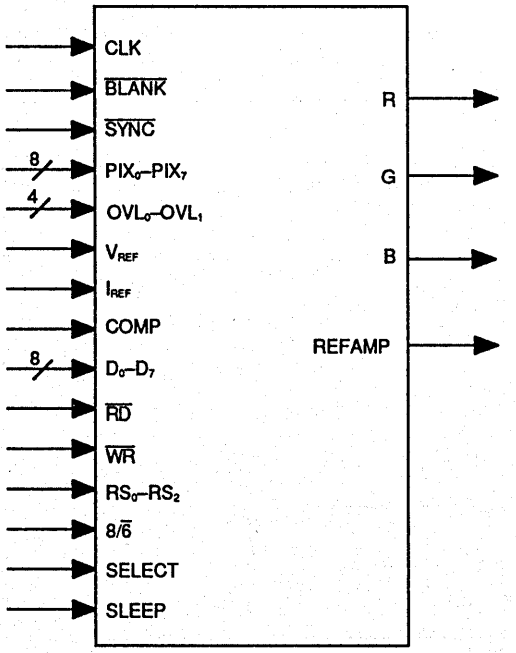
The Am81EC471/478 includes an input latch and programmable bit-plane Read Mask. They are available in versions with pixel rates as high as 80 MHz. Proprietary DAC decoding techniques minimize glitch energy and skew.

Both the Am81EC471 and Am81EC478 include programmable pedestals (0 or 7.5 IRE) and can be used with an external voltage or current reference. EGA emulation, overlaying cursors, text, grids, etc., can be implemented using the 15 overlay registers.

The Am81EC471/478 generates RS-343A-compatible outputs into doubly-terminated 75-ohm loads and RS-170-compatible output into a singly-terminated 75-ohm load, without external buffers.

The Am81EC471 and Am81EC478 are fabricated using AMD's state-of-the-art 1.2-µ CMOS process. The devices are available in a 44-lead PLCC package.

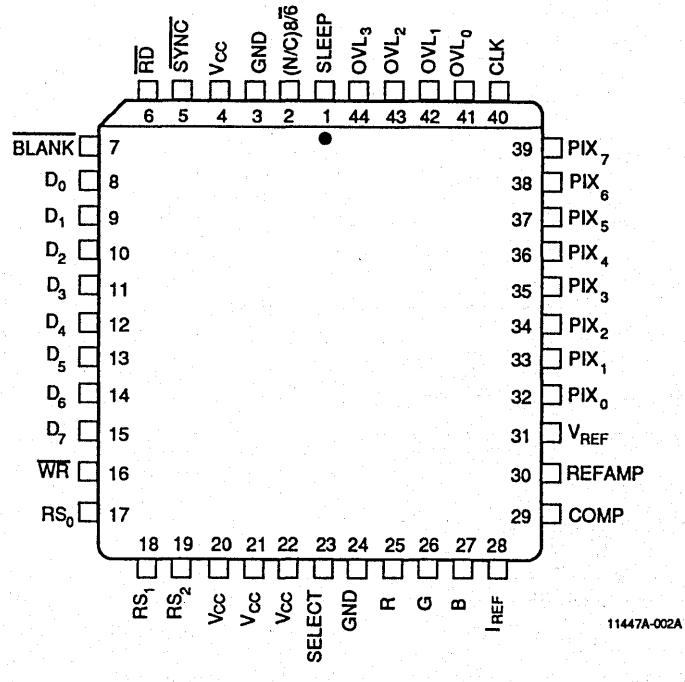
LOGIC SYMBOL



1147A-003A

CONNECTION DIAGRAM

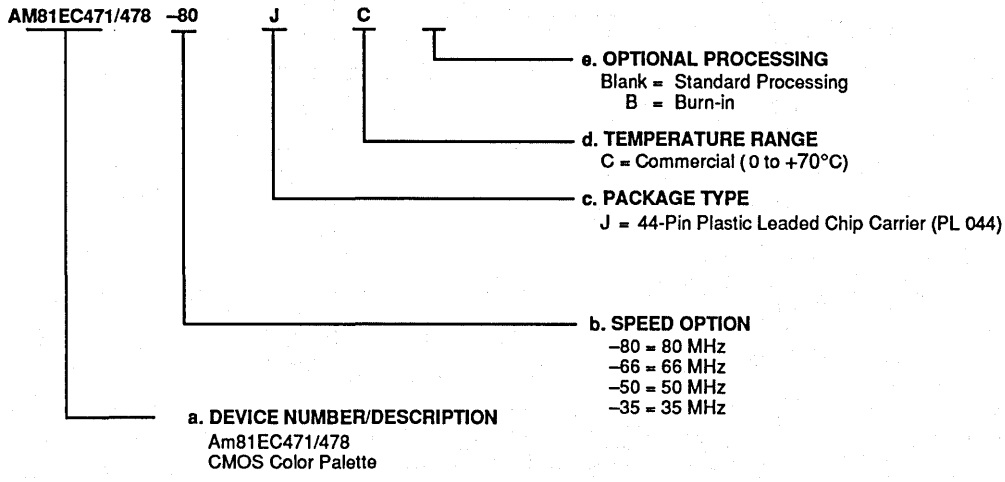
PL 044



ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM81EC471-80	JC, JCB
AM81EC471-66	
AM81EC471-50	
AM81EC471-35	
AM81EC478-80	
AM81EC478-66	
AM81EC478-50	
AM81EC478-35	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Am95C60

Quad Pixel Dataflow Manager

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- Generates mixed text and graphics within Display Memory
- Draws vectors up to 3.3 million pixels per second, or places text at 50,000 characters per second
- One chip handles four Display Memory planes of any size up to 4K x 4K bits and screen sizes to 2K x 2K pixels
- Capable of cascading to handle multiple memory planes without system performance degradation
- Reflects GKS, CGI, and NAPLPS software standards
- Supports windowing, panning, and scrolling
- Supports drawing of anti-aliased vectors, circles, and arcs with various user-definable line styles
- Fills arbitrary polygons
- Supports dual-port video DRAMs
- CMOS technology
- Provides memory and video refresh at user-definable rates
- Interfaces to any 8- or 16-bit system bus
- Comprehensive instruction set

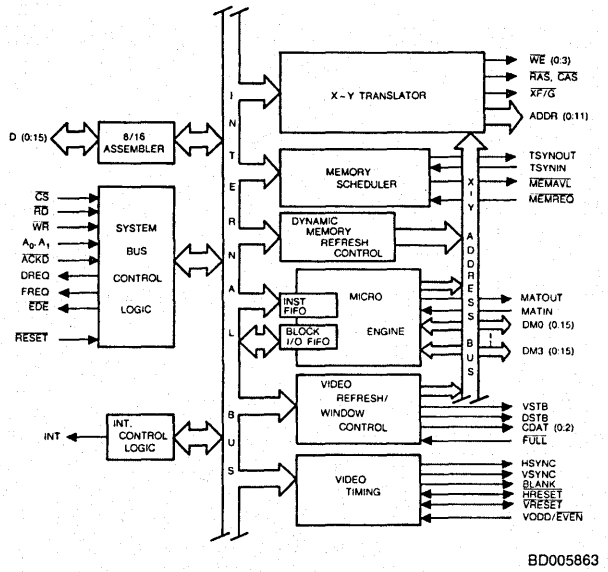
GENERAL DESCRIPTION

The Am95C60 Quad Pixel Dataflow Manager (QPDM) is a CMOS graphics processor which contains the necessary circuitry and control functions for driving four bit-mapped memory arrays. Featuring a maximum system clock speed of 20 MHz, the Am95C60 can interface to any 8- or 16-bit system bus and can draw vectors up to 3.3 million pixels per second, or place text at a rate of 50,000 characters per second. Such performance allows the user to efficiently mix text and graphics within the bit map. The Am95C60 QPDM also contains graphics primitives which smoothly interface with the GKS, CGI, and NAPLPS software standards.

The Am95C60 interfaces directly to memory planes consisting of dual-port video dynamic memories (VRAMs) and is capable of supporting four planes up to 4K by 4K bits and display screens up to 2K by 2K pixels. The Am95C60 is fully cascadable and can manage up to 256 memory planes with no system performance degradation.

The Am95C60 QPDM provides support for the drawing of anti-aliased vectors, circles, and arcs with various user-defined linestyles. Other features include windowing, independent X and Y zoom factors, pan and scroll, picking, clipping, and logical PEL. The Am95C60 is packaged in a 145-lead Pin Grid Array (PGA), and a 160-pin Plastic Quad Flatpack (PQFP).

BLOCK DIAGRAM



BD005863

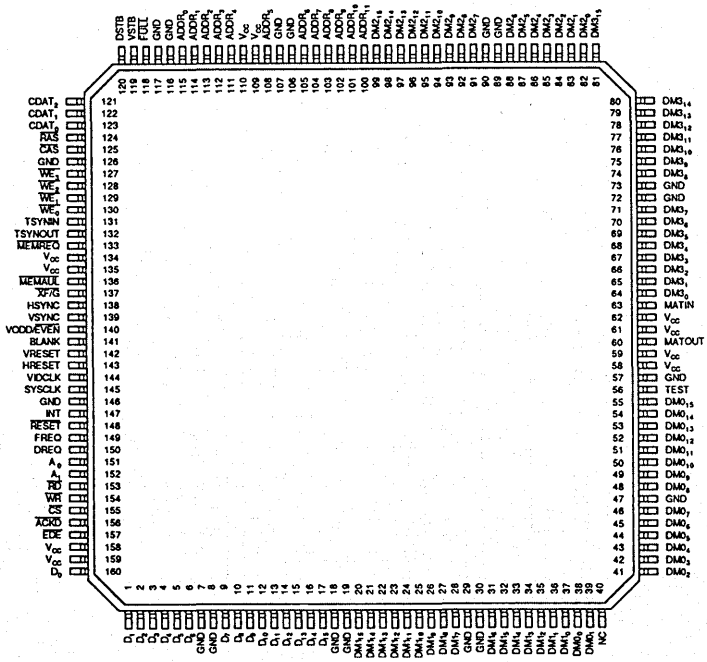
Publication #	Rev.	Amendment
07013	C	/0

Issue Date: October 1988

CONNECTION DIAGRAMS
PGA
(Pins Facing Up)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
1	$DM3_{14}$	$DM3_{13}$	$DM3_{11}$	$DM3_{10}$	$DM3_7$	$DM3_5$	$DM3_3$	$DM3_1$	TEST	$DM0_{12}$	$DM0_{10}$	$DM0_9$	$DM0_7$	$DM0_5$	$DM0_1$	1
2	$DM2_0$	$DM3_{15}$	$DM3_{12}$	$DM3_8$	$DM3_4$	$DM3_2$	$DM3_0$	MATOUT	$DM0_{14}$	$DM0_{13}$	$DM0_{11}$	$DM0_8$	$DM0_2$	$DM0_3$	$DM0_0$	2
3	$DM2_2$	$DM2_4$	$DM2_1$	$DM3_9$	GND	$DM3_6$	V_{CC}	MATIN	V_{CC}	$DM0_{15}$	GND	$DM0_6$	$DM0_4$	$DM1_4$	$DM1_0$	3
4	$DM2_6$	$DM2_7$	$DM2_5$	NC									$DM1_1$	$DM1_3$	$DM1_2$	4
5	$DM2_9$	$DM2_{11}$	$DM2_3$										$DM1_7$	$DM1_6$	$DM1_5$	5
6	$DM2_{13}$	$DM2_8$	GND										GND	$DM1_8$	$DM1_9$	6
7	$DM2_{15}$	$DM2_{12}$	$DM2_{10}$										$DM1_{14}$	$DM1_{10}$	$DM1_{11}$	7
8	$DM2_{14}$	$ADDR_{10}$	$ADDR_{11}$										$DM1_{12}$	$DM1_{13}$	$DM1_{15}$	8
9	$ADDR_8$	$ADDR_9$	GND										GND	D_{14}	D_{15}	9
10	$ADDR_6$	$ADDR_5$	V_{CC}										D_{13}	D_{12}	D_{11}	10
11	$ADDR_4$	$ADDR_7$	$ADDR_1$										GND	D_{10}	D_9	11
12	$ADDR_3$	$ADDR_2$	GND										D_8	D_7	D_6	12
13	$\overline{FULL}$	$ADDR_0$	$\overline{RAS}$	$\overline{WE_0}$	$\overline{CAS}$	$\overline{XF/G}$	V_{CC}	VODD/ $\overline{EVEN}$	DREQ	INT	V_{CC}	$\overline{WR}$	D_5	D_3	D_4	13
14	VSTB	$CDAT_2$	$CDAT_0$	$\overline{WE_1}$	TSYNIN	$\overline{MEMAVL}$	VSYNC	BLANK	$\overline{HRESET}$	SYSCLK	$\overline{RESET}$	$\overline{RD}$	$\overline{ACKD}$	D_1	D_2	14
15	DSTB	$CDAT_1$	$\overline{WE_3}$	$\overline{WE_2}$	TSYNOUT	$\overline{MEMREQ}$	HSYNC	$\overline{VRESET}$	VIDCLK	FREQ	A_0	A_1	$\overline{CS}$	$\overline{EDE}$	D_0	15
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	

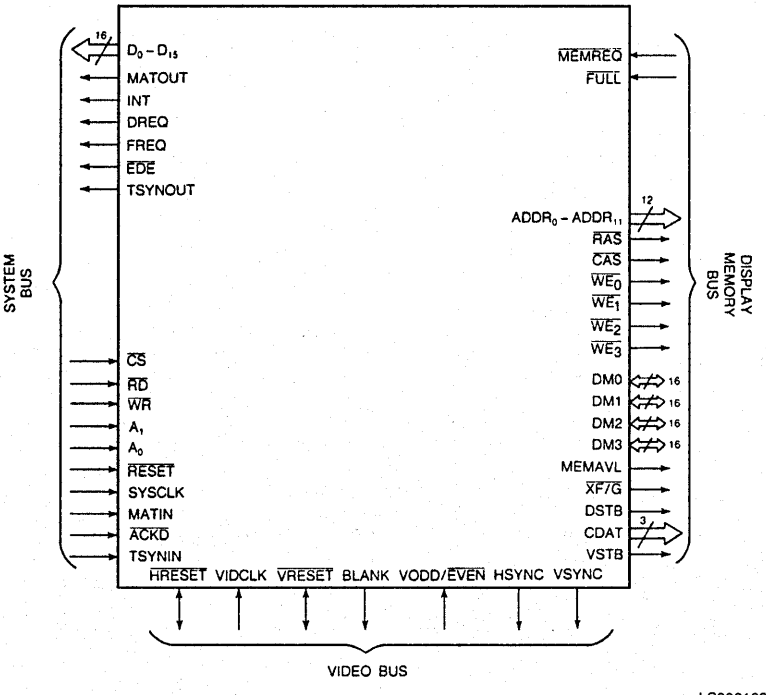
Flatpack
(Top View)



07013-007A

CD011720

LOGIC SYMBOL



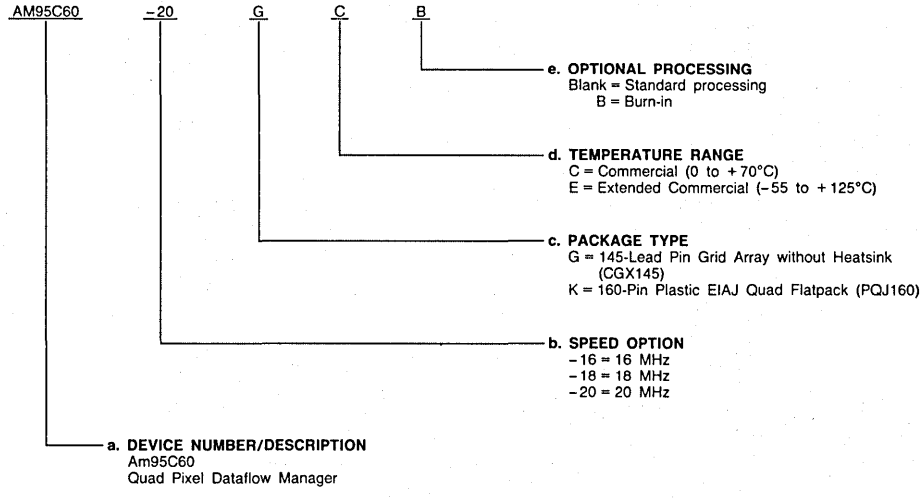
LS002163

3

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C60-16	GC, GCB, GE, GEB, KC
AM95C60-18	
AM95C60-20	

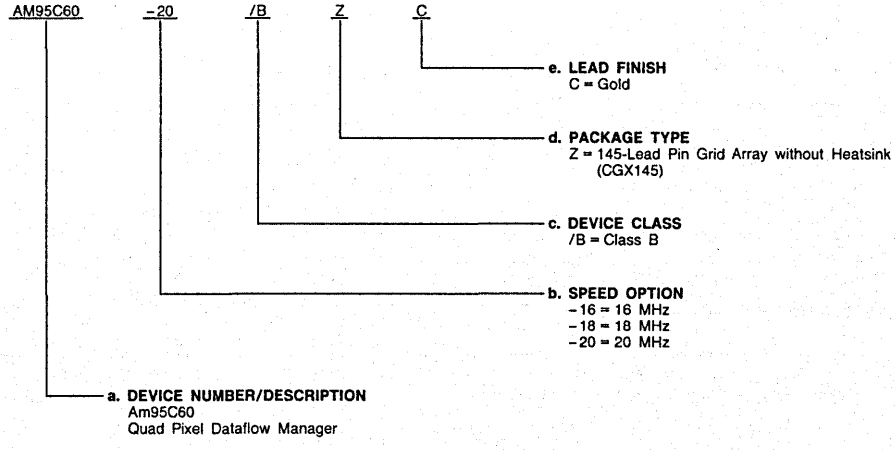
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defence applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM95C60-16	/BZC
AM95C60-18	
AM95C60-20	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

The Am95C60 interfaces through three buses: the System Bus, the Display Memory Bus, and the Video Bus.

System Bus

A_0, A_1 Port Address (Input)
These two inputs are used for selecting the appropriate port to be read or written.

$\overline{ACKD}$ Acknowledge DMA (Input; Active LOW)
The external DMA device may drive this pin LOW in response to a DMA request to strobe in or read out data in fly-by DMA transfer format.

$\overline{CS}$ Chip Select (Input; Active LOW)
Signal used for selecting the Am95C60 from several peripherals connected to the same system bus.

D(0:15) Command/Data/Status (Input/Output; Three-State)
These sixteen lines are used for transferring commands/data/status on the System Bus. The nature of the information transferred on the D(0:15) lines is specified with the port address pins A_0, A_1.

DREQ Data FIFO Request (Output; Open Drain)
Signal used to start and suspend a transfer of data between the System Memory and the Display Memory.

$\overline{EDE}$ External Driver Enable (Output; Active LOW)
This pin is used to enable external data bus drivers on the system bus. $\overline{EDE}$ is inactive (HIGH) during an output block operation on those Am95C60 devices that do not participate in the output. This signal eliminates contention on the system bus.

FREQ Instruction FIFO request (Output; Open Drain)
This signal is used to start and suspend a transfer of instructions from the system memory into the QPDM instruction FIFO.

INT Interrupt (Output; Active HIGH)
High-level interrupt output used to signal that an exception has occurred. The nature of the exception can be determined by reading the Status register.

MATIN Match In (Input, Active HIGH)
This pin is connected to the output of the AND gate connected to the MATOUT outputs.

MATOUT Match Out (Output; Active HIGH)
This pin is used in multiple-QPDM applications to search for a matching color pattern. As long as the pattern is not found, this pin stays LOW. When the matching pattern is found the pin is driven HIGH. Since all the MATOUT outputs are logically ANDed externally, a match in a multiple-QPDM environment is visible on MATIN when all the MATOUT outputs are HIGH. This pin is also used for instruction-execution synchronization by re-aligning the Am95C60 devices in a system at the beginning of each instruction execution and at the beginning of each word transfer in Block I/O instruction.

$\overline{RD}$ Read (Input; Active LOW)
Signal used for reading information (data/status) from the Am95C60 QPDM on the D(0:15) lines by a bus master.

$\overline{RESET}$ System Reset (Input; Active LOW)
The $\overline{RESET}$ signal brings all Am95C60s in the system to the same initial state. All the outputs are brought into the inactive state. If $\overline{RESET}$ is activated during the time when the Am95C60 was active, all the activities will be suspended.

SYSCLK System Clock (Input)
20 MHz maximum frequency clock. Controls the Am95C60 QPDM internal timing except for video timing.

TSYNIN Timing Synchronization (Input; Active HIGH)
All TSYNIN input pins are connected to the AND of the TSYNOUT.

TSYNOUT Timing Synchronization (Output, Active HIGH)
In conjunction with TSYNIN, this pin is used to synchronize Display Memory Bus activities. The TSYNOUT pins of all Am95C60 devices in a system are ANDed together and connected to the TSYNIN input pins of all Am95C60s. User-transparent information signals all Am95C60 devices about Display Memory Bus activities.

$\overline{WR}$ Write (Input; Active LOW)
Signal used for strobing information (commands/data) into the Am95C60 from the D(0:15) lines.

Display Memory Bus

ADDR(0:11) Address (Output)
The twelve lines of address are used for addressing bit-map planes each up to 4K x 4K bits. The addresses are multiplexed and contain row and column addresses and bank-select bits.

$\overline{CAS}$ Column Address Strobe (Output; Active LOW)
This line is used to strobe the column address from the multiplexed ADDR lines into the array.

DM0-3(0:15) Display Memory Bus (Input/Output)
These 64 lines are used for transferring data between the Am95C60 and the Display Memory. There are sixteen lines for each of the four planes.

$\overline{MEMAVL}$ Memory Bus Available (Output; Active LOW)
This line is used to inform external devices that are requesting the bus that the Am95C60 is not driving the data lines.

$\overline{MEMREQ}$ Memory Request (Input; Active LOW)
This asynchronous signal is used by an external device to request access to the Display Memory Bus.

$\overline{RAS}$ Row Address Strobe (Output; Active LOW)
This line is used to strobe the Row address from the multiplexed ADDR lines into the array.

$\overline{WE}$(0:3) Write Enable (Output; Active LOW)
The $\overline{Write\ Enable}$, when active, signifies that the current transaction on the Display Memory Bus is a write to the corresponding bit plane.

$\overline{XF/G}$ Transfer/Output Enable (Output; Active LOW)
This pin interfaces directly to the video DRAM. During a transfer cycle this pin indicates a transfer to the video DRAMs. During a random read cycle, this pin enables the output buffer of the video DRAMs.

Video Control Bus

BLANK Blank Video (Output; Active HIGH)
BLANK is an active HIGH output which serves to blank out inactive display areas of the CRT. This output is held HIGH when the Am95C60 is reset.

CDAT(0:2) Control Data (Output; Active HIGH)
These lines are used to output 3 bits of information to the VDAF. Information is sent to the VDAF during the transfer cycle and at VSTB time.

DSTB Data Strobe (Output; Active HIGH)
This clock signal loads 8 bits of video data into the VDAF. DSTB is synchronous with SYSCLK and has twice the frequency of VSTB.

$\overline{\text{FULL}}$ $\overline{\text{Full}}$ (Input; Active LOW)
This input alerts the Am95C60 that the VDAF cannot accept more video data. If the $\overline{\text{FULL}}$ signal is active, the Am95C60 stops generating the VSTB and DSTB signals.

$\overline{\text{HRESET}}$ $\overline{\text{Horizontal Reset}}$ (Input/Output; Active LOW)
This pin is an output for horizontal video masters, and an input for horizontal video slaves. It is used for vertical video synchronization to other Am95C60s or to an external video source.

HSYNC Horizontal Sync (Output; Active HIGH)
HSYNC is an active HIGH output which causes horizontal retrace of the CRT's electron beam. This output is held LOW when the Am95C60 is reset to prevent any uncontrolled synchronization to the CRT, which may cause damage to the tube.

VIDCLK Video Clock (Input)
15 MHz maximum frequency clock used for generating video synchronization signals.

VODD/$\overline{\text{EVEN}}$ Vertical Odd/$\overline{\text{Even}}$ (Input)
This input is optionally used in the interlaced Display mode to distinguish between the even frame and the odd frame specified by an external device.

VSTB Video Strobe (Output, Active HIGH)
This strobe signal is used in a system with video dynamic RAMs (VRAMs) and an external VDAF (Am8171/8172 Video Data Assembly FIFO or a similar circuit built of discrete components) to shift video data out of the Video Memory. With every strobe, a 16-bit-wide word is shifted out of the Video Memory.

VSYNC Vertical Sync (Output; Active HIGH)
VSYNC is an active HIGH output which causes vertical retrace of the CRT's electron beam. This output is held LOW when the Am95C60 is reset.

$\overline{\text{VRESET}}$ $\overline{\text{Vertical Reset}}$ (Input/Output; Active LOW)
This pin is an output for vertical video masters and an input for vertical video slaves. It is used for vertical video synchronization to other Am95C60 devices or to an external video source.

Power Connections

GND Ground
Each GND pin must be connected to the power supply ground.

TEST Test
This pin may be grounded or it may be pulled up to V_{CC}; it must not be allowed to float. This pin for factory test purposes only.

V_{CC} Power Supply
Each V_{CC} must be connected to a +5-V power supply.

FUNCTIONAL DESCRIPTION

The Am95C60 Quad Pixel Dataflow Manager is a graphics processor which maintains, updates, and displays information in four bit-mapped video planes. As depicted in Figure 1, the System Interface communicates with either an 8- or 16-bit host CPU while the Display Memory Interface controls four bit-mapped memory planes. These planes consist of dual-port video dynamic memories (VRAMs). The Am95C60 connects to the random ports while the serial ports are used to access video information for the screen refresh. When used in conjuction with the Am8172 Video Data Assembly FIFO, the Am95C60 is capable of displaying a hardware window with the associated features of smooth pan and soft scroll.

The Am95C60 QPDM performs three fundamental functions described below:

Video Refresh

The Am95C60 QPDM manages the screen (display) refresh function by generating the addresses to the bit-map VRAMs as required to access data for display on the screen. The data from the VRAMs are serialized externally to the Am95C60.

The Video Refresh operation is fully programmable, allowing the user to tailor the system as required. The screen display can be aligned on any pixel boundary and can also include one hardware window overlay. Additionally, the total video process can be externally synchronized to any external source at the horizontal or vertical synchronization rate. Video Refresh can be disabled for operation as a slave device.

Dynamic Memory Refresh

The Am95C60 performs the Dynamic Memory Refresh function for the Display Memory. The Dynamic Memory Refresh process is interleaved with the Video Refresh and with the Display Memory updating. The refresh rate is programmed by loading the 10-bit Dynamic Memory Refresh Rate register. This register holds the number of SYSCLK cycles between two refresh cycles.

The Dynamic Memory Refresh Rate register is loaded into a counter, which is down-counted by the SYSCLK. When the counter reaches its zero state, it sends a refresh request to the Memory Scheduler. As soon as the Memory Scheduler arbitrates a time slice for the Dynamic Memory Refresh block, a refresh cycle is initiated.

The Am95C60 executes $\overline{\text{CAS}}$ before $\overline{\text{RAS}}$ refresh cycles.

Display Memory Update

The Am95C60 has access to the Display Memory Bus for updating purposes, except when it is performing Video or Dynamic Memory Refresh.

Placing an instruction in the execution FIFO (see Block Diagram) signals the Micro Engine to begin operation. If the execution of the instruction requires access to the Display Memory Bus (this is the case in most instructions), the Am95C60 continues the instruction execution unless the Display Memory Bus is occupied by the Video Refresh or the Dynamic Memory Refresh process. If the Display Memory Bus cannot be accessed momentarily, the execution is suspended until the Display Memory is available.

The drawing instruction set includes Line, Circle, Filled Triangle, String and many others which will be briefly described in the following pages.

Feature Description

Window Display Mechanism

The Am95C60 QPDM, in conjunction with the Am8172 Video Data Assembly FIFO (VDAF) or the equivalent, can support a single non-destructive hardware window. The image to appear in the window is located in some other area of Display Memory than that visible on the screen (this is shown in Figure 2). The size and position of the window is programmed into a set of registers on the Am95C60. Since the window position is dynamically programmable, it is easy to 'drag' a rectangular area containing an object. It is also easy to perform soft scrolling and smooth panning of either background or foreground.

Clipping

The clipping feature on the Am95C60 allows a rectangular region to be defined outside of which vectors and arcs will not be drawn, blocks will not be moved or modified, and polygons will not be filled. The clipping window is specified by the user, and remains in effect until changed or disabled.

Picking

If a drawing consists of a large number of objects and each object is defined by a number of drawing primitives, any object can be identified by the following picking process.

First, the picking area is defined as a rectangular region in Display Memory. Whenever a drawing intersects the picking area, a 'Pick Detect' bit in the status register is set. Objects to be displayed are labeled by using the Signal instruction which will return the label of the object that intersected the picking area.

Multiple Am95C60 Operation

In order to accomodate systems requiring access to more than four bit-planes, the Am95C60 is designed to be fully cascadable with no performance degradation. Multiple Am95C60 devices can communicate with each other to share timing information for synchronization purposes and status information for color comparisons in depth.

Instructions are broadcast to all Am95C60 devices at once. Each plane will use the instruction in conjunction with its own activity and color bits to decide whether to execute or to ignore the instruction. Each plane may execute the instruction differently, depending on the contents of the individual memory plane and the contents of plane-specific status information.

Broadcasting is accomplished by chip-selecting all Am95C60 QPDMs simultaneously and writing to Port 0. (Write to the instruction FIFO.)

TSYNOUT and TSYNIN are used to synchronize Display Memory operations in a multiple Am95C60 system.

MATOUT and MATIN are used to exchange color searching information in a multiple Am95C60 device system and to synchronize instruction execution.

Interface Description

System Bus Interface

The host connects to the system side of the Am95C60 as depicted in the Logic Symbol diagram. There is a 16-bit data path, and the 8- or 16-bit option allows the Am95C60 QPDM to be connected to an 8- or 16-bit host processor.

The normal bus interface control lines are supported through $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, and two address bits. The address bits are decoded to select one of four ports:

A_1	A_0	WRITE FUNCTION	READ FUNCTION
0	0	Write Instruction FIFO	Read Status
0	1	Write Block Input FIFO	Read Block Output FIFO
1	0	Write Register Addresss	Read Register Address
1	1	Write Register	Read Register

The Am95C60 QPDM can be supported with a DMA controller, allowing blocks of information or instructions to be transferred without tying up the host. In addition to the normal flow-through DMA operations, the Am95C60 also supports fly-by operations.

The Am95C60 also can use interrupts to signal the occurrence of certain events. Some events are repetitive (e.g., Frame), some indicate error conditions (e.g., Stack Overflow), and some merely report status (e.g., Idle). There are registers for masking interrupts, reading interrupt requests, and acknowledging interrupts.

Display Memory Bus Interface

On the Display Memory side, the Am95C60 is capable of controlling four bit planes. Addresses, $\overline{RAS}$, and $\overline{CAS}$ are common signals to all the bit planes while each plane has its own set of data lines and write enable ($\overline{WE}$). Typically, eight or nine address lines (multiplexed Row/Column) go to the VRAM devices while the others may be used for bank select. If multiple banks of memory are used for each bit plane, row addresses must be decoded to select the proper bank.

Each bit plane has a 16-bit data bus used for the Display Memory Update function. Typically, to write a single pixel (one bit in each plane), the Am95C60 would perform simultaneous 16-bit reads from all planes, followed by simultaneous 16-bit writes to all planes. Logic is also provided in the Am95C60 to perform individual pixel writes.

The Am95C60 is intended to be used with a variety of dual-ported Video RAMS (VRAMs). Prior to the beginning of each scan line, the Am95C60 executes a transfer cycle which copies the contents of the scan line into shift registers on the VRAM devices. The scan line image is then shifted out of the VRAM devices 16 bits at a time and further serialized at the dot clock rate for display purposes. The primary VRAM port is available during this time for Display Memory Update.

Display Memory/RAM Size Examples

BIT MAP SIZE	VRAM SIZE	VRAMS/ PLANE	VRAMS/ Am95C60
1024 x 1024	64K x 4	4	16
1024 x 2048	64K x 4	8	32
2048 x 2048	64K x 4	16	64
4096 x 4096	256K x 4	16	64

MEMREQ and MEMAVL are used to allow another processor direct access to the Display Memory.

Video Bus Interface

The Am95C60 generates video timing. Horizontal timing is programmed in terms of VIDCLK cycles and vertical timing is programmed in terms of ½ scan lines.

$\overline{HRESET}$ and $\overline{VRESET}$ are either inputs or outputs depending on the Am95C60's master/slave status. In the case where multiple Am95C60s control the display, one Am95C60 would be programmed as the timing master and the others would be programmed as timing slaves.

Power Connections

The Am95C60 uses +5 volts only.

Performance Figures

Working with VRAMs, the Am95C60 can easily manage high resolution screen formats requiring dot clock rates up to 160 MHz. Moreover, it performs the bit-map update function as indicated in Table 1.

TABLE 1. PERFORMANCE FIGURES OF Am95C60

Instruction	Instruction Overhead	Intermediate Overhead	Execution Time	Comments
Line	12.9 μs	(Not Applicable)	300 nx/pixel	
Line	12.9 μs	(Not Applicable)	4750 ns/pixel	Anti-Aliased
Polyline	10.6 μs	4.8 μs/segment	300 ns/pixel	Connected Segments
Arc	28.2 μs	2.7 μs/octant	750 ns/pixel	
Arc	28.2 μs	2.7 μs/octant	4750 ns/pixel	Anti-Aliased
Circle	9.9 μs	2.7 μs/octant	750 ns/pixel	
Circle	9.9 μs	2.7 μs/octant	4750 ns/pixel	Anti-Aliased
Copy Block	10.9 μs	1.8 μs/scan line	59 ns/pixel	BITBLT
Transform Block	11.0 μs	(Included)	1280 ns/pixel	3X Zoom
Seed Fill	10.0 μs	12.1 μs/scan line	280 ns/pixel	Intermediate Overhead varies with shape
Filled Rectangle	11.9 μs	2.2 μs/scan line	19 ns/pixel	Graphical SET
Filled Triangle	54.9 μs	8.0 μs/scan line	19 ns/pixel	Intermediate Overhead varies with shape
String	6.3 μs	9.4 μs/character	2000 ns/scan line	

The Am95C60 provides extremely fast access at the host interface. A host write requires as little as 80 ns, while a register may be read in as little as 120 ns.

Consult the technical manual (Order No. 07785) to see how these performance measurements are determined.

Register Description

The Am95C60 QPDM contains a number of registers which are programmable from the host. These registers are listed below:

Video Control uses nine registers which define video operation parameters:

- Horizontal Sync Pulse Width (HSYNC)
- Horizontal Scan Delay (HDEL)
- Horizontal Active (HACT)
- Horizontal Total Count (HTOT)
- Vertical Sync Pulse Width (VSYNC)
- Vertical Scan Delay Odd (VDELODD)
- Vertical Scan Delay Even (VDELEVEN)
- Vertical Active Lines (VACT)
- Vertical Total Lines (VTOT)

Visible Screen Coordinates use four registers which contain the (x,y) address in real memory of the top-left and bottom-right corner of the visible screen in Display Memory:

- Screen X Start
- Screen Y Start
- Screen X Terminate
- Screen Y Terminate

Window Control uses six registers which specify where the window is on the screen (the apparent window) and where it begins in memory (the real window):

- Window Apparent X Start
- Window Apparent Y Start
- Window Apparent X Terminate
- Window Apparent Y Terminate
- Window Real X
- Window Real Y

Video Mode Register is used to indicate to each Am95C60 whether it is a timing master or slave, and whether or not interlaced display mode is to be used.

Memory Mode Register specifies Display Memory configuration in terms of memory width and device size (e.g., 4K wide/256K devices).

Dynamic Memory Refresh Rate Register specifies the number of SYSCLK clock cycles between row refreshes in Dynamic Memory Refresh.

Interrupt Enable Register indicates which conditions are allowed to cause interrupts to the host.

Interrupt Acknowledge Register indicates that a specific interrupt condition is known to the host and that the request is to be cancelled in the Am95C60.

Video Timing Enable is a 1-bit register used to enable and disable video sync and output.

Video Refresh Enable is a 1-bit register used to enable and disable the collection of video information from the Display RAM.

System Bus Width Register configures the Am95C60 to 8-bit system bus modes.

Instruction Set

The Am95C60 QPDM is a graphics processor with a powerful instruction set oriented toward graphics processing. One may use any of four addressing modes to specify locations in the display memory: 1) Absolute, 2) Relative, 3) Viewpoint, or 4) Indirect. Each member of the Am95C60 QPDM instruction set is briefly described as follows:

Arc draws the image of a circular arc in Display Memory. The parameters are the center of the arc, the radius of the arc, and the two end-points. The image may be drawn using anti-aliasing, line style, and a logical PEL.

Arc Current draws the image of a circular arc in Display Memory. It is similar to Arc except the start point is taken to be the current pen position (rather than being specified).

Call begins fetching instructions from Display Memory rather than from the instruction FIFO. The parameters specify the

location of the program to be executed. A stack in Display Memory is used to contain the return location.

Circle draws the image of a circle in Display Memory. The parameters are the circle's center and radius. The image may be drawn using anti-aliasing, line style, and a logical PEL.

Circle Current draws the image of a circle in Display Memory. It is similar to Circle, except the center is taken to be the current pen position.

Control Clipping enables or disables the clipping function. When clipping is enabled, all drawing primitives will change only that portion of Display Memory which lies within the rectangular clipping region.

Control Picking enables or disables the picking function. When picking is enabled, drawing primitives will not execute any writes to the Display Memory. The Pick Detect status bit is set whenever a drawing primitive intersects the picking area.

Copy Block moves a block of data within Display Memory and may optionally combine the source image with the destination image. The size of the block will have been determined by a Set Block Size instruction. The location of each block, source and destination, is determined by the instruction.

Copy Block Current is identical to the Copy Block instruction except that the source operand is the current pen position.

Define Logical PEL specifies the logical PEL (pen size and content) used by the drawing primitives. This can be used to draw thick lines.

Fill Bounded Region fills an arbitrary polygon with a specific color. The polygon is defined as the group of dots completely contained within a boundary of pixels of the edge color. All pixels will be changed to the current drawing color. The location of the seed is specified in the instruction.

Fill Bounded Region Current fills an arbitrary polygon with a specific color. The polygon is defined as the group of dots completely contained within a boundary of pixels of the edge color. All pixels in the region will be changed to the current drawing color. The location of the seed is the current pen position.

Fill Connected Region fills an arbitrary polygon with a specific color. The polygon is defined as any group of connected dots of the seed's color. All pixels connected to the seed point having the same color will be changed to the current drawing color. The location of the seed is specified in the instruction.

Fill Connected Region Current fills an arbitrary polygon with a specific color. The polygon is defined as any group of connected dots of the seed's color. All pixels connected to the seed point having the same color will be changed to the current drawing color. The location of the seed is the current pen position.

Filled Rectangle creates the image of a rectangle and fills it. The parameters specify two opposite corners of the rectangle. The color of the filled rectangle is the current drawing color.

Filled Rectangle Current is similar to Fill Rectangle except the current pen position is taken to be the starting corner of the rectangle. The other corner is specified by the instruction.

Filled Triangle creates the image of a triangle and fills it. The parameters specify the three vertices of the triangle. The color of the filled triangle is the current drawing color.

Filled Triangle Current creates the image of a triangle and fills it. The parameters specify two vertices of the triangle. The current pen position is taken to be the third vertex of the triangle. The color of the filled triangle is the current drawing color.

Input Block transfers a rectangular block of data from the host to Display Memory. The size of the block will have been determined by a Set Block Size instruction. The destination address in the Display Memory is specified in the instruction. The data to be stored in Display Memory is written into the Data FIFO.

Input Block Current transfers a rectangular block of data from the host to Display Memory. The size of the block will have been determined by a Set Block Size instruction. The destination address in the Display Memory is the current pen position. The data to be stored in Display Memory is written into the Data FIFO.

Jump unconditionally changes the location counter when executing instructions from Display Memory.

Line draws the image of a line in Display Memory. The input parameters are the two ends of the line. The image may be drawn using anti-aliasing, line style, and a logical PEL. Multiple lines may be drawn with a single line instruction.

Line Current is similar to Line, except the current drawing position is taken to be the starting point of the line.

Line Reversible draws the image of a line in Display Memory. It is exactly the same as Line except that the standard algorithm is modified to guarantee that the same set of pixels is chosen regardless of the order of the end points.

Line Reversible Current is similar to Line Reversible except that the current pen position is taken to be the starting point of the line.

Move Pen sets the current pen position.

No Operation ensures that no operation is performed.

Output Block transfers a rectangular block of data from Display Memory to the host. The size of the block will have been determined by a Set Block Size instruction. The source address in the Display Memory is specified in the instruction. The host is expected to remove the data from the Data FIFO.

Output Block Current transfers a rectangular block of data from Display Memory to the host. The size of the block will have been determined by a Set Block Size instruction. The source address in the Display Memory is the current pen position. The host is expected to remove the data from the Data FIFO.

Point draws the image of the current logical PEL at the location specified in the instruction.

Point Current draws the image of the current logical PEL at the current pen position.

Return exits from a subroutine or from program mode when executing instructions from Display Memory.

Set Activity Bits indicates which of the four Display Memory planes, controlled by the Am95C60, are to be written into.

Set Anti-Aliasing Distance programs the anti-aliasing distance deviation from the ideal line.

Set Block Size specifies the number of pixels moved in any block operation. This is used for Input Block, Output Block, Copy Block, Transform Block, and the logical PEL.

Set Character Font Base specifies the character font address in the Display Memory. The character font contains the patterns of letters and numbers used by the String instruction.

Set Character Font Base Current specifies the character font addresses in Display Memory.

Set Clipping Boundary specifies where the clipping region is in Display Memory. When clipping is enabled, all drawing primitives will change only that portion of Display Memory which

lies within the rectangular clipping region. The parameters are the addresses of two opposite corners of the clipping rectangle.

Set Clipping Boundary Current specifies where the clipping region is in Display Memory. When clipping is enabled, all drawing primitives will change only that portion of Display Memory which lies within the rectangular clipping region. The current pen position is taken to be the start corner of the clipping rectangle.

Set Color Bits sets the current drawing color.

Set Search Color specifies the edge color used in Fill instructions.

Set Line Style specifies the line style. This defines the dash length, the interspace length, and the dot length.

Set Line Style Phase indicates where the line begins within the line style cycle.

Set Listen Bits indicates which planes take part in polygon and color change operations. If set, the corresponding plane does not participate in the color matching.

Set Picking Region specifies the rectangular area to be picked. The parameters are two opposite corners of the picking rectangle. When picking is enabled, drawing primitives that intersect the picking region will cause the "Pick Detect" bit in the status register to be set and no writes will be executed to Display Memory.

Set Picking Region Current specifies the rectangular area to be picked. The parameters are two opposite corners of the picking rectangle. The current pen position is taken to be the start corner of the picking region. When picking is enabled, drawing primitives that intersect the picking region will cause the "Pick Detect" bit in the status register to be set and no writes will be executed to Display Memory.

Set QPDM Position specifies the logical addresses for each Display Memory plane.

Set Scale Factor provides values used to multiply the operands of instructions which address the bit map.

Set Search Color specifies the color of the boundary for Fill Boundary Area operations.

Set Stack Boundaries specifies to the Am95C60 which area of Display Memory has been set aside for the stack. Stack overflow is detected and signaled to the host with an interrupt.

Set Viewpoint Location specifies the base address for Viewpoint Addressing Mode.

Signal is used to indicate to the host when a particular point in the instruction stream has been reached, to delimit objects during picking, or to pause operation pending a signal from the host.

Store Immediate deposits a specified number of 16-bit words in the Display Memory.

Store Immediate Current deposits a specified number of 16-bit words in Display Memory beginning at the current pen position.

String is used to create the image of a string of text in the Display Memory. The parameters are the address at which the string should begin, followed by a variable length list of 16-bit pointers. Each pointer is used to look up a pattern in the character font table.

String Current is similar to String except the address at which the string should begin is set to the current pen position.

Transform Block allows a block of data to be taken from Display Memory, operated on, and written to a different area of Display Memory. The operations which may be performed are Rotate (90 degree increments), Zoom (by pixel replication), and Mirror. The Zoom in X, and Zoom in Y are independently specified. The size of the source block (prior to rotation and zooming) will have been specified by a Set Block Size instruction.

Transform Block Current is similar to Transform Block except the source operand is at the current pen position.

APPLICATIONS

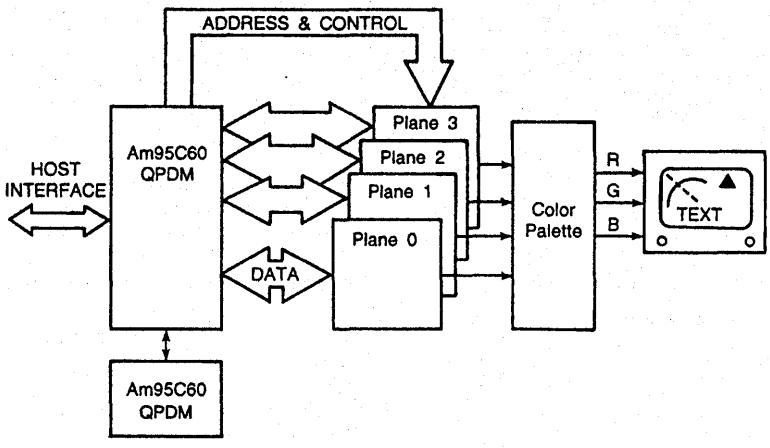


Figure 1. Typical System Application

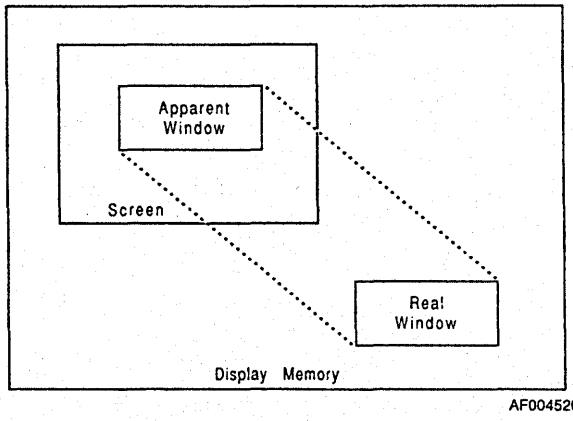


Figure 2. Windows

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Operating Temperature ..... −55 to +125°C
Maximum V_{CC} Relative to V_{SS} ..... −0.3 to +7.0 V
DC Voltage Applied to Any
Pin Relative to V_{SS} ..... −0.5 to V_{CC} +0.3 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V

Military (M) and Extended-Commercial (E) Devices
Case Temperature (T_C) ..... −55 to +125°C
Supply Voltage (V_{CC}) ..... +4.50 to 5.50 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified (for APL products, Group A, Subgroups 1, 2, 3, 7, 8 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions		Min.	Max.	Unit
V_{IL}	Input LOW Voltage			−0.3	+0.8	V
V_{IH}	Input HIGH Voltage		COM'L	+2.0	V_{CC} + 0.3	V
			MIL & E-COM'L	+2.2		
V_{OL}	Output LOW Voltage COM'L Devices CAS and $\overline{XF/G}$	I_{OL} = 4.0 mA			0.4	V
	Output LOW Voltage Commercial Devices All Other Outputs	I_{OL} = 2.0 mA			0.4	
	Output LOW Voltage Military Devices CAS and $\overline{XF/G}$	I_{OL} = 4.0 mA			0.45	
	Output LOW Voltage Military Devices All Other Outputs	I_{OL} = 2.0 mA			0.45	
V_{OH}	Output HIGH Voltage	I_{OH} = 250 µA		2.4		V
I_{OZ}	Output Leakage Current	0.4 < V_{OUT} < V_{CC}			±10	µA
I_I	Input Current	0.4 < V_{IN} < V_{CC}			±10	µA
I_{CC}	Power Supply Current (Note 1)				250	mA

PRELIMINARY

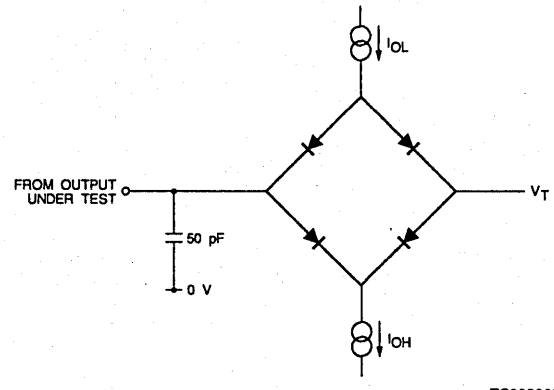
CAPACITANCE*

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
C_{IN}	Input Capacitance			15	pF
$C_{I/O}$	I/O Pin Capacitance			25	pF
C_{OUT}	Output Pin Capacitance			25	pF

*Parameters are not 100% "Tested." Characterization data on file.

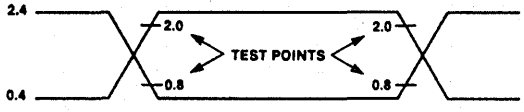
Notes: 1. For operation with SYSCLK less than 20 MHz, deduct approximately 6.5 mA/MHz.

SWITCHING TEST CIRCUIT

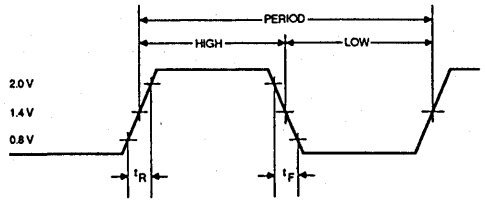


TC003860

SWITCHING TEST WAVEFORMS



WF009541



WF025740

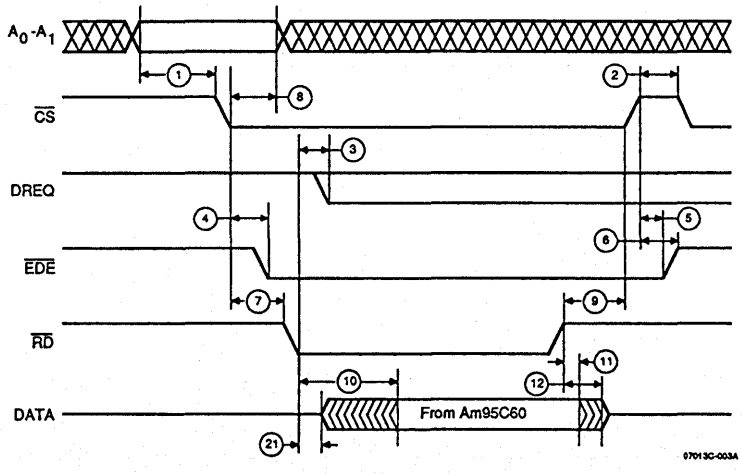
VIDCLK/SYSCLK Waveform

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

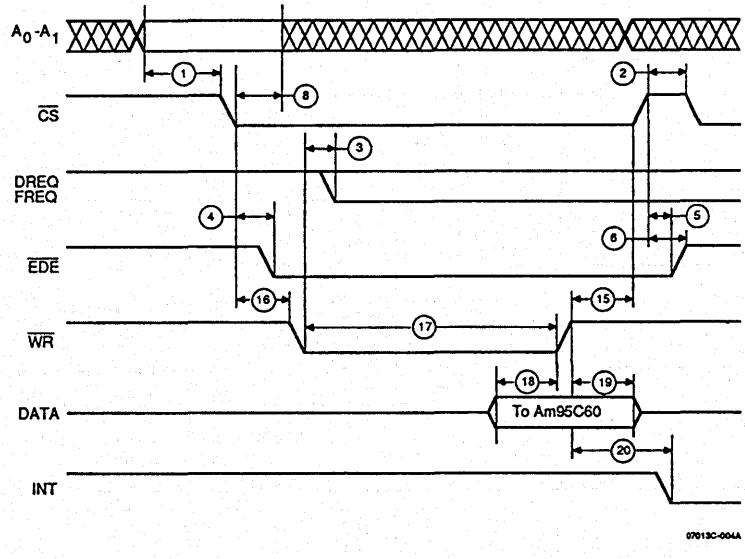
KS000010

SWITCHING WAVEFORMS/CHARACTERISTICS



WF026510

Processor/DMA Flow-Through Read Cycle

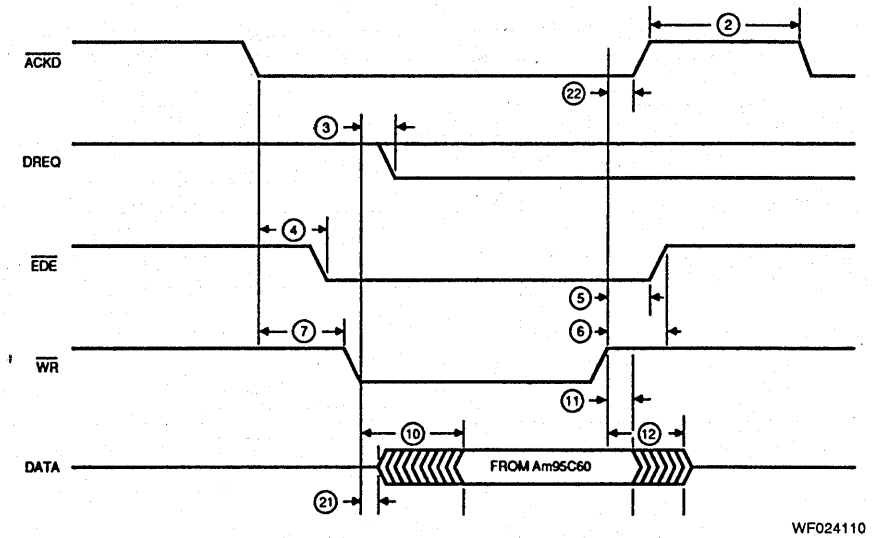


WF026520

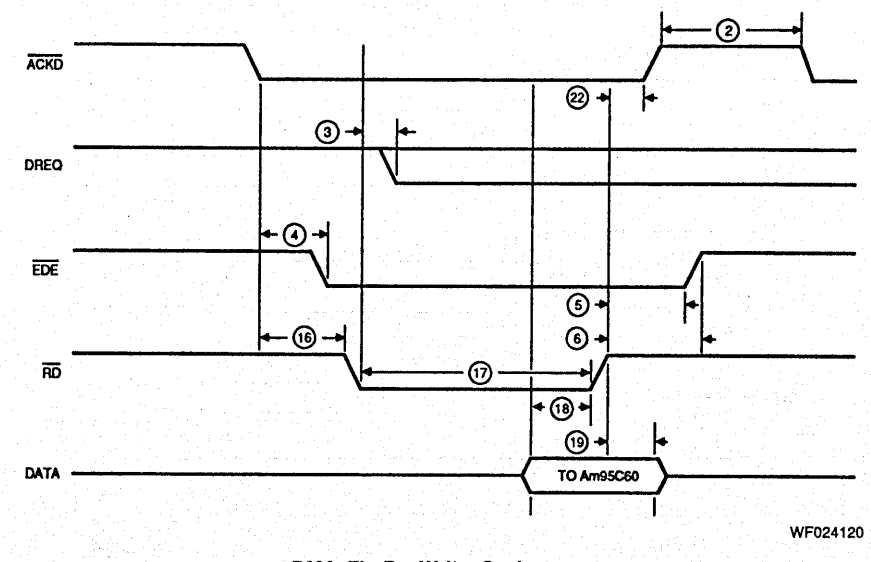
Processor/DMA Flow-Through Write Cycle

3

SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)



DMA Fly-By Read Cycle



DMA Fly-By Write Cycle

SWITCHING WAVEFORMS/CHARACTERISTICS over operating ranges unless otherwise specified (for APL products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

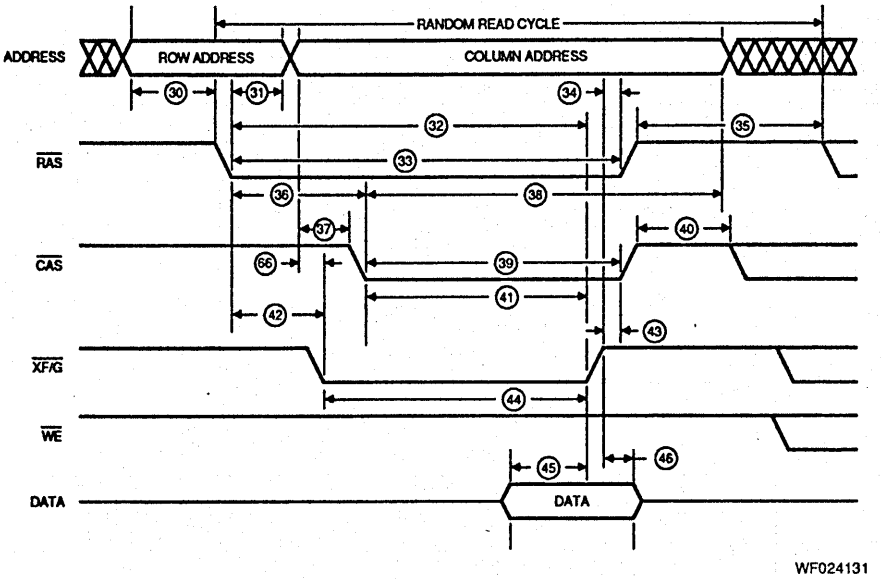
System Bus Timing

No.	Parameter Symbol	Parameter Description	-20 Min.	-20 Max.	-18 Min.	-18 Max.	-16 Min.	-16 Max.	Unit
1	t_S	Requires that the address be valid a minimum of [] ns before $\overline{CS}$ begins to fall.	0		0		0		ns
2	t_W (Note 1)	Requires that both $\overline{CS}$ and $\overline{ACKD}$ be inactive a minimum of [] ns before either can go active.	65		65		95		ns
3	t_{PHL}	Guarantees that DREQ will become inactive a maximum of [] ns after $\overline{RD}$ or $\overline{WR}$ becomes active.		55		55		60	ns
4	t_{PHL} (Note 1)	Guarantees that $\overline{EDE}$ will become active a maximum of [] ns after $\overline{CS}$ becomes active. Also guarantees that $\overline{EDE}$ will become active a maximum of [] ns after $\overline{ACKD}$ becomes active.		60		60		60	ns
5	t_{PLH} (Notes 1, 3)	Guarantees that $\overline{EDE}$ will remain active a minimum of [] ns after $\overline{CS}$ becomes inactive. Also guarantees that $\overline{EDE}$ will remain active a minimum of [] ns after $\overline{RD}$ has become inactive.	10		10		10		ns
6	t_{PLH} (Notes 1, 3)	Guarantees that $\overline{EDE}$ will have gone inactive no more than [] ns after $\overline{CS}$ has become inactive. Also guarantees that $\overline{EDE}$ will have gone inactive no more than [] ns after $\overline{RD}$ has become inactive.		65		65		70	ns
7	t_S (Notes 1, 2)	Requires that $\overline{CS}$ be valid a minimum of [] ns before $\overline{RD}$ can begin to go active. Also requires that $\overline{ACKD}$ be valid a minimum of [] before $\overline{WR}$ can begin to go active.	0		0		0		ns
8	t_H	Requires that the address remain valid a minimum of [] ns after $\overline{CS}$ has gone active.	20		20		20		ns
9	t_H	Requires that $\overline{CS}$ remain active a minimum of [] ns after $\overline{RD}$ has gone inactive.	0		0		0		ns
10	t_{PD} (Note 2)	Guarantees that the read data will be valid within [] ns of $\overline{RD}$ becoming active. Also guarantees that the read data will be valid within [] ns of $\overline{WR}$ becoming active in a Fly-By Read Cycle.		110		110		110	ns
11	t_H (Notes 2, 3)	Guarantees that the read data will remain valid a minimum of [] ns after $\overline{RD}$ has gone inactive. Also guarantees that the read data will remain valid a minimum of [] ns after $\overline{WR}$ has gone inactive in a Fly-By Read Cycle.	10		10		10		ns
12	t_{PD} (Notes 2, 3)	Guarantees that the read buffers will begin to enter high impedance within [] ns of $\overline{RD}$ having gone inactive. Also guarantees that the read buffers will begin to enter high impedance within [] ns of $\overline{WR}$ having gone inactive in a Fly-By Read Cycle.		40		40		40	ns
13	(Not used)	Not Used							
14	(Not used)	Not Used							
15	t_H	Requires that $\overline{CS}$ remain active a minimum of [] ns after $\overline{WR}$ has gone inactive.	10		10		20		ns
16	t_S (Notes 1, 2)	Requires that $\overline{CS}$ be active a minimum of [] ns before $\overline{WR}$ can begin to go active. Also requires that $\overline{ACKD}$ be active a minimum of [] ns before $\overline{RD}$ can begin to go active in a Fly-By Write Cycle.	0		0		0		ns
17	t_W	Requires that $\overline{WR}$ be active for a minimum of [] ns. Also requires that $\overline{RD}$ be active a minimum of [] ns in the case of a Fly-By Write Cycle. Also requires that $\overline{ACKD}$ remain active a minimum of [] ns after $\overline{RD}$ has gone active in a Fly-By Write Cycle.	70		70		90		ns
18	t_S (Notes 2, 3)	Requires that the write data be active for a minimum of [] ns before $\overline{WR}$ begins to go inactive. Also requires that the write data be active for a minimum of [] ns before $\overline{RD}$ or $\overline{ACKD}$ (whichever is first) begins to go inactive in a Fly-By Write Cycle.	55		55		75		ns
19	t_H (Notes 2, 3)	Requires that the write data be kept valid for a minimum of [] ns after $\overline{WR}$ has gone inactive. Also requires that the write data be kept valid for a minimum of [] ns after $\overline{RD}$ or $\overline{ACKD}$ (whichever is first) has gone inactive in a Fly-By Write Cycle. — Word Mode	0		0		0		ns
		— Byte Mode	15		15		25		ns
20	t_{PHL}	Guarantees that the INT line will become inactive no more than [] ns after $\overline{WR}$ has gone inactive.		120		120		150	ns
21	t_{PD}	Guarantees that the data buffers will not become active before $\overline{RD}$ goes active. Also guarantees that the data buffers will not become active before $\overline{WR}$ goes active in a Fly-By Read Cycle.	0		0		0		ns
22	t_H	Requires that $\overline{ACKD}$ be active a minimum of [] ns after $\overline{RD}$ or $\overline{WR}$ has gone inactive.	0		0		0		ns
23–29	(Not used)	Not Used							

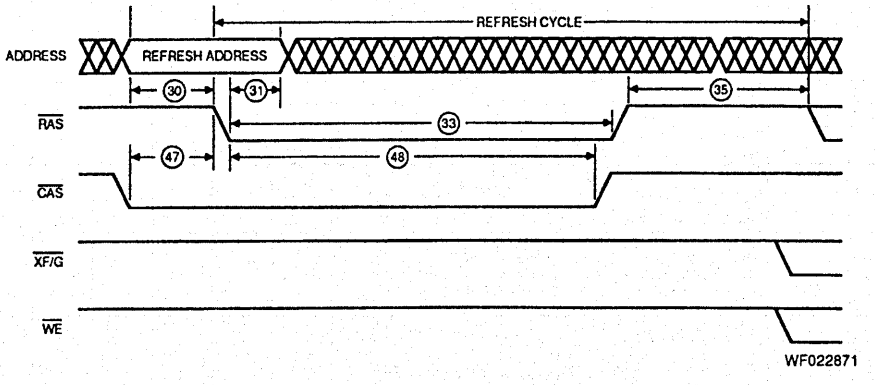
Notes: See notes at end of this section.

PRELIMINARY

SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)

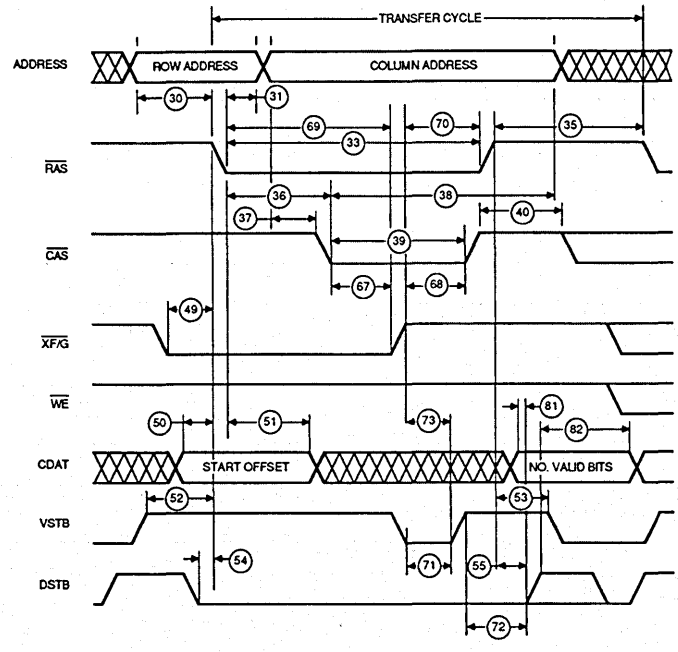


Display Memory Read Cycle



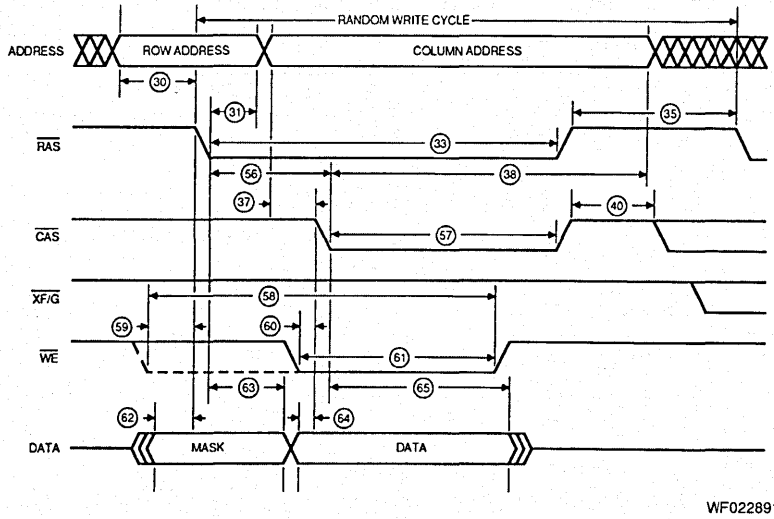
Display Memory Dynamic RAM Refresh Cycle

SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)



WF025730

Display Memory Transfer Cycle



WF022891

Display Memory Write Cycle

SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)
Display Memory Interface

No.	Parameter Symbol	Parameter Description	-20		-18		-16		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
30	t_S (Note 4)	Guarantees the row addess will be stable (valid) a minimum of [] ns before $\overline{RAS}$ begins to go active.	15		18		16		ns
31	t_H (Note 4)	Guarantees the row address will remain valid a minimum of [] ns after $\overline{RAS}$ has gone active.	35		41		45		ns
32	t_{PD} (Note 4)	Guarantees that $\overline{XF/G}$ will not go inactive until a minimum of [] ns after $\overline{RAS}$ has gone active for a read cycle.	160		181		202		ns
33	t_W (Note 4)	Guarantees that $\overline{RAS}$ will be active for a minimum of [] ns.	180		204		225		ns
34	t_H (Note 4)	Guarantees that $\overline{RAS}$ will remain active for a minimum of [] ns after $\overline{XF/G}$ has gone inactive for a read cycle.	14		14.5		15		ns
35	t_W (Note 4)	Guarantees that $\overline{RAS}$ will remain not active for a minimum of [] ns.	95		106		115		ns
36	t_{PD} (Note 4)	Guarantees that $\overline{CAS}$ will not become active until a minimum of [] ns after $\overline{RAS}$ has gone active. See parameter 56 for write cycles.	65		74		78		ns
37	t_S (Note 4)	Guarantees that the column address will be valid and stable a minimum of [] ns before $\overline{CAS}$ will go active.	13		14		15		ns
38	t_H (Note 4)	Guarantees that the column address will remain valid and stable a minimum of [] ns after $\overline{CAS}$ has gone active.	80		92		104		ns
39	t_W (Note 4)	Guarantees that $\overline{CAS}$ will be active a minimum of [] ns for a read cycle or a transfer cycle. See parameter 57 for writes cycles.	100		115		130		ns
40	t_W (Note 4)	Guarantees that $\overline{CAS}$ will remain inactive for a minimum of [] ns. This is important when a refresh cycle follows any cycle.	40		46		47		ns
41	t_{PD} (Note 4)	Guarantees that $\overline{XF/G}$ will not have gone inactive until a minimum of [] ns after $\overline{CAS}$ has gone active for a read cycle.	90		95		113		ns
42	t_{PD} (Note 4)	Guarantees that $\overline{XF/G}$ will not begin to go active until a minimum of [] ns after $\overline{RAS}$ has gone active.	39		45		46		ns
43	t_H (Note 4)	Guarantees that $\overline{CAS}$ will remain active until a minimum of [] ns after $\overline{XF/G}$ has gone inactive for a read cycle.	13		14.5		16		ns
44	t_W (Note 4)	Guarantees that $\overline{XF/G}$ will be active a minimum of [] ns for a read cycle.	105		120		125		ns
45	t_S	Requires that the read data be valid a minimum of [] ns before $\overline{XF/G}$ begins to go inactive.	17		20		23		ns

Notes: See notes at end of this section.

PRELIMINARY

SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)
Display Memory Interface (Cont'd.)

No.	Parameter Symbol	Parameter Description	-20		-18		-16		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
46	t_H	Requires that the read data remain valid a minimum of [] ns after $\overline{XF7G}$ has gone inactive.	0		0		0		ns
47	t_S (Note 4)	Guarantees that $\overline{RAS}$ will not begin to go active until a minimum of [] ns after $\overline{CAS}$ has become active in a refresh cycle.	37		43		45		ns
48	t_H (Note 4)	Guarantees that $\overline{CAS}$ will not begin to go inactive until a minimum of [] ns after $\overline{RAS}$ has gone active in a refresh cycle.	185		204		230		ns
49	t_S (Note 4)	Guarantees that $\overline{RAS}$ will not begin to go active until a minimum of [] ns after $\overline{XF7G}$ is active in a transfer cycle.	12		12		13		ns
50	t_S (Note 4)	Guarantees that the Start Offset on CDAT will be valid and stable a minimum of [] ns before $\overline{RAS}$ begins to go active in a transfer cycle.	10		10		11		ns
51	t_H (Note 4)	Guarantees that the Start Offset on CDAT will remain valid and stable a minimum of [] ns after $\overline{RAS}$ has gone active in a transfer cycle.	65		71.5		78		ns
52	t_S (Note 4)	Guarantees that VSTB will be HIGH a minimum of [] ns before $\overline{RAS}$ begins to go active in a transfer cycle.	90		102		109		ns
53	t_H (Note 4)	Guarantees that VSTB will remain HIGH a minimum of [] ns after $\overline{RAS}$ has gone inactive in a transfer cycle.	80		92		99		ns
54	t_S (Note 4)	Guarantees that DSTB will be LOW a minimum of [] ns before $\overline{RAS}$ begins to go active in a transfer cycle.	90		102		109		ns
55	t_H (Note 4)	Guarantees that DSTB will remain LOW a minimum of [] ns after $\overline{RAS}$ has gone inactive in a transfer cycle.	80		92		99		ns
56	t_{PD} (Note 4)	Guarantees that $\overline{CAS}$ will not become active until a minimum of [] ns after $\overline{RAS}$ has gone active. This is for a write cycle. See parameter 36 for read and transfer cycles.	90		99.5		109		ns
57	t_W (Note 4)	Guarantees that $\overline{CAS}$ will be active a minimum of [] ns for a write cycle. See parameter 39 for read and transfer cycles.	80		90		100		ns
58	t_W (Note 4)	Guarantees that $\overline{WE_0}$–$\overline{WE_3}$ will be active a minimum of [] in the case of a masked write.	180		204		225		ns
59	t_S (Notes 4, 5)	Guarantees that $\overline{WE_0}$–$\overline{WE_3}$ will be active a minimum of [] before $\overline{RAS}$ begins to fall in the case of masked write.	11		12.5		14		ns

Notes: See notes at end of this section.

PRELIMINARY

SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)

Display Memory Interface (Cont'd.)

No.	Parameter Symbol	Parameter Description	-20		-18		-16		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
60	t_S (Note 4)	Guarantees that $\overline{WE_0}$–$\overline{WE_3}$ will be active a minimum of ‖ before $\overline{CAS}$ begins to fall in the case of a unmasked write.	13		13		14		ns
61	t_W (Notes 4, 5)	Guarantees that $\overline{WE_0}$–$\overline{WE_3}$ will be active a minimum of ‖ in the case of an unmasked write.	78		89		100		ns
62	t_S	Guarantees that the write mask will be valid and stable on the DM pins ‖ before $\overline{RAS}$ begins to go active.	2		2		7		ns
63	t_H (Notes 4, 5)	Guarantees that the write mask will remain active ‖ after $\overline{RAS}$ has gone active.	60		66.5		73		ns
64	t_S	Guarantees that the write data will be valid and stable on the DM pins ‖ before $\overline{CAS}$ begins to go active.	2		2		7		ns
65	t_H (Notes 4, 5)	Guarantees that the write data will remain active ‖ ns after $\overline{CAS}$ has gone active.	60		66.5		73		ns
66	t_S	Guarantees that the column address will be valid ‖ ns before $\overline{XF/G}$ goes active.	−8		−10		−12		ns
67	t_{PD} (Note 4)	Guarantees that $\overline{XF/G}$ will not go inactive until a minimum of ‖ ns after $\overline{CAS}$ has gone active for a transfer cycle.	55		64		69		ns
68	t_H (Note 4)	Guarantees that $\overline{CAS}$ will remain active until a minimum of ‖ ns after $\overline{XF/G}$ has gone inactive for a transfer cycle.	38		44		47		ns
69	t_{PD} (Note 4)	Guarantees that $\overline{XF/G}$ will not go inactive until a minimum of ‖ ns after $\overline{RAS}$ has gone active for a transfer cycle.	135		153		171		ns
70	t_H (Note 4)	Guarantees that $\overline{RAS}$ will remain active until a minimum of ‖ ns after $\overline{XF/G}$ has gone inactive for a transfer cycle.	39		45		46		ns
71	t_W (Note 4)	Guarantees that VSTB will remain LOW a minimum of ‖ during a transfer cycle.	35		41		42		ns
72	t_{PH} (Note 4)	Guarantees that the first positive edge on VSTB will have occurred a minimum of ‖ before the first positive edge on DSTB following a transfer cycle.	80		92		99		ns
73	t_S (Note 4)	Guarantees that $\overline{XF/G}$ will have gone inactive a minimum of ‖ before the first positive edge on VSTB following a transfer cycle.	40		46		47		ns

Notes: See notes at end of this section.

PRELIMINARY

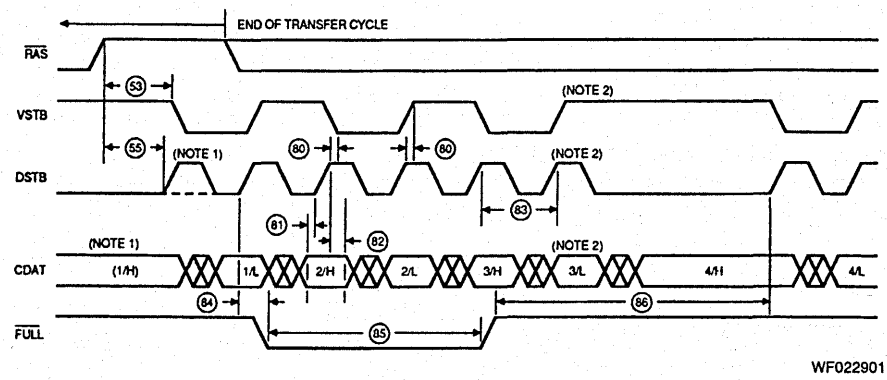
SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)

VDAF Interface Timing

No.	Parameter Symbol	Parameter Description	-20 Min.	-20 Max.	-18 Min.	-18 Max.	-16 Min.	-16 Max.	Unit
80	t_{PD}	Guarantees that VSTB will change to the new state within 0–[ ] ns following the positive edge of DSTB.	-3	10	-3	15	-3	15	ns
81	t_S	Guarantees the value on CDAT will be valid and stable a minimum of [ ] before DSTB begins to rise.	8		8		10		ns
82	t_H	Guarantees the value on CDAT will remain valid a minimum of [ ] after the rising edge of DSTB.	15		15		20		ns
83	t_{PD} (Note 4)	Guarantees the positive edges on DSTB will occur with the same period as SYSCLK.	50		56		62		ns
84	t_S	Guarantees that $\overline{FULL}$ will be recognized at the current SYSCLK cycle if it is valid a minimum of [ ] ns before the edge. This is not an operating parameter; if this setup time is not met, the part will not go metastable.		25		28		30	ns
85	t_W (Note 4)	Requires that $\overline{FULL}$ remain active at least one SYSCLK period.	50		56		62		ns
86	t_{PLH} (Note 4)	Guarantees that a positive edge will not occur on DSTB until a minimum of [ ] ns following $\overline{FULL}$ going inactive.	150		168		186		ns
87–89	(Not used)	Not Used							

Notes: See notes at end of this section.

PRELIMINARY



WF022901

Notes: 1. DSTB & CDAT for (1/H) only occurs on even boundaries.
2. One less byte may be strobed after $\overline{FULL}$ active.
3. CDAT 1/H = CDAT for 1st word HIGH byte.

VDAF Interface Timing

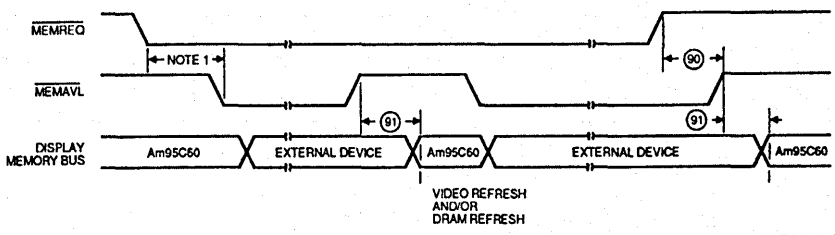
SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)

VDAF Interface Timing

No.	Parameter Symbol	Parameter Description	-20 Min.	-20 Max.	-18 Min.	-18 Max.	-16 Min.	-16 Max.	Unit
90	t_{PLH} (Note 4)	Guarantees that the Am95C60 will make MEMAVL inactive within [ ] ns of MEMREQ going inactive.		180		175		190	ns
91	t_{PD} (Notes 4, 6)	Guarantees that the Am95C60 will not begin a memory cycle for a minimum of [ ] ns following the rising edge of MEMAVL.	450		504		558		ns
92–99	(Not used)	Not Used							

Notes: See notes at end of this section.

PRELIMINARY



WF022910

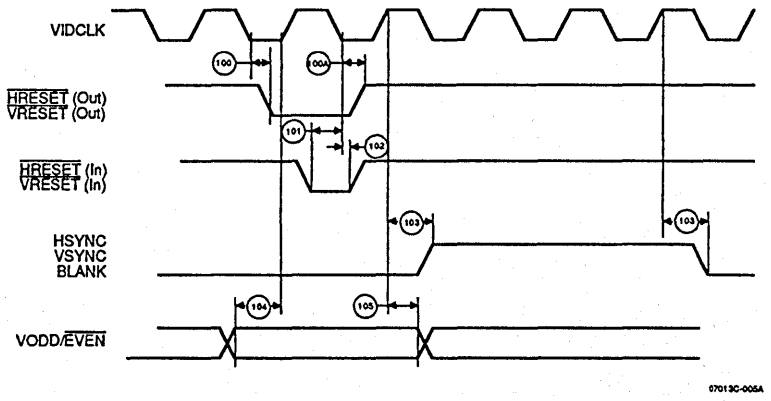
Notes: 1. $\overline{MREQ}$ to $\overline{MEMAVL}$ DELAY = Minimum 6C for arbitration cycle
= Maximum 24C

Not an operating parameter; provided for reference only.

2. Display Memory Bus = ADDR, $\overline{RAS}$, $\overline{CAS}$, $\overline{XF7G}$, $\overline{WE}$, DATA
3. Parameter 91 is referenced to $\overline{RAS}$ falling edge, which defines the start of the QPDM cycle

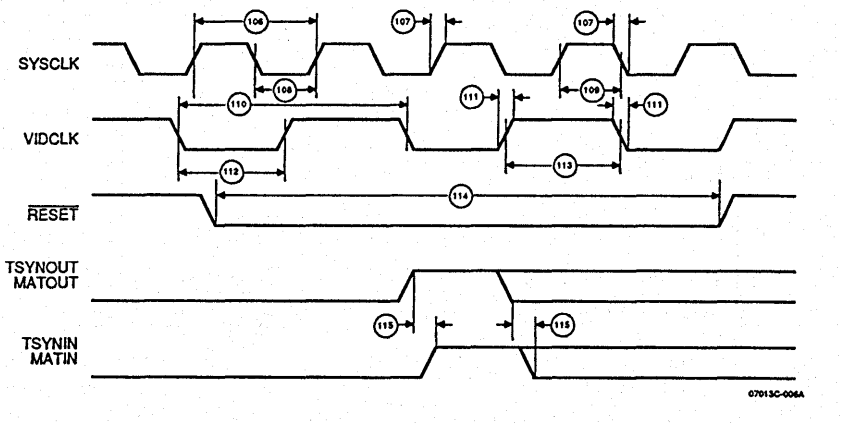
Display Memory Arbitration Timing

SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)



WF026530

Video Control Timings



WF026540

Clocks, Reset, and Am95C60 Synchronization Timings

3

SWITCHING WAVEFORMS/CHARACTERISTICS (Cont'd.)

Miscellaneous Timing

No.	Parameter Symbol	Parameter Description	-20		-18		-16		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
100	t_{PD}	Guarantees that the HIGH-to-LOW transition on HRESET and VRESET will occur a maximum of ‖ ns following the HIGH-to-LOW transition of VIDCLK. This is effective only if HRESET and VRESET are programmed as output(s).		25		25		30	ns
100A	t_H	Guarantees that the LOW-to-HIGH transition on HRESET will not occur until a minimum of ‖ following the HIGH-to-LOW transition of VIDCLK. This is effective only if HRESET and VRESET are programmed as output(s).	0		0		0		ns
101	t_S	Requires the HRESET and VRESET be valid a minimum of ‖ ns prior to the HIGH-to-LOW transition of VIDCLK.	20		20		25		ns
102	t_H	Requires that HRESET and VRESET remain valid a minimum of ‖ ns following the HIGH-to-LOW transition of VIDCLK.	0		0		0		ns
103	t_{PD}	Guarantees that transitions on BLANK, HSYNC, and VSYNC will occur within a maximum of ‖ ns of the rising edge of VIDCLK.		30		30		40	ns
104	t_S	Requires that VODD/$\overline{EVEN}$ be valid a minimum of ‖ ns prior to the rising edge of VIDCLK.	15		15		20		ns
105	t_H	Requires that VODD/$\overline{EVEN}$ remain valid a minimum of ‖ ns after VIDCLK has risen.	15		15		20		ns
106	t_{CYC}	Requires that the period of SYSCLK be between ‖ and ‖ ns.	50	500	56	500	62	500	ns
107	t_T (Note 8)	Requires that SYSCLK transition times be a maximum of ‖ ns.		5		5		5	ns
108	t_W	Requires that SYSCLK LOW time be a minimum of ‖ ns.	23		25		28		ns
109	t_W	Requires that SYSCLK HIGH time be a minimum of ‖ ns.	23		25		28		ns
110	t_{CYC}	Requires that the period of VIDCLK be between ‖ and ‖.	66	4000	66	4000	72	4000	ns
111	t_T (Note 8)	Requires that the VIDCLK transition time be a maximum of ‖ ns.		5		5		5	ns
112	t_W	Requires that the VIDCLK LOW time be a minimum of ‖ ns.	27		27		29		ns
113	t_W	Requires that VIDCLK HIGH time be a minimum of ‖ ns.	27		27		29		ns
114	t_W (Note 4)	Requires that $\overline{RESET}$ remain active a minimum of ‖ ns.	200		224		248		ns
115	t_{PD}	Requires that the external delay from TSYNOUT to TSYNIN and the external delay from MATOUT to MATIN be a maximum of ‖ ns.		17		17		20	ns

Notes: 1. Timings are relative to $\overline{CS}$ or $\overline{ACKD}$.
2. $\overline{RD}$ and $\overline{WR}$ reverse operations in Fly-By DMA cycles.
3. Timings are relative to $\overline{RD}$/$\overline{WR}$ or $\overline{ACKD}$ rising edge, whichever occurs first.
4. See Switching Characteristics Formulas.
5. This timing applies for masked writes.
6. All display memory cycles are exactly six SYSCLK cycles.
7. The units of all Switching parameters are ns.
8. Tests for parameters 107 and 111 are not being performed in manufacturing.

PRELIMINARY

Switching Characteristics Formulas

All the Switching Characteristics which reference Note 4 are calculated with the formulas which follow. Each parameter is some number of SYSCLK cycles (or parts of SYSCLK cycles). In addition, there is an adder for each of the three speed classes.

The numbers which are given in the Switching Characteristics are derived by evaluating the formulas assuming the following SYSCLK periods shown below:

FREQ	PERIOD
-20	50 ns
-18	56 ns
-16	62 ns

Users who wish to operate the Am95C60 at more than these SYSCLK periods may use these formulas to calculate the Switching Characteristics for their actual system.

Switching Characteristics Formulas

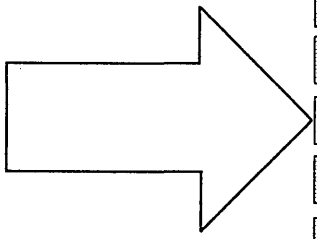
Parameter No.	Nominal	-20	-18	-16
30	c/2	-10	-10	-15
31	c	-15	-15	-17
32	7c/2	-15	-15	-15
33	4c	-20	-20	-23
34	c/2	-11	-13.5	-16
35	2c	-5	-6	-15
36	3c/2	-10	-10	-15
37	c/2	-12	-14	-16
38	2c	-20	-20	-20
39	5c/2	-25	-25	-25
40	c	-10	-10	-15
41	2c	-20	-20	-24
42	c	-11	-11	-16
43	c/2	-12	-13.5	-15
44	5c/2	-20	-20	-30
47	c	-13	-13	-17
48	4c	-15	-20	-18
49	c/2	-13	-16	-18
50	c/2	-15	-18	-20
51	3c/2	-10	-12.5	-15
52	2c	-10	-10	-15
53	2c	-10	-12.5	-15
54	2c	-10	-10	-15
55	2c	-10	-12.5	-15
56	2c	-10	-12.5	-15
57	2c	-20	-22	-24
58	4c	-20	-20	-23
59	c/2	-14	-15.5	-17
60	c/2	-12	-15	-17
61	2c	-22	-23	-24
63	3c/2	-15	-17.5	-20
65	3c/2	-15	-17.5	-20
67	3c/2	-20	-20	-24
68	c	-12	-12	-15
69	3c	-15	-15	-15
70	c	-11	-11	-16
71	c	-10	-10	-15
72	2c	-10	-10	-15
73	c	-10	-10	-15
83	c	+0	+0	+0
85	c	+0	+0	+0
86	3c	+0	+0	+0
90	5c/2	+35	+35	+35
91	9c	+0	+0	+0
114	4c	+0	+0	+0

Parameter Type Definitions

The following letter(s) specify the parameter type:

t_H	Hold Time	t_S	Setup Time
t_{PHL}	Propagation Delay Time where the output is going from HIGH to LOW	t_W	Width Time
t_{PLH}	Propagation Delay Time where the output is going from LOW to HIGH	t_T	Transition Time
t_{PD}	Propagation Delay Time where the output(s)go either HIGH or LOW orLOW to HIGH	t_{CYC}	Cycle Time
		C	Clock Period

Advanced
Micro
Devices



CHAPTER 4
MASS STORAGE

Am5380/Am53C80N*

SCSI Interface Controller

DISTINCTIVE CHARACTERISTICS

SCSI Interface

- Asynchronous interface to 1.5 megabytes per second
- Supports Initiator and Target roles
- Parity generation with optional checking
- Supports Arbitration
- Direct control of all bus signals
- High current outputs drive SCSI Bus directly

CPU Interface

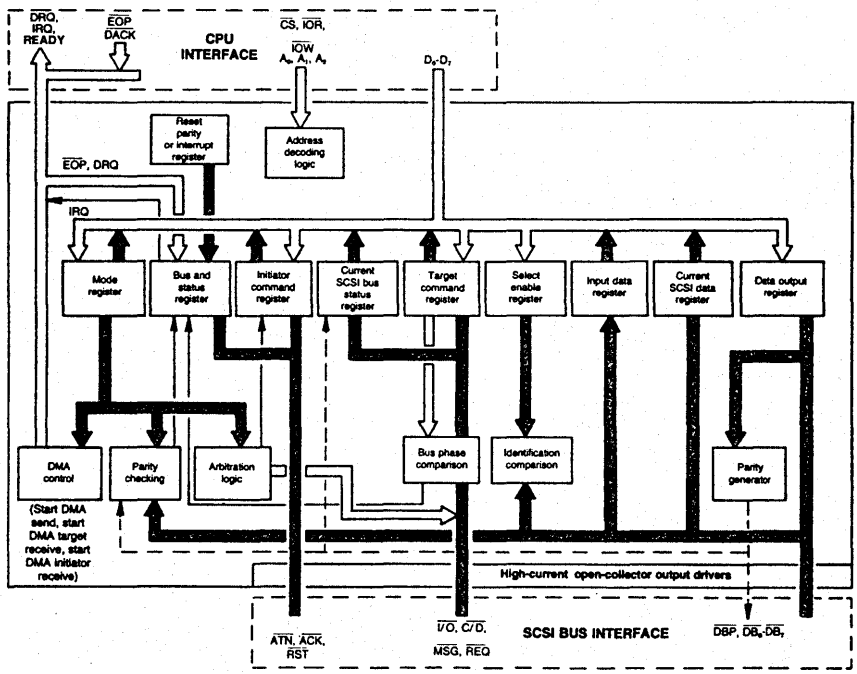
- Memory or I/O mapped interface
- DMA or programmed I/O
- Normal or Block mode DMA
- Optional CPU interrupts

GENERAL DESCRIPTION

The Am5380/Am53C80N Small Computer Systems Interface (SCSI) Interface Controller is a 40-pin NMOS/CMOS device designed to accommodate the SCSI as defined by the ANSI X3T9.2 committee. The Am5380/Am53C80N operates in both the Initiator and Target roles and can, therefore, be used in host adapter, host port and formatter designs. This device supports Arbitration, including Reselection. Special high-current open-collector output drivers, capable of sinking 48 mA at 0.5 V, allow for direct connection to the SCSI Bus.

The Am5380/Am53C80N communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers which may be addressed as standard or memory-mapped I/O. Minimal processor intervention is required for DMA transfers because the Am5380/Am53C80N controls the necessary handshake signals. The Am5380/Am53C80N interrupts the CPU when it detects a bus condition that requires attention. Normal and Block mode DMA is provided to match many popular DMA controllers.

BLOCK DIAGRAM

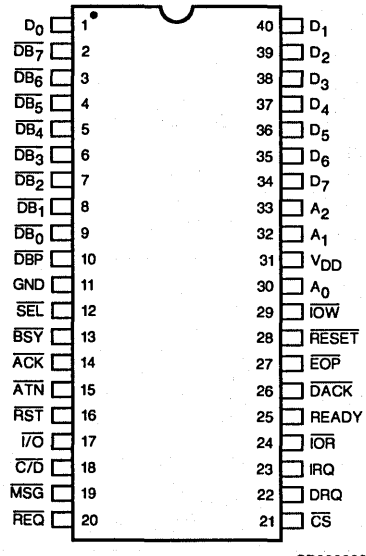


BD006561

Publication # 08289 | Rev. B | Amendment /0
Issue Date: October 1988

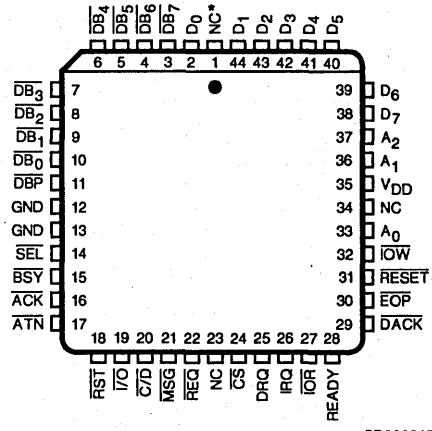
CONNECTION DIAGRAMS
Top View

DIPs



CD009900

PLCC

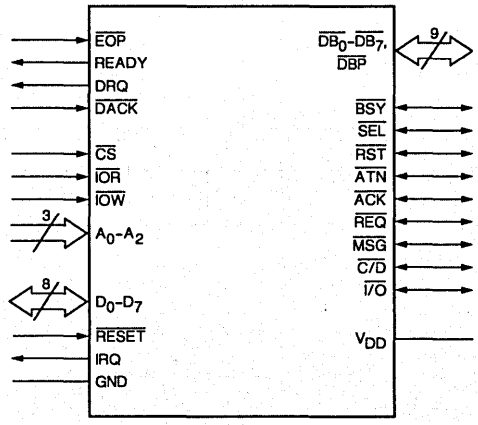


CD009915

*NC = No Connection

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

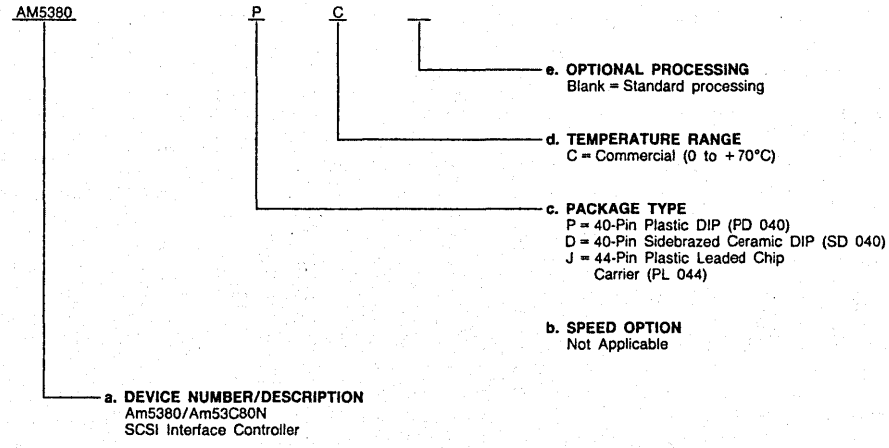


LS002643

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM5380	PC, DC, JC
AM53C80N	PC, JC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

4

PIN DESCRIPTION

Microprocessor Interface Signals

$A_0 - A_2$ Address Lines (Input)
These signals are used with $\overline{CS}$, $\overline{IOR}$ or $\overline{IOW}$ to address all internal registers.

$\overline{CS}$ Chip Select (Input, Active LOW)
$\overline{CS}$ enables a read or write of the internal register selected by $A_0 - A_2$.

$\overline{DACK}$ DMA Acknowledge (Input, Active LOW)
$\overline{DACK}$ resets DRQ and selects the data register for input or output data transfers.

DRQ DMA Request (Output)
DRQ indicates that the data register is ready to be read or written. DRQ occurs only if DMA mode is TRUE in the Command Register. DRQ is cleared by $\overline{DACK}$.

$D_0 - D_7$ Data Lines (Input/Output; Three-State, Active HIGH)
Bidirectional microprocessor data bus lines.

$\overline{EOP}$ End of Process (Input, Active LOW)
$\overline{EOP}$ is used to terminate a DMA transfer. If asserted during a DMA cycle, the current byte will be transferred, but no additional bytes will be requested.

$\overline{IOR}$ I/O Read (Input, Active LOW)
$\overline{IOR}$ is used to read an internal register selected by $\overline{CS}$ and $A_0 - A_2$. It also selects the Input Data Register when used with $\overline{DACK}$.

$\overline{IOW}$ I/O Write (Input, Active LOW)
$\overline{IOW}$ is used to write an internal register selected by $\overline{CS}$ and $A_0 - A_2$. It also selects the Output Data Register when used with $\overline{DACK}$.

IRQ Interrupt Request (Output)
IRQ alerts a microprocessor of an error condition or an event completion.

READY Ready (Output)
READY can be used to control the speed of Block mode DMA transfers. This signal goes active to indicate the chip is ready to send/receive data and remains FALSE after a transfer until the last byte is sent or until the DMA MODE bit is reset.

$\overline{RESET}$ Reset (Input, Active LOW)
$\overline{RESET}$ clears all registers. It does not force the SCSI $\overline{RST}$ signal to the active state.

Power Signals

V_{DD} +5-Volt Power Supply

GND Ground

SCSI Interface Signals

The following signals are all bidirectional, active-LOW, open-collector signals. With 48-mA sink capability, all pins interface directly with the SCSI Bus.

$\overline{ACK}$ Acknowledge (Input/Output; Open Collector, Active LOW)
Driven by an Initiator, $\overline{ACK}$ indicates an acknowledgment for a $\overline{REQ}/\overline{ACK}$ data-transfer handshake. In the Target role, $\overline{ACK}$ is received as a response to the $\overline{REQ}$ signal.

$\overline{ATN}$ Attention (Input/Output; Open Collector, Active LOW)
Driven by an Initiator, $\overline{ATN}$ indicates an Attention condition. This signal is received in the Target role.

$\overline{BSY}$ Busy (Input/Output; Open Collector, Active LOW)
This signal indicates that the SCSI Bus is being used and can be driven by both the Initiator and the Target device.

$\overline{C/D}$ Control/Data (Input/Output; Open Collector, Active LOW)
A signal driven by the Target, $\overline{C/D}$ indicates Control or Data information is on the Data Bus. This signal is received by the Initiator.

$\overline{I/O}$ Input/Output (Input/Output; Open Collector, Active LOW)
$\overline{I/O}$ is a signal driven by a Target which controls the direction of data movement on the SCSI Bus. TRUE indicates input to the Initiator. This signal is also used to distinguish between Selection and Reselection phases.

$\overline{MSG}$ Message (Input/Output; Open Collector, Active LOW)
$\overline{MSG}$ is a signal driven by the Target during the Message phase. This signal is received by the Initiator.

$\overline{REQ}$ Request (Input/Output; Open Collector, Active LOW)
Driven by a Target, $\overline{REQ}$ indicates a request for a $\overline{REQ}/\overline{ACK}$ data-transfer handshake. This signal is received by the Initiator.

$\overline{RST}$ SCSI Bus RESET (Input/Output; Open Collector, Active LOW)
The $\overline{RST}$ signal indicates an SCSI Bus RESET condition.

$\overline{DB_0} - \overline{DB_7}$, $\overline{DBP}$ Data Bits, Parity Bit (Input/Output; Open Collector, Active LOW)
These eight data bits ($\overline{DB_0} - \overline{DB_7}$), plus a parity bit ($\overline{DBP}$) form the Data Bus. $\overline{DB_7}$ is the most significant bit (MSB) and has the highest priority during the Arbitration phase. Data parity is odd. Parity is always generated and optionally checked. Parity is not valid during Arbitration.

$\overline{SEL}$ Select (Input/Output; Open Collector, Active LOW)
$\overline{SEL}$ is used by an Initiator to select a Target, or by a Target to reselect an Initiator.

FUNCTIONAL DESCRIPTION

General

The Am5380/Am53C80N Small Computer Systems Interface (SCSI) device appears as a set of eight registers to the controlling CPU. By reading and writing the appropriate registers, the CPU may initiate any SCSI Bus activity or may sample and assert any signal on the SCSI Bus. This allows the user to implement all or portions of the SCSI protocol in software. These registers are read (written) by activating $\overline{CS}$ with an address on $A_0 - A_2$ and then issuing an $\overline{IOR}$ ($\overline{IOW}$) pulse. This section describes the operation of the internal registers.

TABLE 1. REGISTER SUMMARY

Address				
A_2	A_1	A_0	R/W	Register Name
0	0	0	R	Current SCSI Data
0	0	0	W	Output Data
0	0	1	R/W	Initiator Command
0	1	0	R/W	Mode
0	1	1	R/W	Target Command
1	0	0	R	Current SCSI Bus Status
1	0	0	W	Select Enable
1	0	1	R	Bus and Status
1	0	1	W	Start DMA Send
1	1	0	R	Input Data
1	1	0	W	Start DMA Target Receive
1	1	1	R	Reset Parity/Interrupts
1	1	1	W	Start DMA Initiator Receive

Data Registers

The data registers are used to transfer SCSI commands, data, status, and message bytes between the microprocessor Data Bus and the SCSI Bus. The Am5380/Am53C80N does not interpret any information that passes through the data registers. The data registers consist of the transparent Current SCSI Data Register, the Output Data Register, and the Input Data Register.

Current SCSI Data Register — Address 0 (Read Only)

The Current SCSI Data Register is a read-only register which allows the microprocessor to read the active SCSI Data Bus. This is accomplished by activating $\overline{CS}$ with an address on $A_2 - A_0$ of 000 and issuing an $\overline{IOR}$ pulse. If parity checking is enabled, the SCSI Bus parity is checked at the beginning of the read cycle. This register is used during a programmed I/O data read or during Arbitration to check for higher priority arbitrating devices. Parity is not guaranteed valid during Arbitration.

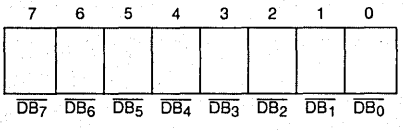


Figure 1. Current SCSI Data Register

Output Data Register — Address 0 (Write Only)

The Output Data Register is a write-only register that is used to send data to the SCSI Bus. This is accomplished by either using a normal CPU write, or under DMA control, by using $\overline{IOW}$ and $\overline{DACK}$. This register is also used to assert the proper ID bits or the SCSI Bus during the Arbitration and Selection phases.

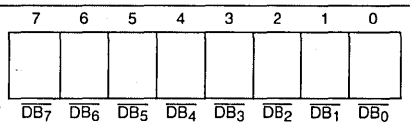


Figure 2. Output Data Register

Input Data Register — Address 6 (Read Only)

The Input Data Register is a read-only register that is used to read latched data from the SCSI Bus. Data is latched either during a DMA Target receive operation when $\overline{ACK}$ goes active or during a DMA Initiator receive when $\overline{REQ}$ goes active. The DMA MODE bit (port 2, bit 1) must be set before data can be latched in the Input Data Register. This register may be read under DMA control using $\overline{IOR}$ and $\overline{DACK}$. Parity is optionally checked when the Input Data Register is loaded.

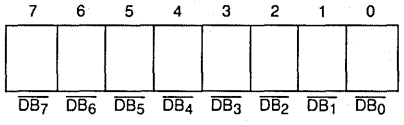


Figure 3. Input Data Register

Initiator Command Register — Address 1 (Read/Write)

The Initiator Command Register is a read/write register which is used to assert certain SCSI Bus signals, to monitor those signals, and to monitor the progress of bus arbitration. Many of these bits are significant only when being used as an Initiator; however, most can be used during Target role operation.

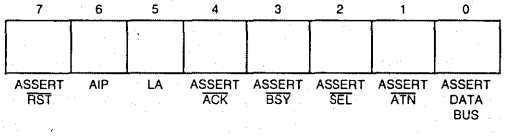


Figure 4-1. Initiator Command Register — Register Read

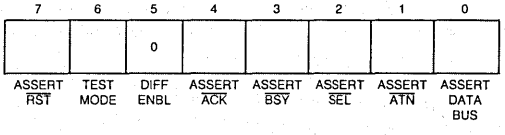


Figure 4-2. Initiator Command Register — Register Write

The following describes the operation of all bits in the Initiator Command Register.

Bit 7 — ASSERT $\overline{RST}$

Whenever a one is written to bit 7 of the Initiator Command Register, the $\overline{RST}$ signal is asserted on the SCSI Bus. The $\overline{RST}$ signal will remain asserted until this bit is reset or until an external $\overline{RESET}$ occurs. After this bit is set (1), IRQ goes active and all internal logic and control registers are reset (except for the interrupt latch and the ASSERT $\overline{RST}$ bit). Writing a zero to bit 7 of the Initiator Command Register de-asserts the $\overline{RST}$ signal. Reading this register simply reflects the status of this bit.

Bit 6 — AIP (Arbitration in Progress) (Read Bit)

This bit is used to determine if Arbitration is in progress. For this bit to be active, the ARBITRATE bit (port 2, bit 0) must

have been set previously. It indicates that a Bus-Free condition has been detected and that the chip has asserted $\overline{BSY}$ and the contents of the Output Data Register (port 0) onto the SCSI Bus. AIP will remain active until the ARBITRATE bit is reset.

Bit 6 — TEST MODE (Write Bit)

This bit may be written during a test environment to disable all output drivers, effectively removing the Am5380/Am53C80N from the circuit. Resetting this bit returns the part to normal operation.

Bit 5 — LA (Lost Arbitration) (Read Bit)

This bit, when active, indicates that the Am5380/Am53C80N detected a Bus-Free condition, arbitrated for use of the bus by asserting $\overline{BSY}$ and its ID on the Data Bus, and lost Arbitration due to $\overline{SEL}$ being asserted by another bus device. For this bit to be active, the ARBITRATE bit (port 2, bit 0) must be active.

Bit 5 — DIFF ENBL (Differential Enable) (Write Bit)

This bit should be written with a zero for proper operation.

Bit 4 — ASSERT $\overline{ACK}$

This bit is used by the bus initiator to assert $\overline{ACK}$ on the SCSI Bus. In order to assert $\overline{ACK}$, the TARGETMODE bit (port 2, bit 6) must be FALSE. Writing a zero to this bit resets $\overline{ACK}$ on the SCSI Bus. Reading this register simply reflects the status of this bit.

Bit 3 — ASSERT $\overline{BSY}$

Writing a one (1) into this bit position asserts $\overline{BSY}$ onto the SCSI Bus. Conversely, a zero resets the $\overline{BSY}$ signal. Asserting $\overline{BSY}$ indicates a successful selection or reselection and resetting this bit creates a Bus-Disconnect condition. Reading this register simply reflects the status of this bit.

Bit 2 — ASSERT $\overline{SEL}$

Writing a one (1) into this bit position asserts $\overline{SEL}$ onto the SCSI Bus. $\overline{SEL}$ is normally asserted after Arbitration has been successfully completed. $\overline{SEL}$ may be de-asserted by resetting this bit to a zero. A read of this register simply reflects the status of this bit.

Bit 1 — ASSERT $\overline{ATN}$

$\overline{ATN}$ may be asserted on the SCSI Bus by setting this bit to a one (1) if the TARGETMODE bit (port 2, bit 6) is FALSE. $\overline{ATN}$ is normally asserted by the initiator to request a Message Out bus phase. Note that since ASSERT $\overline{SEL}$ and ASSERT $\overline{ATN}$ are in the same register, a select with $\overline{ATN}$ may be implemented with one CPU write. $\overline{ATN}$ may be de-asserted by resetting this bit to zero. A read of this register simply reflects the status of this bit.

Bit 0 — ASSERT DATA BUS

The ASSERT DATA BUS bit, when set, allows the contents of the Output Data Register to be enabled as chip outputs on the signals $\overline{DB_0} - \overline{DB_7}$. Parity is also generated and asserted on $\overline{DBP}$.

When connected as an Initiator, the outputs are only enabled if the TARGETMODE bit (port 2, bit 6) is FALSE, the received signal $\overline{I/O}$ is FALSE, and the phase signals ($\overline{C/D}$, $\overline{I/O}$, and $\overline{MSG}$) match the contents of the ASSERT $\overline{C/D}$, ASSERT $\overline{I/O}$, and ASSERT $\overline{MSG}$ in the Target Command Register.

This bit should also be set during DMA send operations.

Mode Register — Address 2 (Read/Write)

The Mode Register is used to control the operation of the chip. This register determines whether the Am5380/Am53C80N operates as an Initiator or a Target, whether DMA transfers are being used, whether parity is checked, and whether interrupts are generated on various external conditions. This register may be read to check the value of these internal control bits. Figure 5 describes the operation of these control bits.

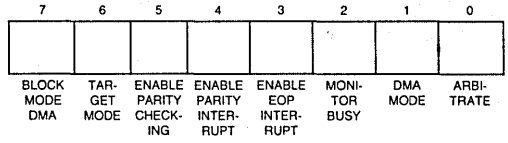


Figure 5. Mode Register

Bit 7 — BLOCK MODE DMA

The BLOCK MODE DMA bit controls the characteristics of the DMA DRQ-$\overline{DACK}$ handshake. When this bit is reset (0) and the DMA MODE bit is active (1), the DMA handshake uses the normal interlocked handshake, and the rising edge of $\overline{DACK}$ indicates the end of each byte being transferred. In block mode operations, BLOCK MODE DMA bit set (1) and DMA MODE bit set (1), the end of $\overline{IOR}$ or $\overline{IOW}$ signifies the end of each byte transferred and $\overline{DACK}$ is allowed to remain active throughout the DMA operation. READY can then be used to request the next transfer.

Bit 6 — TARGETMODE

The TARGETMODE bit allows the Am5380/Am53C80N to operate as either an SCSI Bus Initiator, bit reset (0), or as an SCSI Bus Target device, bit set (1). In order for the signals $\overline{ATN}$ and $\overline{ACK}$ to be asserted on the SCSI Bus, the TARGETMODE bit must be reset (0). In order for the signals $\overline{C/D}$, $\overline{I/O}$, $\overline{MSG}$, and $\overline{REQ}$ to be asserted on the SCSI Bus, the TARGETMODE bit must be set (1).

Bit 5 — ENABLE PARITY CHECKING

The ENABLE PARITY CHECKING bit determines whether parity errors will be ignored or saved in the parity error latch. If this bit is reset (0), parity will be ignored. Conversely, if this bit is set (1), parity errors will be saved.

Bit 4 — ENABLE PARITY INTERRUPT

The ENABLE PARITY INTERRUPT bit, when set (1), will cause an interrupt (IRQ) to occur if a parity error is detected. A parity interrupt will only be generated if the ENABLE PARITY CHECKING bit (bit 5) is also enabled (1).

Bit 3 — ENABLE EOP INTERRUPT

The ENABLE EOP INTERRUPT, when set (1), causes an interrupt to occur when the $\overline{EOP}$ (End of Process) signal is received from the DMA controller logic.

Bit 2 — MONITOR BUSY

The MONITOR BUSY bit, when TRUE (1), causes an interrupt to be generated for an unexpected loss of $\overline{BSY}$. When the interrupt is generated due to loss of $\overline{BSY}$, the lower six bits of the Initiator Command Register are reset (0) and all signals are removed from the SCSI Bus.

Bit 1 — DMA MODE

The DMA MODE bit is normally used to enable a DMA transfer and must be set (1) prior to writing ports 5 through 7. Ports 5 through 7 are used to start DMA transfers. The TARGETMODE bit (port 2, bit 6) must be consistent with writes to port 6 and 7 [i.e., set (1) for a write to port 6 and reset (0) for a write to port 7]. The control bit ASSERT DATA BUS (port 1, bit 0) must be TRUE (1) for all DMA send operations. In the DMA mode, $\overline{REQ}$ and $\overline{ACK}$ are automatically controlled.

The DMA MODE bit is not reset upon the receipt of an $\overline{EOP}$ signal. Any DMA transfer may be stopped by writing a zero into this bit location; however, care must be taken not to cause $\overline{CS}$ and $\overline{DACK}$ to be active simultaneously.

Bit 0 — ARBITRATE

The ARBITRATE bit is set (1) to start the Arbitration process. Prior to setting this bit, the Output Data Register should contain the proper SCSI device ID value. Only one data bit should be active for SCSI Bus Arbitration. The Am5380/Am53C80N will wait for a Bus-Free condition before entering the Arbitration phase. The results of the Arbitration phase may be determined by reading the status bits LA and AIP (port 1, bits 5 and 6, respectively).

Target Command Register — Address 3 (Read/Write)

When connected as a target device, the Target Command Register allows the CPU to control the SCSI Bus Information Transfer phase and/or to assert $\overline{REQ}$ simply by writing this register. The TARGETMODE bit (port 2, bit 6) must be TRUE (1) for bus assertion to occur. The SCSI Bus phases are described in Table 2.

TABLE 2. SCSI INFORMATION TRANSFER PHASES

Bus Phase	ASSERT $\overline{I/O}$	ASSERT $\overline{C/D}$	ASSERT $\overline{MSG}$
Data Out	0	0	0
Unspecified	0	0	1
Command	0	1	0
Message Out	0	1	1
Data In	1	0	0
Unspecified	1	0	1
Status	1	1	0
Message In	1	1	1

When connected as an Initiator with DMA Mode TRUE, if the phase lines ($\overline{I/O}$, $\overline{C/D}$, and $\overline{MSG}$) do not match the phase bits in the Target Command Register, a phase-mismatch interrupt is generated when $\overline{REQ}$ goes active. In order to send data as an Initiator, the ASSERT $\overline{I/O}$, ASSERT $\overline{C/D}$, and ASSERT $\overline{MSG}$ bits must match the corresponding bits in the Current SCSI Bus Status Register (port 4). The ASSERT $\overline{REQ}$ bit (bit 3) has no meaning when operating as an Initiator.

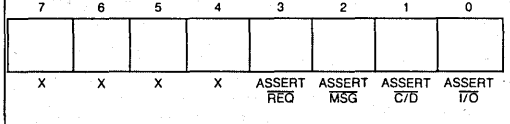


Figure 6. Target Command Register

Current SCSI Bus Status Register — Address 4 (Read Only)

The Current SCSI Bus Status Register is a read-only register which is used to monitor seven SCSI Bus control signals plus the Data Bus parity bit. For example, an Initiator device can use this register to determine the current bus phase and to poll $\overline{REQ}$ for pending data transfers. This register may also be used to determine why a particular interrupt occurred. Figure 7 describes the Current SCSI Bus Status Register.

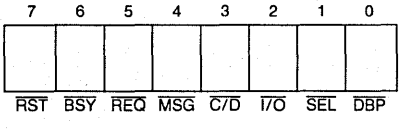


Figure 7. Current SCSI Bus Status Register

Select Enable Register — Address 4 (Write Only)

The Select Enable Register is a write-only register which is used as a mask to monitor a signal ID during a selection attempt. The simultaneous occurrence of the correct ID bit, $\overline{BSY}$ FALSE, and $\overline{SEL}$ TRUE will cause an interrupt. This interrupt can be disabled by resetting all bits in this register. If the ENABLE PARITY CHECKING bit (port 2, bit 5) is active (1), parity will be checked during selection.

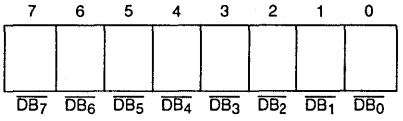


Figure 8. Select Enable Register

Bus and Status Register — Address 5 (Read Only)

The Bus and Status Register is a read-only register which can be used to monitor the remaining SCSI control signals not found in the Current SCSI Bus Status Register ($\overline{ATN}$ and $\overline{ACK}$), as well as six other status bits. The following describes each bit of the Bus and Status Register individually.

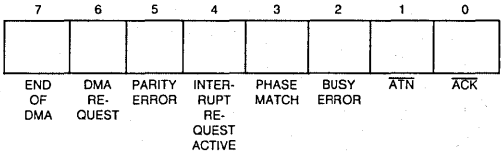


Figure 9. Bus and Status Register

Bit 7 — END OF DMA TRANSFER

The END OF DMA TRANSFER bit is set if $\overline{EOP}$, $\overline{DACK}$, and either $\overline{IOR}$ or $\overline{IOW}$ are simultaneously active for at least 100 ns. Since the $\overline{EOP}$ signal can occur during the last byte sent to the Output Data Register (port 0), the $\overline{REQ}$ and $\overline{ACK}$ signals should be monitored to ensure that the last byte has been transferred. This bit is reset when the DMA MODE bit is reset (0) in the Mode Register (port 2).

Bit 6 — DMA REQUEST

The DMA REQUEST bit allows the CPU to sample the output pin DRQ. DRQ can be cleared by asserting $\overline{DACK}$ or by resetting the DMA MODE bit (bit 1) in the Mode Register (port 2). The DRQ signal does not reset when a phase-mismatch interrupt occurs.

Bit 5 — PARITY ERROR

This bit is set if a parity error occurs during a data receive or a device selection. The PARITY ERROR bit can only be set (1) if the ENABLE PARITY CHECK bit (port 2, bit 5) is active (1). This bit may be cleared by reading the Reset Parity/Interrupt Register (port 7).

Bit 4 — INTERRUPT REQUEST ACTIVE

This bit is set if an enabled interrupt condition occurs. It reflects the current state of the IRQ output and can be cleared by reading the Reset Parity/Interrupt Register (port 7).

Bit 3 — PHASE MATCH

The SCSI signals, $\overline{MSG}$, $\overline{C/D}$, and $\overline{I/O}$, represent the current Information Transfer phase. The PHASE MATCH bit indicates whether the current SCSI Bus phase matches the lower 3 bits of the Target Command Register. PHASE MATCH is continu-

ously updated and is only significant when operating as a Bus Initiator. A phase match is required for data transfers to occur on the SCSI Bus.

Bit 2 — BUSY ERROR

The BUSY ERROR bit is active if an unexpected loss of the $\overline{BSY}$ signal has occurred. This latch is set whenever the MONITOR BUSY bit (port 2, bit 2) is TRUE and $\overline{BSY}$ is FALSE. An unexpected loss of $\overline{BSY}$ will disable any SCSI outputs and will reset the DMA MODE bit (port 2, bit 1).

Bit 1 — ATN

This bit reflects the condition of the SCSI Bus control signal $\overline{ATN}$. This signal is normally monitored by the Target device.

Bit 0 — ACK

This bit reflects the condition of the SCSI Bus control signal $\overline{ACK}$. This signal is normally monitored by the Target device.

DMA Registers

Three write-only registers are used to initiate all DMA activity. They are Start DMA Send (port 5), Start DMA Target Receive (port 6), and Start DMA Initiator Receive (port 7). Simply writing these registers starts the DMA transfers. Data presented to the Am5380/Am53C80N on signals $D_0 - D_7$ during the register write is meaningless and has no effect on the operation. Prior to writing these registers, the BLOCK MODE DMA bit (bit 7). The DMA MODE bit (bit 1) and the TARGETMODE bit (bit 6) in the Mode Register (port 2) must be appropriately set. The individual registers are briefly described as follows:

Start DMA Send — Address 5 (Write Only)

This register is written to initiate a DMA send, from the DMA to the SCSI Bus, for either Initiator or Target role operations. The DMA MODE bit (port 2, bit 1) must be set prior to writing this register.

Start DMA Target Receive — Address 6 (Write Only)

This register is written to initiate a DMA receive — from the SCSI Bus to the DMA, for Target operation only. The DMA MODE bit (bit 1) and the TARGETMODE bit (bit 6) in the Mode Register (port 2) must both be set (1) prior to writing this register.

Start DMA Initiator Receive — Address 7 (Write Only)

This register is written to initiate a DMA receive — from the SCSI Bus to the DMA, for Initiator operation only. The DMA MODE bit (bit 6) must be FALSE (0) in the Mode Register (port 2) prior to writing this register.

Reset Parity/Interrupt — Address 7 (Read Only)

Reading this register resets the PARITY ERROR bit (bit 5), the INTERRUPT REQUEST bit (bit 4), and the BUSY ERROR bit (bit 2) in the Bus and Status Register (port 5).

On-Chip SCSI Hardware Support

The Am5380/Am53C80N is easy to use because of its simple architecture. The chip allows direct control and monitoring of the SCSI Bus by providing a latch for each signal. However, portions of the protocol define timings which are much too quick for traditional microprocessors to control. Therefore, hardware support has been provided for DMA transfers, bus arbitration, phase-change monitoring, bus disconnection, bus reset, parity generation, parity checking, and device selection/reselection.

Arbitration is accomplished using a Bus-Free filter to continuously monitor $\overline{BSY}$. If $\overline{BSY}$ remains inactive for at least 400 ns then the SCSI Bus is considered free and Arbitration may begin. Arbitration will begin if the bus is free, $\overline{SEL}$ is inactive, and the ARBITRATION bit (port 2, bit 0) is active. Once arbitration has begun ($\overline{BSY}$ asserted), an arbitration delay of 2.2 μs must elapse before the Data Bus can be examined to determine if Arbitration has been won. This delay must be implemented in the controlling software driver.

The Am5380/Am53C80N is a clockwise device. Delays such as bus-free delay, bus-set delay, and bus-settle delay are implemented using gate delays. These delays may differ between devices because of inherent process variations, but are well within the proposed ANSI X3T9.2 specification.

Interrupts

The Am5380/Am53C80N provides an interrupt output (IRQ) to indicate a task completion or an abnormal bus occurrence. The use of interrupts is optional and may be disabled by resetting the appropriate bits in the Mode Register (port 2) or the Select Enable Register (port 4).

When an interrupt occurs, the Bus and Status Register and the Current SCSI Bus Status Register must be read to determine which condition created the interrupt. IRQ can be reset simply by reading the Reset Parity/Interrupt Register (port 7) or by an external chip reset ($\overline{RESET}$ active for 200 ns).

Assuming the Am5380/Am53C80N has been properly initialized, an interrupt will be generated if the chip is selected or reselected, if an $\overline{EOP}$ signal occurs during a DMA transfer, if an SCSI Bus reset occurs, if a parity error occurs during a data transfer, if a bus phase mismatch occurs, or if an SCSI Bus disconnection occurs.

Selection/Reselection

The Am5380/Am53C80N can generate a select interrupt if $\overline{SEL}$ is TRUE (1), its device ID is TRUE (1), and $\overline{BSY}$ is FALSE for at least a bus-settle delay (400 ns). If $\overline{I/O}$ is active, this should be considered a reselect interrupt. The correct ID bit is determined by a match in the Select Enable Register (port 4). Only a single bit match is required to generate an interrupt. This interrupt may be disabled by writing zeros into all bits of the Select Enable Register.

If parity is supported, parity should also be good during the selection phase. Therefore, if the ENABLE PARITY bit (port 2, bit 5) is active, then the PARITY ERROR bit should be checked to ensure that a proper selection has occurred. The ENABLE PARITY INTERRUPT bit need not be set for this interrupt to be generated.

The proposed SCSI specification also requires that no more that two device IDs be active during the selection process. To ensure this, the Current SCSI Data Register (port 0) should be read.

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) are displayed in Figures 10 and 11, respectively.

7	6	5	4	3	2	1	0
0	0	0	1	X	0	X	0
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Figure 10. Bus and Status Register

7	6	5	4	3	2	1	0
0	0	0	X	X	X	1	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Figure 11. Current SCSI Bus Status Register

End of Process (EOP) Interrupt

An End of Process signal ($\overline{EOP}$) which occurs during a DMA transfer (DMAMODE TRUE) will set the END OF DMA Status bit (port 5, bit 7) and will optionally generate an interrupt if ENABLE EOP INTERRUPT bit (port 2, bit 3) is TRUE. The $\overline{EOP}$ pulse will not be recognized (END OF DMA bit set) unless $\overline{EOP}$, $\overline{DACK}$, and either $\overline{IOR}$ or $\overline{IOW}$ are concurrently active for at least 100 ns. DMA transfers can still occur if $\overline{EOP}$ was not asserted at the correct time. This interrupt can be disabled by resetting the ENABLE EOP INTERRUPT bit.

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) for this interrupt are shown in Figures 12 and 13, respectively.

7	6	5	4	3	2	1	0
1	0	0	1	0	0	0	X
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Figure 12. Bus and Status Register

7	6	5	4	3	2	1	0
0	1	X	X	X	X	0	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Figure 13. Current SCSI Bus Status Register

The END OF DMA bit is used to determine when a block transfer is complete. Receive operations are complete when there is no data left in the chip and no additional handshakes occurring. The only exception to this is receiving data as an Initiator and the Target opts to send additional data for the same phase. In this case, $\overline{REQ}$ goes active and the new data is present in the Input Data Register. Since a phase-mismatch interrupt will not occur, $\overline{REQ}$ and $\overline{ACK}$ need to be sampled to determine that the Target is attempting to send more data.

For send operations, the END OF DMA bit is set when the DMA finishes its transfer, but the SCSI transfer may still be in progress. If connected as a Target, $\overline{REQ}$ and $\overline{ACK}$ should be sampled until both are FALSE. If connected as an Initiator, a phase change interrupt can be used to signal the completion of the previous phase. It is possible for the Target to request additional data for the same phase. In this case, a phase change will not occur and both $\overline{REQ}$ and $\overline{ACK}$ must be sampled to determine when the last byte was transferred.

SCSI Bus Reset

The Am5380/Am53C80N generates an interrupt when the $\overline{RST}$ signal transitions to TRUE. The device releases all bus signals within a bus-clear delay (800 ns) of this transition. This interrupt also occurs after setting the ASSERT $\overline{RST}$ bit (port 1, bit 7). This interrupt cannot be disabled. (Note: $\overline{RST}$ is not latched in bit 7 of the Current SCSI Bus Status Register and may not be active when this port is read. For this case, the Bus Reset interrupt may be determined by default).

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) are displayed in Figures 14 and 15, respectively.

7	6	5	4	3	2	1	0
0	X	0	1	X	0	X	X
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Figure 14. Bus and Status Register

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Figure 15. Current SCSI Bus Status Register

Parity Error

An interrupt is generated for a received parity error if the ENABLE PARITY CHECK (bit 5) and the ENABLE PARITY INTERRUPT (bit 4) bits are set (1) in the Mode Register (port 2). Parity is checked during a read of the Current SCSI Data Register (port 0) and during a DMA receive operation. A parity error can be detected without generating an interrupt by disabling the ENABLE PARITY INTERRUPT bit and checking the PARITY ERROR flag (port 5, bit 5).

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) are displayed in Figures 16 and 17, respectively.

7	6	5	4	3	2	1	0
0	X	1	1	1	0	X	X
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Figure 16. Bus and Status Register

7	6	5	4	3	2	1	0
0	1	1	X	X	X	0	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Figure 17. Current SCSI Bus Status Register

Bus Phase Mismatch

The SCSI phase lines are comprised of the signals $\overline{I/O}$, $\overline{C/D}$, and $\overline{MSG}$. These signals are compared with the corresponding bits in the Target Command Register: ASSERT $\overline{I/O}$ (bit 0), ASSERT $\overline{C/D}$ (bit 1), and ASSERT $\overline{MSG}$ (bit 2). The comparison occurs continually and is reflected in the PHASE MATCH bit (bit 3) of the Bus and Status Register (port 5). If the DMA MODE bit (port 2, bit 1) is active and a phase mismatch occurs when $\overline{REQ}$ transitions from FALSE to TRUE, an interrupt (IRQ) is generated.

A phase mismatch prevents the recognition of $\overline{REQ}$ and removes the chip from the bus during an Initiator send operation ($\overline{DB_0}$ – $\overline{DB_7}$ and $\overline{DBP}$ will not be driven even though the ASSERT DATA BUS bit (port 1, bit 0) is active). This interrupt is only significant when connected as an Initiator and may be disabled by resetting the DMA MODE bit (Note: it is possible for this interrupt to occur when connected as a Target if another device is driving the phase lines to a different state).

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) are displayed in Figures 18 and 19, respectively.

7	6	5	4	3	2	1	0
0	0	0	1	0	0	X	0
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Figure 18. Bus and Status Register

7	6	5	4	3	2	1	0
0	1	X	X	X	X	0	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Figure 19. Current SCSI Bus Status Register

Loss of $\overline{BSY}$

If the MONITOR BUSY bit (bit 2) in the Mode Register (port 2) is active, an interrupt will be generated if the $\overline{BSY}$ signal goes FALSE for at least a bus-settle delay (400 ns). This interrupt may be disabled by resetting the MONITOR BUSY bit. Register values are displayed in Figures 20 and 21.

7	6	5	4	3	2	1	0
0	0	0	1	X	1	0	0
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Figure 20. Bus and Status Register

7	6	5	4	3	2	1	0
0	0	0	X	X	X	0	0
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Figure 21. Current SCSI Bus Status Register

Reset Conditions

Three possible reset situations exist with the Am5380/Am53C80N, as follows:

Hardware Chip Reset

When the signal $\overline{RST}$ is active for at least 200 ns, the Am5380/Am53C80N device is re-initialized and all internal logic and control registers are cleared. This is a chip reset only and does not create an SCSI Bus-Reset condition.

SCSI Bus Reset ($\overline{RST}$) Received

When an SCSI $\overline{RST}$ signal is received, an IRQ interrupt is generated and a chip reset is performed. All internal logic and registers are cleared, except for the IRQ interrupt latch and the ASSERT $\overline{RST}$ bit (bit 7) in the Initiator Command Register (port 1). (Note: The $\overline{RST}$ signal may be sampled by reading the Current SCSI Bus Status Register (port 4); however, this signal is not latched and may not be present when this port is read.)

SCSI Bus Reset ($\overline{RST}$) Issued

If the CPU sets the ASSERT $\overline{RST}$ bit (bit 7) in the Initiator Command Register (port 1), the $\overline{RST}$ signal goes active on the SCSI Bus and an internal reset is performed. Again, all internal logic and registers are cleared except for the IRQ interrupt latch and the ASSERT $\overline{RST}$ bit (bit 7) in the Initiator Command Register (port 1). The $\overline{RST}$ signal will continue to be active until the ASSERT $\overline{RST}$ bit is reset or until a hardware reset occurs.

Data Transfers

Data may be transferred between SCSI Bus devices in one of four modes: 1) Programmed I/O, 2) Normal DMA, 3) Block Mode DMA, or 4) Pseudo DMA. The following sections describe these modes in detail (Note: for all data transfer operations $\overline{DACK}$ and $\overline{CS}$ should never be active simultaneously).

Programmed I/O Transfers

Programmed I/O is the most primitive form of data transfer. The $\overline{REQ}$ and $\overline{ACK}$ handshake signals are individually monitored and asserted by readinig and writing the appropriate register bits. This type of transfer is normally used when transferring small blocks of data such as command blocks or message and status bytes.

An Initiator send operation would begin by setting the $\overline{C/D}$, $\overline{I/O}$, and $\overline{MSG}$ bits in the Target Command Register to the correct state so that a phase match exists. In addition to the phase match condition, it is necessary for the ASSERT DATA BUS bit (port 1, bit 0) to be TRUE and the received I/O signal to be FALSE for the Am5380/Am53C80N to send data.

For each transfer, the data is loaded into the Output Data Register (port 0). The CPU then waits for the $\overline{REQ}$ bit (port 4, bit 5) to become active. Once $\overline{REQ}$ goes active, the PHASE MATCH bit (port 5, bit 3) is checked and the ASSERT $\overline{ACK}$ bit (port 1, bit 4) is set. The $\overline{REQ}$ bit is sampled until it becomes FALSE and the CPU resets the ASSERT $\overline{ACK}$ bit to complete the transfer.

Normal DMA Mode

DMA transfers are normally used for large block transfers. The SCSI chip outputs a DMA request (DRQ) whenever it is ready for a byte transfer. External DMA logic uses this DRQ signal to generate $\overline{DACK}$ and an $\overline{IOR}$ or an $\overline{IOW}$ pulse to the Am5380/Am53C80N. DRQ goes inactive when $\overline{DACK}$ is asserted and $\overline{DACK}$ goes inactive some time after the minimum read or write pulse width. This process is repeated for every byte. For this mode, $\overline{DACK}$ should not be allowed to cycle unless a transfer is taking place.

Block Mode DMA

Some popular DMA controllers such as the Am9517A provide a Block mode DMA transfer. This type of transfer allows the DMA controller to transfer blocks of data without relinquishing the use of the Data Bus to the CPU after each byte is transferred; thus, faster transfer rates are achieved by eliminating the repetitive access and release of the CPU Bus.

If the BLOCK MODE DMA bit (port 2, bit 7) is active, the Am5380/Am53C80N will begin the transfer by asserting DRQ. The DMA controller then asserts $\overline{DACK}$ for the remainder of the block transfer. DRQ goes inactive for the duration of the transfer. The READY output is used to control the transfer rate.

Non-Block mode DMA transfers end when $\overline{DACK}$ goes FALSE, whereas Block mode transfers end when $\overline{IOR}$ or $\overline{IOW}$ becomes inactive. Since this is the case, DMA transfers may be started sooner in a Block mode transfer.

To obtain optimum performance in Block mode operation, the DMA logic may optionally use the normal DMA mode interlocking handshake. READY is still available to throttle the DMA transfer, but DRQ is 30 to 40 ns faster than READY and may be used to start the cycle sooner.

The methods described under 'Halting a DMA Operation' apply for all DMA operations.

Pseudo DMA Mode

To avoid the tedium of monitoring and asserting the request/acknowledge handshake signals for programmed I/O transfers, the system may be designed to implement a pseudo DMA mode. This mode is implemented by programming the Am5380/Am53C80N to operate in the DMA mode, but using the CPU to emulate the DMA handshake. DRQ may be detected by polling the DMA REQUEST bit (bit 6) in the Bus and Status Register (port 5), by sampling the signal through an external port or by using it to generate a CPU interrupt. Once DRQ is detected, the CPU can perform a read or write data transfer. This CPU read/write is externally decoded to generate the appropriate $\overline{DACK}$ and $\overline{IOR}$ or $\overline{IOW}$ signals.

Often, external decoding logic is necessary to generate the Am5380/Am53C80N $\overline{CS}$ signal. This same logic may be used to generate $\overline{DACK}$ at no extra system cost and provide an increased performance in programmed I/O transfers.

Halting a DMA Operation

The $\overline{EOP}$ signal is not the only way to halt a DMA transfer. A bus phase mismatch or a reset of the DMA MODE bit (port 2, bit 1) can also terminate a DMA cycle for the current bus phase.

Using the $\overline{EOP}$ Signal

If $\overline{EOP}$ is used, it should be asserted for at least 100 ns while $\overline{DACK}$ and $\overline{IOR}$ or $\overline{IOW}$ are simultaneously active. Note, however, that if $\overline{IOR}$ or $\overline{IOW}$ is not active an interrupt will be generated, but the DMA activity will continue. The $\overline{EOP}$ signal does not reset the DMA MODE bit. Since the $\overline{EOP}$ signal can occur during the last byte sent to the Output Data Register (port 0), the $\overline{REQ}$ and $\overline{ACK}$ signals should be monitored to ensure that the last byte has transferred.

Bus Phase Mismatch Interrupt

A bus phase mismatch interrupt may be used to halt the transfer if operating as an Initiator. Using this method frees the host from maintaining a data length counter and frees the DMA logic from providing the $\overline{EOP}$ signal. If performing an Initiator send operation, the Am5380/Am53C80N requires $\overline{DACK}$ to cycle before $\overline{ACK}$ goes inactive. Since phase changes cannot occur if $\overline{ACK}$ is active, either $\overline{DACK}$ must be cycled after the last byte is sent or the DMA MODE bit must be reset in order to receive the phase mismatch interrupt.

Resetting the DMA MODE Bit

A DMA operation may be halted at any time simply by resetting the DMA MODE bit. It is recommended that the DMA MODE bit be reset after receiving an $\overline{EOP}$ or bus phase-mismatch interrupt. The DMA MODE bit must then be set before writing any of the start DMA registers for subsequent bus phases.

If resetting the DMA MODE bit is used instead of $\overline{EOP}$ for Target role operation, then care must be taken to reset this bit at the proper time. If receiving data as a Target device, the DMA MODE bit must be reset once the last DRQ is received and before $\overline{DACK}$ is asserted to prevent an additional $\overline{REQ}$ from occurring. Resetting this bit causes DRQ to go inactive. However, the last byte received remains in the Input Data Register and may be obtained either by performing a normal CPU read or by cycling $\overline{DACK}$ and $\overline{IOR}$. In most cases $\overline{EOP}$ is easier to use when operating as a Target device.

Flowcharts

Flowcharts are provided (see Figures 22 through 25) as a guideline to facilitate your firmware development. Firmware will vary depending on the application and the level of the SCSI protocol being supported.

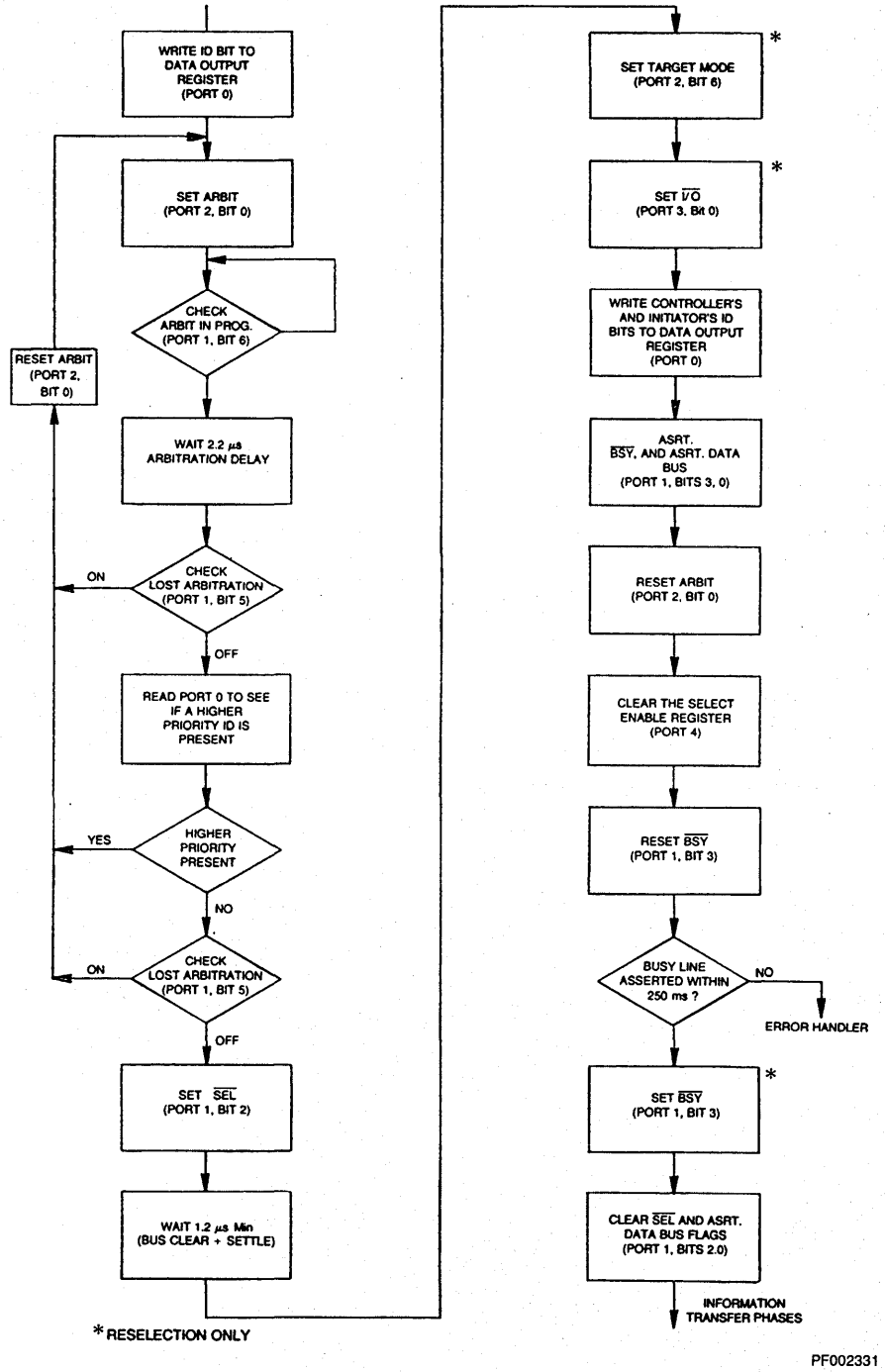


Figure 22. Arbitration and (Re) Selection

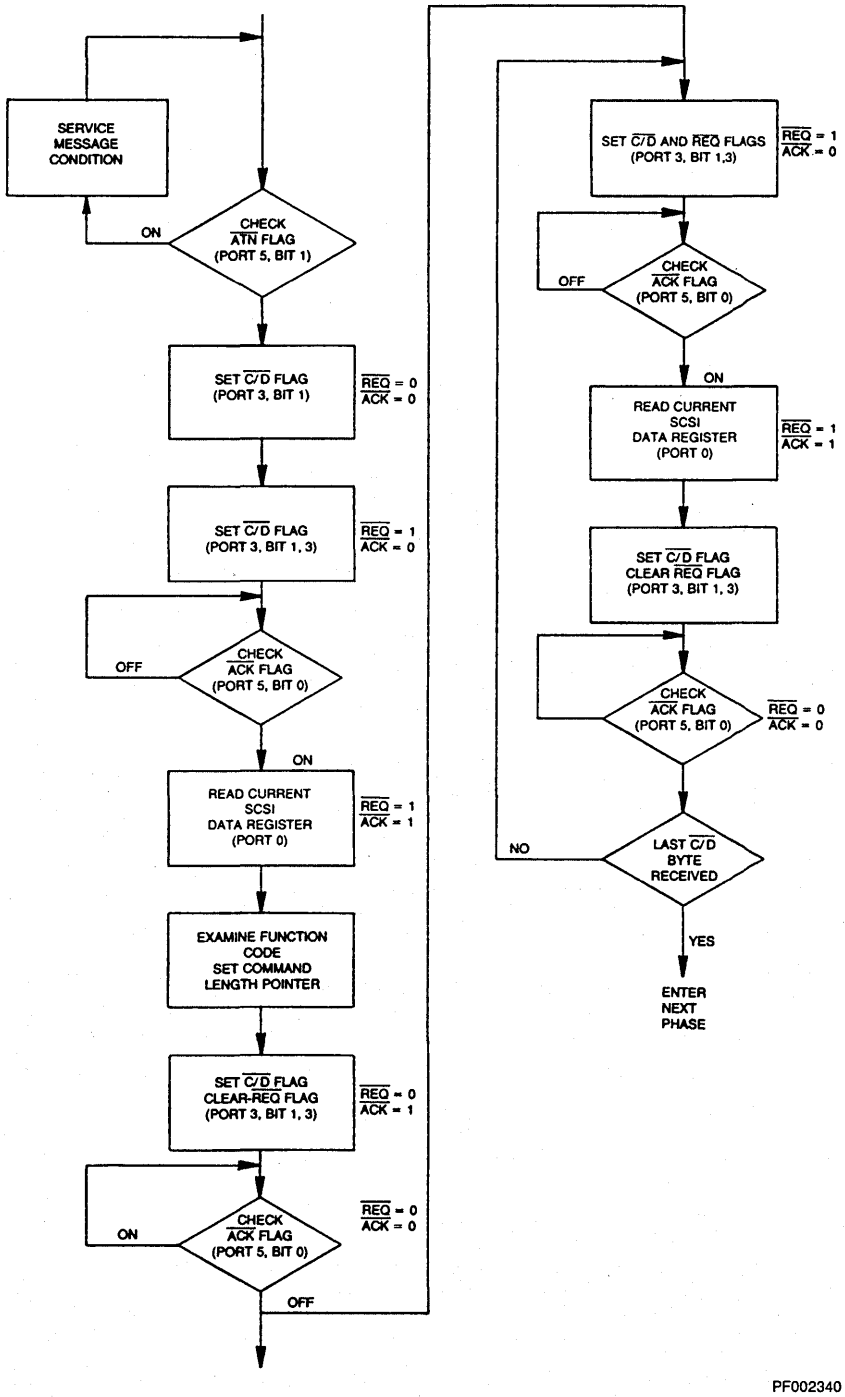


Figure 23. Command Transfer Phase (Target)

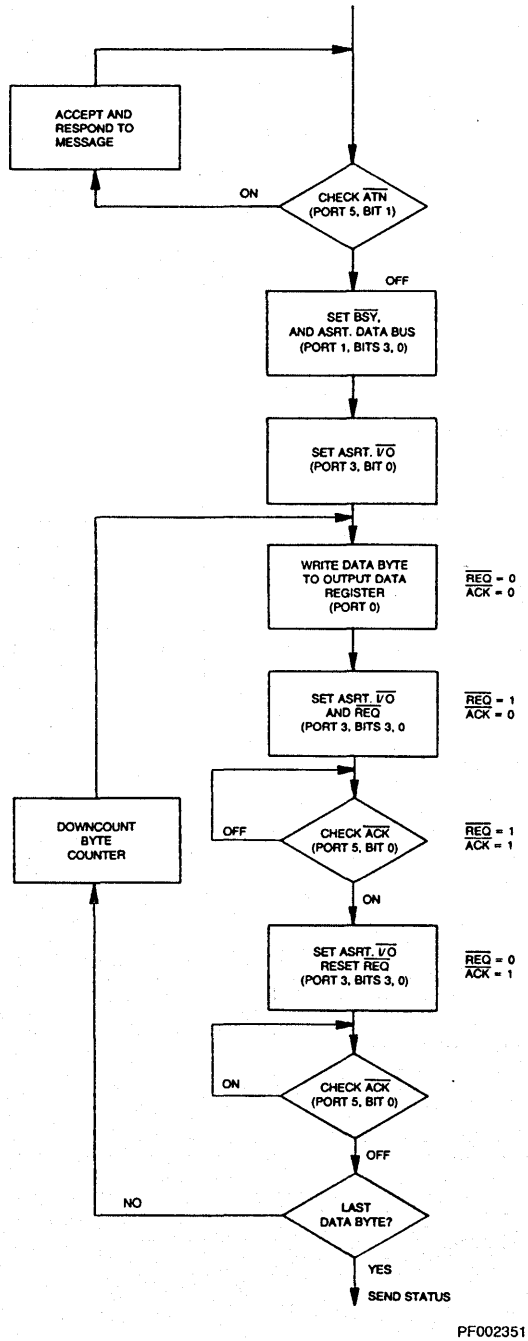


Figure 24. Data Transfer to Host via Programmed I/O

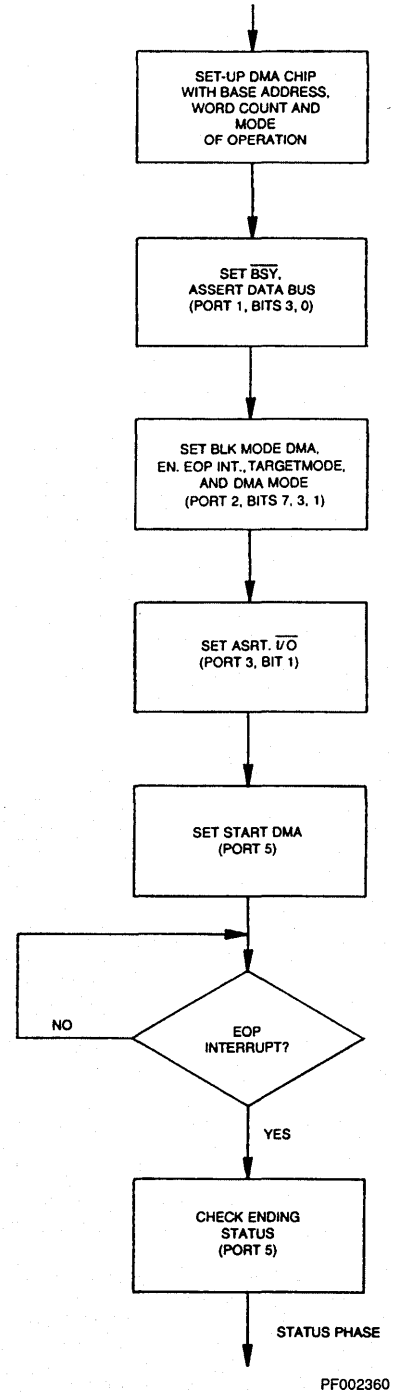


Figure 25. Data Transfer via DMA

READ **WRITE**

CURRENT SCSI DATA (00)

7 6 5 4 3 2 1 0

$\overline{DB_7}$ ... $\overline{DB_0}$

INITIATOR COMMAND REGISTER (01)

7 6 5 4 3 2 1 0

Assert Data Bus
Assert $\overline{ATN}$
Assert $\overline{SEL}$
Assert $\overline{BSY}$
Assert $\overline{ACK}$
Lost Arbitration
Arbitration in Progress
Assert $\overline{RST}$

MODE REGISTER (02)

7 6 5 4 3 2 1 0

Arbitration
DMA Mode
Monitor $\overline{BSY}$
Enable $\overline{EOP}$ interrupt
Enable Parity Interrupt
Enable Parity Checking
Target Mode
Block Mode DMA

TARGET COMMAND REGISTER (03)

7 6 5 4 3 2 1 0

0 0 0 0

Assert $\overline{I/O}$
Assert $\overline{C/D}$
Assert $\overline{MSG}$
Assert $\overline{REQ}$

CURRENT SCSI BUS STATUS (04)

7 6 5 4 3 2 1 0

$\overline{DBP}$
$\overline{SEL}$
$\overline{I/O}$
$\overline{C/D}$
$\overline{MSG}$
$\overline{REQ}$
$\overline{BSY}$
$\overline{RST}$

BUS & STATUS REGISTER (05)

7 6 5 4 3 2 1 0

$\overline{ACK}$
$\overline{ATN}$
Busy Error
Phase Match
Interrupt Request
Parity Error
DMA Request
End of DMA

INPUT DATA REGISTER (06)

7 6 5 4 3 2 1 0

$\overline{DB_7}$ ... $\overline{DB_0}$

RESET PARITY/INTERRUPT (07)

7 6 5 4 3 2 1 0

X X X X X X X X

DF006090

OUTPUT DATA REGISTER (00)

7 6 5 4 3 2 1 0

$\overline{DB_7}$ ... $\overline{DB_0}$

INITIATOR COMMAND REGISTER (01)

7 6 5 4 3 2 1 0

0

Assert Data Bus
Assert $\overline{ATN}$
Assert $\overline{SEL}$
Assert $\overline{BSY}$
Assert $\overline{ACK}$
Test Mode
Assert $\overline{RST}$

MODE REGISTER (02)

7 6 5 4 3 2 1 0

Arbitration
DMA Mode
Monitor $\overline{BSY}$
Enable $\overline{EOP}$ interrupt
Enable Parity Interrupt
Enable Parity Checking
Target Mode
Block Mode DMA

TARGET COMMAND REGISTER (03)

7 6 5 4 3 2 1 0

X X X X

Assert $\overline{I/O}$
Assert $\overline{C/D}$
Assert $\overline{MSG}$
Assert $\overline{REQ}$

SELECT ENABLE REGISTER (04)

7 6 5 4 3 2 1 0

$\overline{DB_7}$ ... $\overline{DB_0}$

START DMA SEND (05)

7 6 5 4 3 2 1 0

X X X X X X X X

START DMA TARGET RECEIVE (06)

7 6 5 4 3 2 1 0

X X X X X X X X

START DMA INITIATOR RECEIVE (07)

X X X X X X X X

NOTE: X = DON'T CARE

DF006100

Figure 26. Register Reference Chart

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Supply Voltage on Any Pin
with Respect to Ground ..... −0.5 to +7.0 V
Power Dissipation:
Am5380 ..... 0.8 W
Am53C80N ..... 0.2 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V

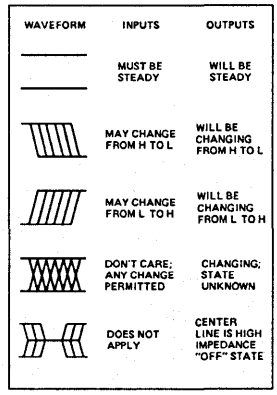
Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (Note 1)

Parameter Description	Test Conditions		Min.	Max.	Unit
INPUT SIGNAL REQUIREMENTS					
HIGH-Level Input (V_{IH})			2.0	5.25	V
LOW-Level Input (V_{IL})			−0.3	0.8	V
HIGH-Level Input Current (I_{IH}):					
Am5380 = SCSI Bus Pins	V_{IH} = 5.25 V, V_{IL} = 0				µA
Am53C80N = SCSI Bus Pins except $\overline{RST}$				50	
All Other Pins				10	
LOW-Level Input Current (I_{IL}):					
Am5380 = SCSI Bus Pins	V_{IH} = 5.25 V, V_{IL} = 0				µA
Am53C80N = SCSI Bus Pins except $\overline{RST}$				−50	
All Other Pins				−10	
OUTPUT SIGNAL REQUIREMENTS					
HIGH-Level Output Voltage (V_{OH}):					
Am5380 = All Pins	V_{DD} = 4.75 V,		2.4		V
Am53C80N = All Pins except SCSI Bus	I_{OH} = −3.0 mA				
LOW-Level Output Voltage (V_{OL}):					
SCSI Bus Pins	V_{DD} = 4.75 V, I_{OL} = 48.0 mA				
All Other Pins	V_{DD} = 4.75 V, I_{OL} = 7.0 mA			0.5	V
Power Supply Current (I_{DD})	V_{CC} = Max.	Am5380		145	mA
		Am53C80N		−35	

Notes: 1. Information for the Am53C80N is Preliminary and Subject to Change.

KEY TO SWITCHING WAVEFORMS



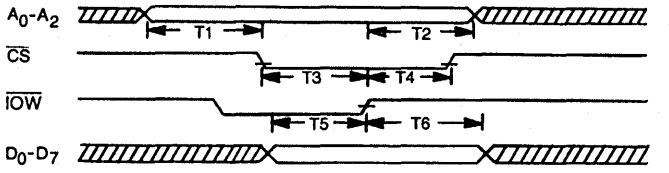
KS000010

SWITCHING CHARACTERISTICS/WAVEFORMS (Cont'd.)

CPU Write Cycle

Name	Description	Min.	Max.	Unit
T1	Address Setup to Write Enable*	10		ns
T2	Address Hold from End Write Enable*	0		ns
T3	Write Enable Width*	40		ns
T4	Chip Select Hold from End of $\overline{IOW}$	0		ns
T5	Data Setup to End of Write Enable*	20		ns
T6	Data Hold Time from End of $\overline{IOW}$	30		ns

*Write Enable is the occurrence of $\overline{IOW}$ and $\overline{CS}$

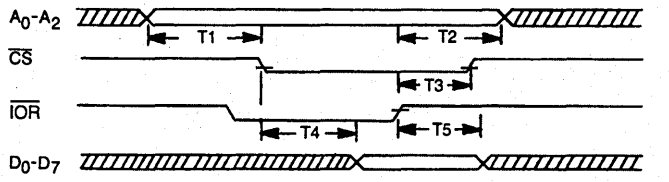


WF022360

CPU Read Cycle

Name	Description	Min.	Max.	Unit
T1	Address Setup to Read Enable*	10		ns
T2	Address Hold from End Read Enable*	0		ns
T3	Chip Select Hold from End of $\overline{IOR}$	0		ns
T4	Data Access Time from Read Enable*		100	ns
T5	Data Hold Time from End of $\overline{IOR}$	0		ns

*Read Enable is the occurrence of $\overline{IOR}$ and $\overline{CS}$



WF022370

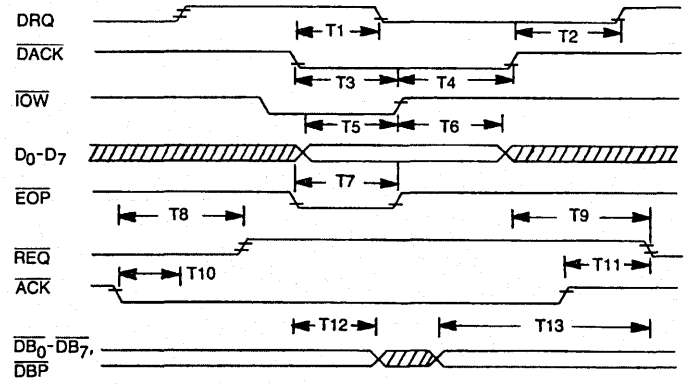
SWITCHING CHARACTERISTICS/WAVEFORMS (Cont'd.)

DMA Write (Non-Block Mode) Target Send Cycle

Name	Description	Min.	Max.	Unit
T1	DRQ FALSE from $\overline{DACK}$ TRUE		100	ns
T2	$\overline{DACK}$ FALSE to DRQ TRUE	30		ns
T3	Write Enable Width*	70		ns
T4	DACK Hold from End of $\overline{IOW}$	0		ns
T5	Data Setup to End of Write Enable*	30		ns
T6	Data Hold Time from End of $\overline{IOW}$	40		ns
T7	Width of $\overline{EOP}$ Pulse (Note 1)	70		ns
T8	$\overline{ACK}$ TRUE to $\overline{REQ}$ FALSE		125	ns
T9	$\overline{REQ}$ from End of $\overline{DACK}$ ($\overline{ACK}$ FALSE)		120	ns
T10	$\overline{ACK}$ TRUE to DRQ TRUE (Target)		110	ns
T11	$\overline{REQ}$ from End of $\overline{ACK}$ ($\overline{DACK}$ FALSE)		120	ns
T12	Data Hold from Write Enable	0		ns
T13	Data Setup to $\overline{REQ}$ TRUE (Target)	60		ns

*Write Enable is the occurrence of $\overline{IOW}$ and $\overline{DACK}$

Notes: 1. $\overline{EOP}$, $\overline{IOW}$, and $\overline{DACK}$ must be concurrently TRUE for at least T7 for proper recognition of the $\overline{EOP}$ pulse.



WF022380

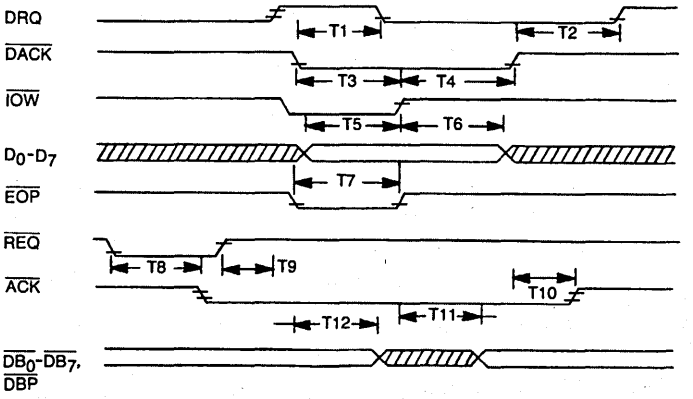
SWITCHING CHARACTERISTICS/WAVEFORMS (Cont'd.)

DMA Write (Non-Block Mode) Initiator Send Cycle

Name	Description	Min.	Max.	Unit
T1	DRQ FALSE from $\overline{DACK}$ TRUE		100	ns
T2	$\overline{DACK}$ FALSE to DRQ TRUE	30		ns
T3	Write Enable Width*	70		ns
T4	$\overline{DACK}$ Hold from End of $\overline{IOW}$	0		ns
T5	Data Setup to End of Write Enable*	30		ns
T6	Data Hold Time from End of $\overline{IOW}$	40		ns
T7	Width of $\overline{EOP}$ Pulse (Note 1)	70		ns
T8	$\overline{REQ}$ TRUE to $\overline{ACK}$ TRUE		110	ns
T9	$\overline{REQ}$ FALSE to DRQ TRUE		110	ns
T10	$\overline{DACK}$ FALSE to $\overline{ACK}$ FALSE		130	ns
T11	$\overline{IOW}$ FALSE to Valid SCSI Data		100	ns
T12	Data Hold from Write Enable	0		ns

*Write Enable is the occurrence of $\overline{IOW}$ and $\overline{DACK}$

Notes: 1. $\overline{EOP}$, $\overline{IOW}$, and $\overline{DACK}$ must be concurrently TRUE for at least T7 for proper recognition of the $\overline{EOP}$ pulse.



WF022390

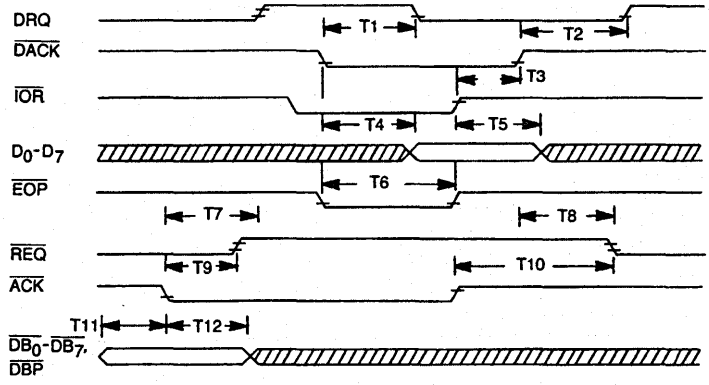
SWITCHING CHARACTERISTICS/WAVEFORMS (Cont'd.)

DMA Read (Non-Block Mode) Target Receive Cycle

Name	Description	Min.	Max.	Unit
T1	DRQ FALSE from $\overline{DACK}$ TRUE		100	ns
T2	$\overline{DACK}$ FALSE to DRQ TRUE	30		ns
T3	$\overline{DACK}$ Hold Time from End of $\overline{IOR}$	0		ns
T4	Data Access Time from Read Enable*		100	ns
T5	Data Hold Time from End of $\overline{IOR}$			ns
T6	Width of $\overline{EOP}$ Pulse (Note 1)	70		ns
T7	$\overline{ACK}$ TRUE to DRQ TRUE		110	ns
T8	$\overline{DACK}$ FALSE to $\overline{REQ}$ TRUE ($\overline{ACK}$ FALSE)		120	ns
T9	$\overline{ACK}$ TRUE to $\overline{REQ}$ FALSE		125	ns
T10	$\overline{ACK}$ FALSE to $\overline{REQ}$ TRUE ($\overline{DACK}$ FALSE)		120	ns
T11	Data Setup Time to $\overline{ACK}$	20		ns
T12	Data Hold Time from $\overline{ACK}$	50		ns

*Read Enable is the occurrence of $\overline{IOR}$ and $\overline{DACK}$

Notes: 1. $\overline{EOP}$, $\overline{IOR}$, and $\overline{DACK}$ must be concurrently TRUE for at least T6 for proper recognition of the $\overline{EOP}$ pulse.



WF022400

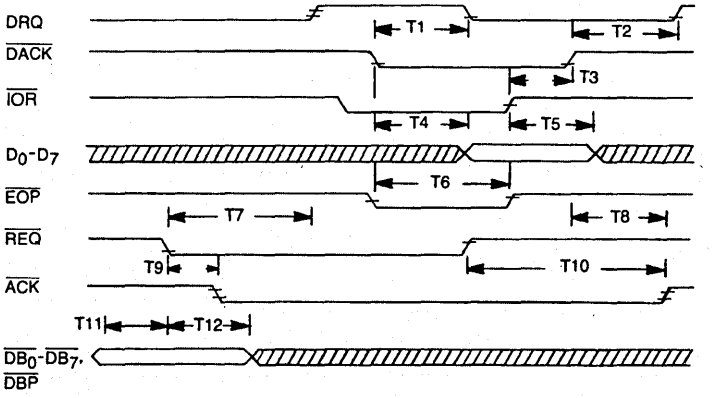
SWITCHING CHARACTERISTICS/WAVEFORMS (Cont'd.)

DMA Read (Non-Block Mode) Initiator Receive Cycle

Name	Description	Min.	Max.	Unit
T1	DRQ FALSE from $\overline{DACK}$ TRUE		100	ns
T2	$\overline{DACK}$ FALSE to DRQ TRUE	30		ns
T3	$\overline{DACK}$ Hold Time from End of $\overline{IOR}$	0		ns
T4	Data Access Time from Read Enable*		100	ns
T5	Data Hold Time from End of $\overline{IOR}$	0		ns
T6	Width of $\overline{EOP}$ Pulse (Note 1)	70		ns
T7	$\overline{REQ}$ TRUE to DRQ TRUE		140	ns
T8	$\overline{DACK}$ FALSE to $\overline{ACK}$ FALSE ($\overline{REQ}$ FALSE)		100	ns
T9	$\overline{REQ}$ TRUE to $\overline{ACK}$ TRUE		110	ns
T10	$\overline{REQ}$ FALSE to $\overline{ACK}$ FALSE ($\overline{DACK}$ FALSE)		100	ns
T11	Data Setup Time to $\overline{REQ}$	20		ns
T12	Data Hold Time from $\overline{REQ}$	50		ns

*Read Enable is the occurrence of $\overline{IOR}$ and $\overline{DACK}$

Notes: 1. $\overline{EOP}$, $\overline{IOR}$, and $\overline{DACK}$ must be concurrently TRUE for at least T6 for proper recognition of the $\overline{EOP}$ pulse.



WF022410

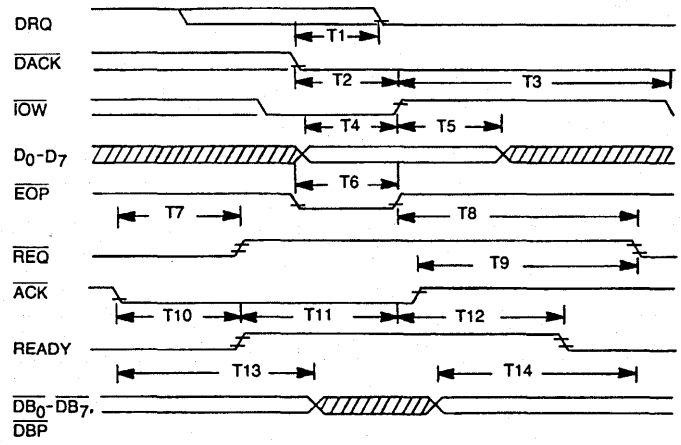
SWITCHING CHARACTERISTICS/WAVEFORMS (Cont'd.)

DMA Write (Block Mode) Target Send Cycle

Name	Description	Min.	Max.	Unit
T1	DRQ FALSE from $\overline{DACK}$ TRUE		100	ns
T2	Write Enable Width*	70		ns
T3	Write Recovery Time	120		ns
T4	Data Setup to End of Write Enable*	30		ns
T5	Data Hold Time from End of $\overline{IOW}$	40		ns
T6	Width of $\overline{EOP}$ Pulse (Note 1)	70		ns
T7	$\overline{ACK}$ TRUE to $\overline{REQ}$ FALSE		120	ns
T8	$\overline{REQ}$ from End of $\overline{IOW}$ ($\overline{ACK}$ FALSE)		130	ns
T9	$\overline{REQ}$ from End of $\overline{ACK}$ ($\overline{IOW}$ FALSE)		110	ns
T10	$\overline{ACK}$ TRUE to READY TRUE		140	ns
T11	READY TRUE to $\overline{IOW}$ FALSE	70		ns
T12	$\overline{IOW}$ FALSE to READY FALSE		120	ns
T13	Data Hold from $\overline{ACK}$ TRUE	0		ns
T14	Data Setup to $\overline{REQ}$ TRUE	60		ns

*Write Enable is the occurrence of $\overline{IOW}$ and $\overline{DACK}$

Notes: 1. $\overline{EOP}$, $\overline{IOW}$, and $\overline{DACK}$ must be concurrently TRUE for at least T6 for proper recognition of the $\overline{EOP}$ pulse.



WF022420

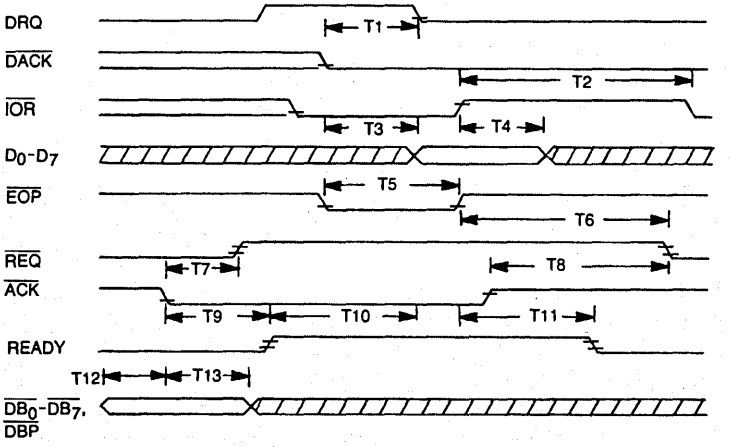
SWITCHING CHARACTERISTICS/WAVEFORMS (Cont'd.)

DMA Read (Block Mode) Target Receive Cycle

Name	Description	Min.	Max.	Unit
T1	DRQ FALSE from $\overline{DACK}$ TRUE		100	ns
T2	$\overline{IOR}$ Recovery Time	120		ns
T3	Data Access Time from Read Enable*		100	ns
T4	Data Hold Time from End of $\overline{IOR}$	0		ns
T5	Width of $\overline{EOP}$ Pulse (Note 1)	70		ns
T6	$\overline{IOR}$ FALSE to $\overline{REQ}$ TRUE ($\overline{ACK}$ FALSE)		130	ns
T7	$\overline{ACK}$ TRUE to $\overline{REQ}$ FALSE		125	ns
T8	$\overline{ACK}$ FALSE to $\overline{REQ}$ TRUE ($\overline{IOR}$ FALSE)		110	ns
T9	$\overline{ACK}$ TRUE to READY TRUE		140	ns
T10	READY TRUE to Valid Data		50	ns
T11	$\overline{IOR}$ FALSE to READY FALSE		120	ns
T12	Data Setup Time to $\overline{ACK}$	20		ns
T13	Data Hold Time from $\overline{ACK}$	50		ns

*Read Enable is the occurrence of $\overline{IOR}$ and $\overline{DACK}$

Notes: 1. $\overline{EOP}$, $\overline{IOR}$, and $\overline{DACK}$ must be concurrently TRUE for at least T5 for proper recognition of the $\overline{EOP}$ pulse.

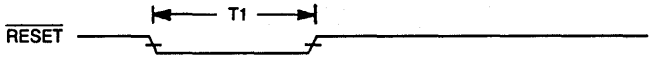


WF022430

SWITCHING CHARACTERISTICS/WAVEFORMS

Reset

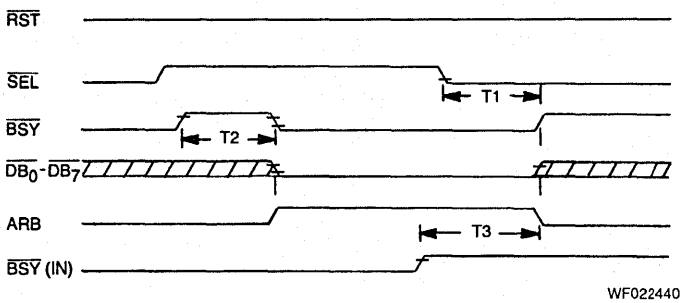
Name	Description	Min.	Max.	Unit
T1	Minimum Width of Reset	100		ns



WF022450

Arbitration

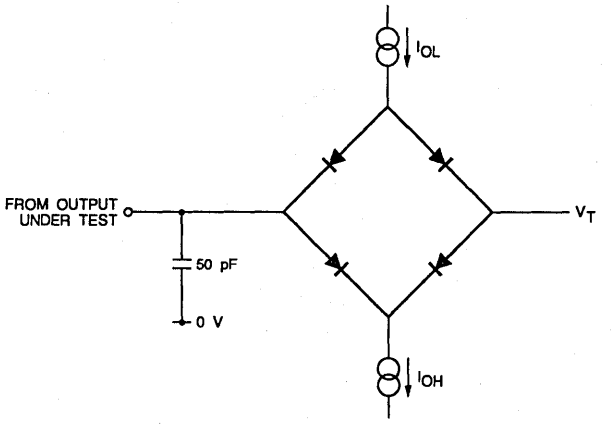
Name	Description	Min.	Max.	Unit
T1	Bus Clear from $\overline{SEL}$ TRUE		600	ns
T2	Arbitrate Start from $\overline{BSY}$ FALSE	1200	2200	ns
T3	Bus Clear from $\overline{BSY}$ FALSE		1100	ns



WF022440

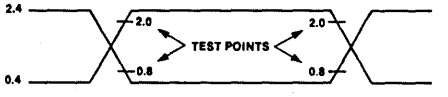
4

SWITCHING TEST CIRCUIT



TC003860

SWITCHING TEST WAVEFORM



WF009541

APPENDIX A — DESIGN MODIFICATIONS IN Am53C80N

Spurious RST Interrupts

If Am5380 is not terminated on the SCSI Interface, the floating input of the $\overline{RST}$ signal can generate spurious interrupts.

Am53C80N has 35 μA pull up on the $\overline{RST}$ signal which prevents the spurious interrupts caused by an unterminated SCSI Bus Interface.

The End of DMA for Send Operations

While sending the data to Am5380, if $\overline{EOP}$ is asserted on the last byte, the End of DMA Status Bit indicates that the last byte has been received from the DMA device; there is no indication that the last byte has been transferred to the SCSI Bus.

The Am53C80N uses Bit 7 of the Target Command Register to indicate that the last byte has been transferred to the SCSI Bus.

Faster $\overline{REQ}/\overline{ACK}$ Transition Times

The Am53C80N has faster $\overline{REQ}/\overline{ACK}$ handshake to improve overall data transfer rates.

Prevents the Possibility of an Additional $\overline{ACK}$ from Occurring

The Am5380, upon receipt of an $\overline{EOP}$ signal, sets the End of DMA Status Bit and prevents additional DMA requests; it does not reset the DMA Mode Bit. If receiving data as an initiator and the target continues to request data for the same bus phase after receiving an $\overline{EOP}$ pulse, the Am5380 will assert $\overline{ACK}$ without issuing DRQ.

The Am53C80N prevents $\overline{ACK}$ from being asserted until the device is instructed to continue by writing the Start DMA Initiator Receive Register.

Am5380/Am53C80N ERRATA

1) Edge triggered $\overline{RST}$ Interrupt — If the SCSI Bus is not terminated, the $\overline{RST}$ interrupt is continually generated.

2) TRUE End of DMA Interrupt — The Am5380/Am53C80N generates an interrupt when it receives the last byte from the DMA, not when the last byte is transferred to the SCSI Bus.

3) Return to READY after $\overline{EOP}$ Interrupt — When operating in Block mode DMA, the Am5380/Am53C80N does not return the READY signal to a Ready condition. This locks up the bus and prevents the CPU from executing.

4) SCSI handshake clean up after $\overline{EOP}$ Interrupt — Currently the $\overline{ACK}$ remains active after the $\overline{EOP}$ Interrupt is generated and must be turned off for the Send operations.

5) SCSI handshake after $\overline{EOP}$ occurs — If an $\overline{EOP}$ occurs when receiving data, a subsequent $\overline{REQ}$ will cause $\overline{ACK}$ to be asserted even though no DRQ is issued.

6) During Reselection, if the Target Command Register does not reflect the current bus phases (most likely Data Out), the Reselection interrupt may get reset.

7) A phase-mismatch interrupt is not guaranteed after a Reselection for the following reasons:

- DMA MODE bit must be set in order to receive a phase-mismatch interrupt
- DMA MODE bit cannot be set unless $\overline{BSY}$ is active
- $\overline{BSY}$ cannot be asserted until after the Reselection has occurred
- Once $\overline{BSY}$ is asserted, the Target may assert $\overline{REQ}$ in less than 500 ns
- The phase-mismatch interrupt is generated on the active edge of $\overline{REQ}$. If the DMA MODE bit is not set before the $\overline{REQ}$ goes active, the phase-mismatch interrupt will not occur

Preliminary

Am33C93A

Enhanced SCSI-Bus Interface Controller

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Implements full SCSI bus features: arbitration, disconnect, reconnect, parity generation/checking on both data ports, soft reset, and synchronous data transfers.
- Synchronous offset selectable from 1 to 12 bytes, with selectable transfer period up to 5 Mbytes/s.
- Compatible with most microprocessors through an 8-bit data bus; supports both multiplexed and non-multiplexed address/data bus systems. Host bus data parity checking and generation is an optional feature.
- Can be used as a host adapter (SCSI Initiator) or peripheral adapter (SCSI Target).
- Data transfer options include programmed I/O, single-byte DMA, burst (multibyte) DMA, or direct bus access (DBA Bus) transfers.
- Includes 48-mA drivers for direct connection to the SCSI bus.
- Burst data transfers up to 4096 bytes.
- Programmable timeout for selection and reselection.
- "Combination" commands greatly reduce interrupt-handling responsibilities.
- Special "Translate Address" command performs the Logical- to-Physical address translation.
- Single +5 V supply.
- Available in 44-pin chip carrier or 40-pin DIP.
- Low power CMOS design.

GENERAL DESCRIPTION

The 33C93A is a MOS/VLSI device implemented in Advanced Micro Devices' CMOS process. It operates from a single 5-Volt supply and is available in either a 44-pin chip carrier or a 40-pin dual-in-line package. All inputs and outputs are TTL compatible.

The 33C93A is intended for use in systems which interface to the Small Computer System Interface (SCSI) Bus. The 33C93A can operate in both the initiator (typically, a host computer system) and the target (typically, a peripheral device) SCSI bus roles.

When used in the host system, the 33C93A interfaces to both the host bus and the SCSI bus. To perform a SCSI operation, the host processor issues a command to the 33C93A to select the desired Target. The 33C93A then arbitrates for the SCSI bus and selects the peripheral unit. If it fails to get the bus because of a device with higher priority, it continues trying and notifies the host when it has succeeded by generating an interrupt. At this point, the 33C93A is operating in the initiator role. When the peripheral requests a SCSI command from the host, the 33C93A receives the request and generates another interrupt to the host. The host responds to this interrupt by issuing a "Transfer Info" command and supplying SCSI command bytes to the 33C93A. The 33C93A transfers the SCSI command to the peripheral, and then waits for the next bus phase request. This process continues until all SCSI information including data, status, and messages have been transferred.

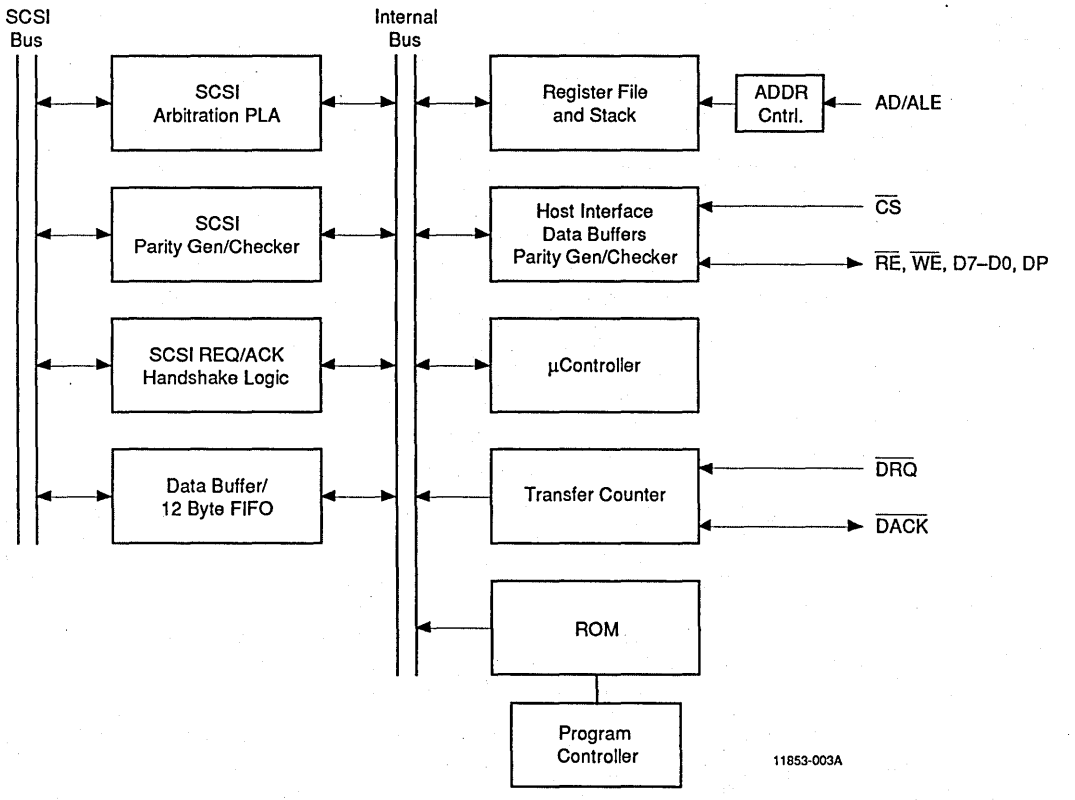
The 33C93A also offers high-level Select-and-Transfer commands which eliminate the interrupt handling otherwise required between each SCSI bus phase.

When the 33C93A is used in a peripheral system, the 33C93A will operate primarily in a Target role. It interfaces with a local processor and the SCSI bus in this environment just as it does when used as a host adapter. The Target-role command set enables the 33C93A to request each SCSI bus phase individually or to sequence the SCSI bus phases automatically through the use of combination commands.

The 33C93A has an internal microcontroller, a register task file, and SCSI interface logic. This architecture supports both tight control of the protocol for non-standard SCSI implementations, as well as a hands-free mode for standard SCSI applications.

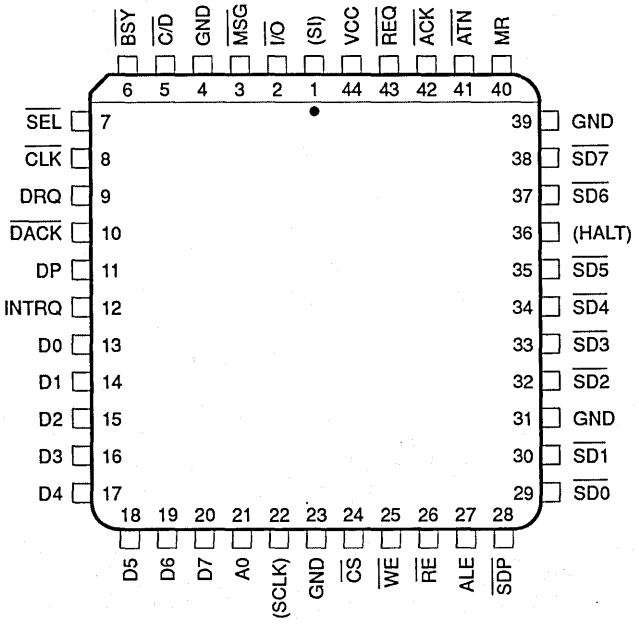
Publication #11853 Rev. A Amendment /0
Issue Date: June 1989

BLOCK DIAGRAM



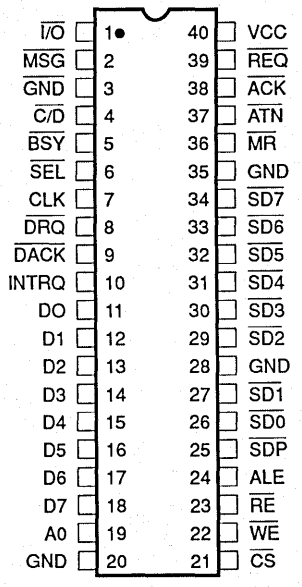
CONNECTION DIAGRAMS

44-PIN CHIP CARRIER



Note: Pins in parentheses are for test purposes only, and should be left unconnected for normal chip operation.

40-PIN DIP

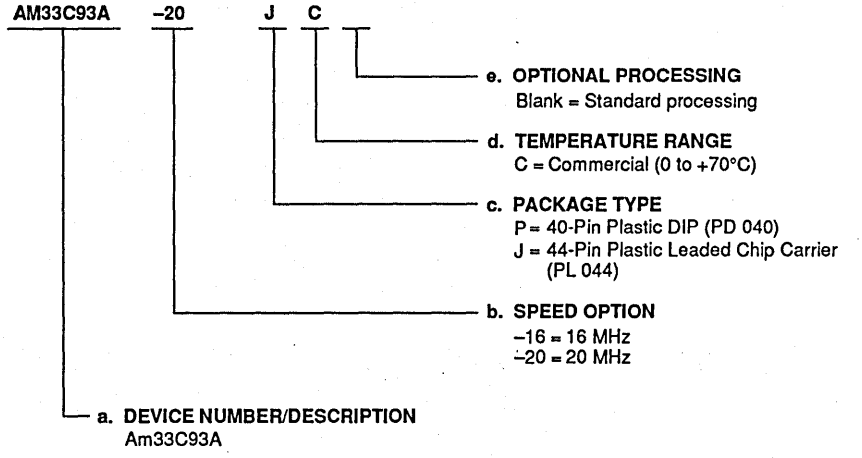


ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. **Device Number**
b. **Speed Option** (if applicable)
c. **Package Type**
d. **Temperature Range**
e. **Optional Processing**



Valid Combinations	
AM33C93A-16	JC, PC
AM33C93A-20	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTIONS

Processor/DMA Interface

Name	I/O	Function
CLK	I	8-20 MHz square wave clock.
$\overline{MR}$	I	Reset is an active-low input which forces the 33C93A into an idle state. All SCSI signals are forced to the negated state.
INTRQ	O	Interrupt Request to external microprocessor indicates a command completion/termination or a need to service the SCSI interface. Reading the SCSI STATUS register clears this bit.
$\overline{RE}$	I/O (TRI-STATE)	Read Enable is an active-low input which is used with $\overline{CS}$ to read a register or with $\overline{DACK}$ to access the DATA register in DMA mode. In DBA Bus mode, it is used as an output to read data from a sector buffer.
$\overline{WE}$	I/O (TRI-STATE)	Write Enable is an active-low input which is used with $\overline{CS}$ to write a register or with $\overline{DACK}$ to access the DATA register in DMA mode. In DBA Bus mode, it is used as an output to write data to a sector buffer.
$\overline{CS}$	I	Chip Select is an active-low input which qualifies $\overline{RE}$ and $\overline{WE}$ when accessing a register. This signal must be inactive during a DMA cycle ($\overline{DACK}$ active in DMA and Burst DMA mode, or DRQ active in DBA Bus mode).
A0	I	Address pin A0 is used to access the internal registers for non-multiplexed address/data bus (i.e. the ALE pin is grounded). The address of the desired register is loaded into the ADDRESS register during a write cycle with A0=0. The selected register is then accessed when A0=1.
ALE	I	Address Latch Enable is used for multiplexed address/data bus to load the address of the desired 33C93A register from the data bus. If indirect addressing is to be used, the ALE pin should be grounded. See the description of the ADDRESS register for a complete discussion of direct and indirect addressing.
$\overline{DACK}$ ($\overline{RCS}$)	I/O (OPENDRAIN)	DMA acknowledge input used for interfacing to an external DMA controller (e.g. 8237). When $\overline{DACK}$ is low, all bus transfers are to/from the DATA register regardless of the contents of the ADDRESS register. In DBA Bus mode, this pin functions as a RAM chip select output to allow the 33C93A to access a sector buffer. $\overline{RE}$ and $\overline{WE}$ are outputs when $\overline{RCS}$ (RAM Chip Select) is active. Since this pin can be an open drain output, a pullup resistor may be required when operating in DBA Bus mode.
$\overline{DRQ}$ (DRQ)	I/O (OPENDRAIN)	Data request is an output when interfacing to an external DMA controller, and an input when in AM Bus mode. When used with an external DMA controller, $\overline{DRQ}$ and $\overline{DACK}$ form the handshake for the data-byte transfers. In Burst mode, $\overline{DRQ}$ remains low as long as there is data to transfer. In AM Bus mode, the 33C93A performs burst transfers while DRQ is high, and when DRQ is low, data transfers are inhibited, $\overline{RCS}$ is false, and the $\overline{RE}$ and $\overline{WE}$ outputs are disabled. Since this pin can be an open drain output, a pullup resistor may be required when operating in DMA or Burst mode.
D7–D0	I/O	Processor data bus.
DP	I/O	Data Parity, used only for checking/generating parity during data transfers.

SCSI Interface

Name	I/O	Function
$\overline{ATN}$	I/O	$\overline{ATN}$ is an output in the initiator role and an input in the target role. It is used to indicate the ATTENTION condition.
$\overline{ACK}$	I/O	$\overline{ACK}$ is an output in the initiator role and an input in the target role. It is used to indicate an acknowledgement for a REQ/ACK data transfer handshake.
$\overline{MSG}$	I/O	$\overline{MSG}$ is an input in the initiator role and an output in the target role. It is asserted during a MESSAGE phase.
C/$\overline{D}$	I/O	C/$\overline{D}$ is an input in the initiator role and an output in the target role. It is used to indicate whether CONTROL or DATA information is on the SCSI data bus.
$\overline{REQ}$	I/O	$\overline{REQ}$ is an input in the initiator role and an output in the target role. It indicates a request for a REQ/ACK data transfer.

SCSI Interface (Cont.)

Name	I/O	Function
$\overline{I/O}$	I/O	$\overline{I/O}$ is an input in the initiator role and an output in the target role. It controls the direction of data movement on the SCSI data bus with respect to an Initiator.
$\overline{SD0–SD7}$	I/O	SCSI data bus.
$\overline{SDP}$	I/O	SCSI data bus parity signal.
$\overline{BSY}$	I/O	$\overline{BSY}$ is asserted when the 33C93A is attempting to arbitrate for the SCSI bus or when connected as a Target.
$\overline{SEL}$	I/O	$\overline{SEL}$ is asserted when the 33C93A is attempting to select or reselect another SCSI device.

Note: All pins have open-drain output drivers.

Am33C93A REGISTERS

Register MAP

A0	R/W	Register Accessed		Address (HEX)
0	R	AUXILIARY STATUS REGISTER		XX
0	W	ADDRESS REGISTER		XX
1	R/W	OWN ID REGISTER	/CDB SIZE	00
1	R/W	CONTROL REGISTER		01
1	R/W	TIMEOUT PERIOD REGISTER		02
1	R/W	TOTAL SECTORS REGISTER	/CDB 1ST	03
1	R/W	TOTAL HEADS REGISTER	/CDB 2ND	04
1	R/W	TOTAL CYLINDERS REGISTER (MSB)	/CDB 3RD	05
1	R/W	TOTAL CYLINDERS REGISTER (LSB)	/CDB 4TH	06
1	R/W	LOGICAL ADDRESS (MSB)	/CDB 5TH	07
1	R/W	LOGICAL ADDRESS (2ND)	/CDB 6TH	08
1	R/W	LOGICAL ADDRESS (3RD)	/CDB 7TH	09
1	R/W	LOGICAL ADDRESS (LSB)	/CDB 8TH	0A
1	R/W	SECTOR NUMBER REGISTER	/CDB 9TH	0B
1	R/W	HEAD NUMBER REGISTER	/CDB 10TH	0C
1	R/W	CYLINDER NUMBER (MSB) REGISTER	/CDB 11TH	0D
1	R/W	CYLINDER NUMBER (LSB) REGISTER	/CDB 12TH	0E
1	R/W	TARGET LUN REGISTER		0F
1	R/W	COMMAND PHASE REGISTER		10
1	R/W	SYNCHRONOUS TRANSFER REGISTER		11
1	R/W	TRANSFER COUNT REGISTER (MSB)		12
1	R/W	TRANSFER COUNT REGISTER (2ND BYTE)		13
1	R/W	TRANSFER COUNT REGISTER (LSB)		14
1	R/W	DESTINATION ID REGISTER		15
1	R/W	SOURCE ID REGISTER		16
1	R	SCSI STATUS		17
1	R/W	COMMAND REGISTER		18
1	R/W	DATA REGISTER		19

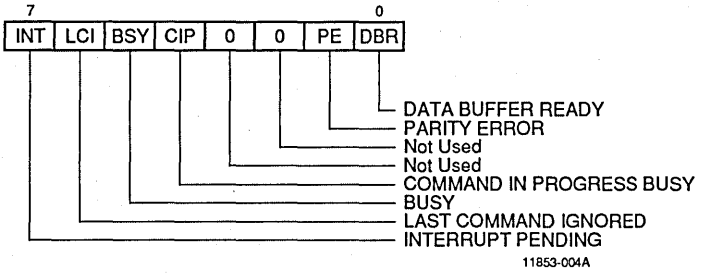
Notes:

1. *All unused bits of a defined register are reserved and must be zero.*
2. *Reading an undefined or unavailable register results in an all-ones data bus output.*
3. *Register addresses are determined by the ADDRESS register bits AR7 through AR0.*
4. *When using a multiplexed address/data bus with ALE, the A0 pin is ignored and the ADDRESS register is loaded with ALE. In this mode, the AUXILIARY STATUS register is mapped at hex 1F.*
5. *See Page 14 for a description of how reset affects the internal registers.*

Register Descriptions

AUXILIARY STATUS REGISTERS

The AUXILIARY STATUS register is a read-only register which contains general status information not directly associated with the interrupt condition. The AUXILIARY STATUS register may be accessed at any time, except during DMA accesses ($\overline{DACK}$ asserted in DMA/Burst mode or $\overline{DRQ}$ asserted in DBA bus mode).



Bit	Name	Description
0	DBR	DATA BUFFER READY is used during programmed I/O to indicate to the processor whether or not the DATA register is available for reading or writing. During Send or Transfer commands which transmit data over the SCSI bus, the DBR bit is set when the 33C93A is ready to take a byte from the host; the bit is reset when the processor writes the byte to the DATA register. During Receive or Transfer commands which receive data over the SCSI bus, the DBR is set when a byte is received; it is reset when the processor reads the byte from the DATA register.
1	PE	PARITY ERROR status indicates that even parity was detected on a data byte received during an information transfer. Parity is checked on data received from the host bus during transfers out to the SCSI bus and is checked on data received from the SCSI bus during transfers out to the host bus. Detection of a parity error will set the PE status bit regardless of the state of the HHP or HSP bits in the CONTROL register. The PE bit is cleared when a new command is issued.
4	CIP	COMMAND IN PROGRESS, when set, indicates that the 33C93A is interpreting the last command entered into the COMMAND register and therefore this register is unavailable. When this bit is reset, a command may be written to the COMMAND register.
5	BSY	BUSY indicates that a Level II command is currently executing and therefore only the COMMAND register (when CIP = 0), the DATA register, and the AUXILIARY STATUS register are accessible by the host. A Level II command may not be written to the COMMAND register when this bit is one.
6	LCI	LAST COMMAND IGNORED indicates that a command was issued by the host just prior to or concurrent with a pending interrupt, and therefore the command will be ignored.
7	INT	INTERRUPT PENDING indicates that the INTRQ pin is asserted. The host should read the SCSI STATUS register to clear INTRQ prior to issuing any commands.

ADDRESS REGISTER

The ADDRESS register is a write-only register which contains the address of the register to be accessed. Registers in the 33C93A may be accessed in one of two ways:

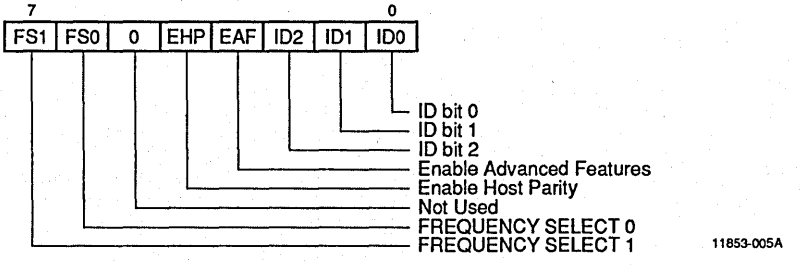
- Direct addressing (multiplexed address/data busses). In direct addressing, the falling edge of the ALE signal is used to latch the address into the ADDRESS register. The A0 pin should be connected to ground when using this method. The ALE is typically then followed by the $\overline{CS}$ and $\overline{WE}$ or $\overline{RE}$ signals that access the selected register. Also, in direct addressing, the AUXILIARY STATUS register is located at address 1F hex.
- Indirect addressing (separate address/data busses). In indirect addressing, the register access is performed in two separate cycles. This method is enabled by attaching ALE to ground. First, the ADDRESS register is loaded by performing a write of the desired address to the 33C93A ($\overline{WE}$ and $\overline{CS}$ asserted) with A0=0. Then the register is accessed by asserting $\overline{CS}$ and $\overline{WE}$ or $\overline{RE}$, with A0=1. Also, following every access with A0=1, the ADDRESS register will automatically increment to point at the next register, with the exception of the following locations: AUXILIARY STATUS register, DATA register, and the COMMAND register. In indirect addressing, the AUXILIARY STATUS register is accessed by performing a read ($\overline{CS}$ and $\overline{RE}$ asserted) with A0=0.

OWN ID/CDB SIZE REGISTER

The OWN ID/CDB SIZE register, in its first mode, contains both the encoded ID of the 33C93A on the SCSI bus and several control bits that are used to initially configure the device during the "Reset" command. These bits control 'advanced feature' selection, host bus parity enable, and selection of the divisor for the input clock. In its second mode (when advanced features are enabled, see p.16), this register is used during the combination commands to specify the SCSI CDB size if the command group is unknown to the 33C93A.

In the first mode, this register (as defined below) is sampled and becomes effective only after a "Reset" command is issued to the device. This register must be initialized, and a "Reset" command must then be issued. Doing this will set the SCSI bus ID, the clock divisor, and operating modes before any other commands are issued.

In the second mode, bits 3-0 of this register are used during the Select-and-Transfer and Wait-for-Select commands to specify the SCSI Command Descriptor Block size if it is not a group 0, group 1, or group 5 command. This mode is enabled only when advanced features are enabled (see p.16).



Bit	Name	Description
0-2	IDn	SCSI ID Bits 0-2 set the SCSI bus ID number that the 33C93A will use during arbitration and selection.
3	EAF	ENABLE ADVANCED FEATURES, when set to one, causes the 33C93A to enable certain advanced features (see Page 16). When this bit is zero, those features are disabled.
4	EHP	ENABLE HOST PARITY, when set to one, enables odd parity checking on the host bus; the PE bit in the AUXILIARY STATUS register will indicate parity errors detected on the host bus, and the HHP bit in the CONTROL register will be used. When this bit is zero, no checking is performed on the host bus; the PE bit is not set when a parity error is detected on the host bus, and the HHP bit must be set to zero. NOTE: Parity is always generated on the host data parity bit (DP), regardless of the state of this bit.

Bit	Name	Description
6–7	FSn	FREQUENCY SELECT 0–1 select the divisor that is applied to the input clock. The resulting clock is used for data transfer timing and for SCSI bus arbitration timing. The table below shows input clock frequency ranges and the corresponding divisors. The correct divisor for the input clock must be used, or SCSI bus timing specifications may not be met.

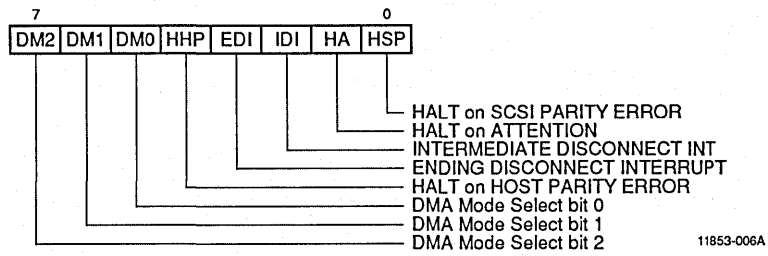
INPUT CLOCK FREQUENCY (MHZ)	FS1	FS0	RESULTING DIVISOR
8-10	0	0	2
12-15	0	1	3
16-20	1	0	4
xx	1	1	Undefined

Note that an 11 MHZ clock rate should not be used, as the resulting SCSI bus clear delay may violate SCSI specifications. The formula for computing the maximum SCSI data transfer rate is:

$$\text{Maximum SCSI Transfer Rate} = \frac{\text{Input Clock Fequency}}{\text{Clock Divisior}} \text{ [Mbyte/sec]}$$

CONTROL REGISTER

The CONTROL register is used to enable/disable certain functions, such as response to parity errors and the SCSI attention condition, interrupt handling, and data transfer modes.



Bit	Name	Description
0	HSP	The HALT on SCSI PARITY ERROR bit enables the 33C93A to immediately terminate a Receive or Transfer command if a parity error is detected on an incoming SCSI data byte. In the Initiator role, termination due to a SCSI parity error causes the $\overline{ACK}$ pin to be left in the active state in order to inhibit any additional data transfers (REQs) by the Target; this facilitates error handling with the Target. Synchronous data transfers check parity every 4096 bytes, or at the end of the remaining transfer count, whichever is less. Asynchronous transfers check parity on every byte.
1	HA	The HALT on ATTENTION bit (in Target mode only) enables the 33C93A to terminate a Send or Receive command if the $\overline{ATN}$ input is asserted. This normally indicates that the Initiator detected a parity error while receiving data from the 33C93A. The $\overline{ATN}$ input is tested before the start of a data transfer, every 4096 bytes if the transfer count is greater than 4096, and after the end of the transfer. These rules apply to both synchronous and asynchronous transfers.
2	IDI	The INTERMEDIATE DISCONNECT INTERRUPT bit, when set, enables the 33C93A to generate an 85H interrupt and complete a Select-and-Transfer command if the Target disconnects according to the defined SCSI protocol. When this bit is reset, no interrupt is generated by a valid disconnect. This feature, when used with the Resume SAT command, provides support for overlapped SCSI operations. IDI is also used to select execution options in Target mode Combination commands that serve to reduce host system overhead. Refer to COMMANDS, p.15 for more details.

Bit	Name	Description
3	EDI	When the ENDING DISCONNECT INTERRUPT bit is set, the 16H interrupt which normally follows the COMMAND COMPLETE message during the execution of a Select-and-Transfer command will be suppressed until the Target disconnects from the SCSI bus. EDI is also used in the Target mode Combination commands to enable chaining between those commands, resulting in reduced host system overhead. Refer to COMMANDS p.15 for more details.
4	HHP	The HALT on HOST PARITY ERROR bit enables the 33C93A to immediately terminate a Send or Transfer command if a parity error is detected on an incoming host data byte. Host parity errors are checked according to the rules for checking SCSI parity errors. However, a halt on a host parity error will not hold the $\overline{ACK}$ signal asserted when an error occurs. Host parity checking is performed at the same intervals as SCSI parity checking.
5-7	DMx	DMA MODE SELECT bits 2-0 are used to select the DMA mode of operation, which describes the host bus transfer mode used during Data In or Data Out phases. The following table describes the different DMA modes, and the state of these bits to select them:

DM2	DM1	DM0	DMA Mode Selected
0	0	0	POLLED MODE, or no DMA enabled. All data phase transfers are performed by polling for DBR in the AUXILIARY STATUS register, and then writing (reading) the data to (from) the DATA register.
0	0	1	BURST MODE selects a demand-mode DMA interface. In this mode, the $\overline{DRQ}$ signal will be active as long as there is data/space in the internal FIFO to allow the transfer to continue. The DMA controller responds by asserting $\overline{DACK}$ and $\overline{RE}$/$\overline{WE}$ as long as $\overline{DRQ}$ is active.
0	1	0	DBA BUS MODE is selected when the 33C93A is connected to a DBA Bus. This mode also can be called Direct Buffer Access (DBA) mode. In this mode, the 33C93A acts as a bus master, and all data access signals reverse their direction: The $\overline{DRQ}$ output signal becomes the DRQ input, which enables the 33C93A to drive the buffer bus control signals. The $\overline{DACK}$ output signal becomes the $\overline{RCS}$ input, which is asserted as a chip select for the buffer. The $\overline{RE}$ and $\overline{WE}$ inputs become outputs which drive the read and write functions of the RAM buffer. As long as the DRQ signal is asserted, transfers will continue in a burst manner, until the transfer is complete or it decides to pause the transfer by negating the $\overline{DRQ}$ signal; one more transfer may occur after this transition, and then the $\overline{DACK}$, $\overline{RE}$, and $\overline{WE}$ signals are negated.
1	0	0	DMA MODE is selected when the 33C93A is to be used with a DMA controller in single-byte transfer mode. In this mode, $\overline{DRQ}$ is asserted and then negated, and the DMA controller responds by asserting $\overline{DACK}$ and $\overline{WE}$ or $\overline{RE}$, for each data byte transferred to/from the 33C93A.

TIMEOUT PERIOD REGISTER

The TIMEOUT PERIOD register is an 8-bit register containing a preset value which determines the timeout period for Select and Reselect commands. This value may be calculated as a function of the input clock frequency and the desired timeout period, as shown in the following equation:

$$\text{register value} = \frac{\text{Tper} \cdot \text{Ficlk}}{80}$$

Where:

Tper = The desired timeout period in milliseconds;

Ficlk = The input clock frequency at the CLK pin in MHz (with no divisor applied).

The constant '80' scales the units of the equation, as is based on the internal timeout cycle time. The user should round the resulting 'register value' up to the next integral value to ensure that the user's minimum timeout requirement is met.

The timeout period specifies how long the 33C93A will wait for a response (indicated by assertion of the $\overline{BSY}$ signal) after it has begun the selection phase (assert $\overline{SEL}$ and negate $\overline{BSY}$) before terminating the command. The timeout function can be disabled by loading the TIMEOUT PERIOD register with zero.

NOTE: The following twelve registers are used exclusively by the Translate Address and/or "combination" commands. The function of each register is determined by the type of command issued.

TOTAL SECTORS REGISTER/CDB 1ST BYTE
Translate Address: The TOTAL SECTORS register should be set to the total number of sectors per track prior to issuing a Translate Address command.

Select-and-Transfer: This register should be loaded with the first byte of the COMMAND DESCRIPTOR BLOCK before issuing a Select-and-Transfer command.

Wait-for-Select-and-Receive: The 33C93A will store the first byte of the received CDB in this register.

TOTAL HEADS REGISTER/CDB 2ND BYTE
Translate Address: This register holds the total number of heads during a Translate Address command.

Select-and-Transfer: This register should be loaded with the second byte of the CDB before issuing a Select-and-Transfer command.

Wait-for-Select-and-Receive: The 33C93A will store the second byte of the received CDB in this register.

TOTAL CYLINDERS REGISTER/CDB 3RD AND 4TH BYTES
Translate Address: This is a 16-bit register which holds the total number of cylinders.

Select-and-Transfer: This register should be loaded with the third and fourth bytes of the CDB before issuing a Select-and-Transfer command.

Wait-for-Select-and-Receive: The 33C93A will store the third and fourth bytes of the received CDB in this register.

LOGICAL ADDRESS REGISTER/CDB 5TH–8TH BYTES
Translate Address: The LOGICAL ADDRESS register is a 32-bit register which should be loaded with the logical address to be translated prior to issuing the Translate Address command.

Select-and-Transfer: For six byte CDBs, only the first two bytes of this register are loaded with the fifth and sixth bytes of the CDB. For ten and twelve byte CDBs, this register is loaded with the fifth, sixth, seventh, and eighth bytes of the CDB.

Wait-for-Select-and-Receive: The 33C93A will store the fifth, sixth, seventh (if any), and eighth (if any) bytes of the received CDB in this register.

SECTOR NUMBER REGISTER/CDB 9TH BYTE
Translate Address: This register will contain the resulting sector number following a Translate Address command.

Select-and-Transfer: This register should be loaded with the ninth byte of a ten or twelve byte CDB before issuing a Select-and-Transfer command.

Wait-for-Select-and-Receive: The 33C93A will store the ninth byte of a ten or twelve byte received CDB in this register.

HEAD NUMBER REGISTER/CDB 10TH BYTE
Translate Address: The HEAD NUMBER register contains the resulting head number following a Translate Address command. If automatic compensation for spare sectors on a disk is to be performed by the Am33C93A, then the number of spare sectors per cylinder must be written into this register before issuing the Translate Address command. It should be noted that when compensation is used, the maximum number of cylinders allowed is 4096, and the maximum number of heads is 15. An initial value of zero in this register indicates that no compensation is to be performed.

Select-and-Transfer: This register should be loaded with the tenth byte of a ten or twelve byte CDB before issuing a Select-and-Transfer command.

Wait-for-Select-and-Receive: The 33C93A will store the tenth byte of a ten or twelve byte received CDB in this register.

CYLINDER NUMBER REGISTER/CDB 11TH AND 12TH BYTES
Translate Address: The CYLINDER NUMBER register is a 16-bit register which contains the resulting cylinder number following execution of the Translate Address command. When a Translate Address command involving automatic compensation for spare sectors is issued (i.e. the HEAD NUMBER register initially contains a nonzero value), then this register must be loaded with total number of sectors per cylinder (total sectors/track • total heads – total spare sectors/cyl) before issuing the command.

Select-and-Transfer: This register should be loaded with the eleventh and twelfth bytes of a twelve byte CDB before issuing a Select-and-Transfer command.

Wait-for-Select-and-Receive: The 33C93A will store the eleventh and twelfth bytes of a twelve byte received CDB in this register.

Send-Status-and-Command-Complete: The CDB11 register is used to specify the returned status byte to be sent during a Send-Status-and-Command-Complete command. The CDB12 register is used to determine the type of Command-Complete message sent by the 33C93A. If bit 0 of the CDB12 register is set to one, then a linked Command Complete message will be sent during command execution. In this case, bit 1 of the CDB12 register is used as a FLAG bit to determine whether a 0A hex (FLAG=0) or a 0B hex (FLAG=1) Linked Command Complete message is sent. If bit 0 is zero, then a simple Command Complete message (00 hex) is sent.

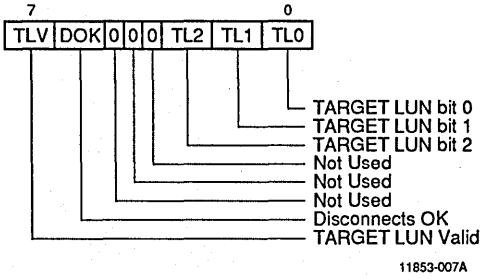
TARGET LUN REGISTER

The TARGET LUN register is used to hold both the Logical Unit Number (LUN) and Target status information during various 33C93A commands and sequences. During a Select-and-Transfer or Reselect-and-Transfer command, the contents of this register (along with the SOURCE ID register) are used to generate and check the IDENTIFY messages transferred across the SCSI bus. In addition, the TARGET LUN register is used to hold the Target Status byte received during a Select-and-Transfer command.

During Wait-for-Select-and-Receive commands, this register may hold the image of the Identify message received from the Initiator. If the TLV bit is zero, there was no Identify message received. If the TLV bit is one, then a valid Identify message was received. The DOK bit will then indicate whether of not the Initiator has enabled disconnects.

During Reselect-and-Transfer commands, this register is used to set the LUN to be used in the Identify message sent to the Initiator after Selection phase. The TLV and DOK bits are not used.

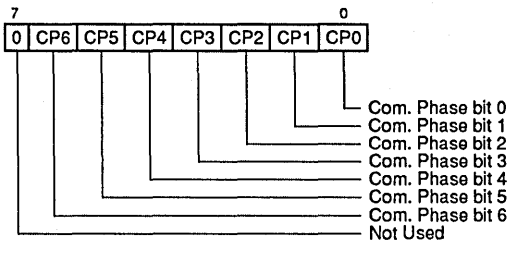
In advanced mode, during Select-and-Transfer commands, this register is used to handle reselection by an unexpected Target. In this case, this register will hold the logical unit number of the reselecting target. The TLV and DOK bits will be zero.



11853-007A

COMMAND PHASE REGISTER

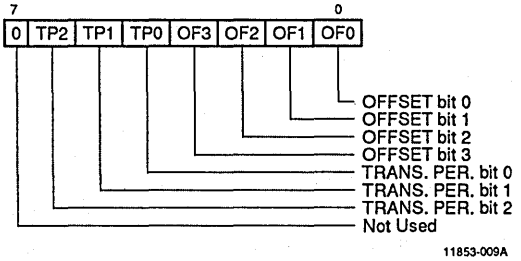
The COMMAND PHASE register is used during combination commands to indicate which phases of these multi-phase commands have been completed. Thus, if the command has terminated abnormally, the processor can read this register to determine the cause of the termination and how to respond to it. This register is also used to resume combination commands by loading this register with a value that indicates the next desired or expected bus phase, and reissuing the command. Refer to the description of the specific commands for details regarding the various command phases and resume values.



11853-008A

SYNCHRONOUS TRANSFER REGISTER

The SYNCHRONOUS TRANSFER register is used to select between synchronous and asynchronous transfers, and is also used to define the maximum transfer rate. For information phases other than a "data" transfer phase, or when the selected offset is zero (OF3=OF2=OF1=OF0=0), asynchronous transfers will occur. Values greater than zero define a synchronous transfer mode and the offset is determined as shown below. This offset determines the effective FIFO depth for synchronous data transfers, and is typically determined by negotiation with the other SCSI device (as defined in the SCSI standard). The Transfer Period control bits select the minimum transfer period for both synchronous and asynchronous SCSI transfers and, if AM-Bus mode is used, the transfer period and the width of the $\overline{RE}/\overline{WE}$ strobes for host transfers. The period is defined in terms of the internal clock cycle time; the frequency of this clock is determined by the divisor selected in the OWN ID register.



11853-009A

Bit	Name	Description
0–3	OFx	The OFFSET bits are used to select the desired offset according to the following:

3	2	1	0	Selected Offset
0	0	0	0	0 (Asynchronous data phase transfers)
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	Undefined
1	1	1	X	Undefined

4–6 TPx

The TRANSFER PERIOD bits are used to select the desired transfer period according to the following table:

SCSI 6	5	4	DBA Bus Transfer Period	(SCSI REQ/ACK Synchronous Pulse Width and DBA Bus $\overline{RE}/\overline{WE}$ Pulse Width)
0	0	X	8 cycles	(4 cycles)
0	1	0	2 "	(1 ")
0	1	1	3 "	(1 ")
1	0	0	4 "	(2 ")
1	0	1	5 "	(3 ")
1	1	0	6 "	(4 ")
1	1	1	7 "	(4 ")

The 'cycle' referred to above is the period of the internal data transfer clock after the divisor chosen in the OWN ID register is applied. This period is calculated by the following formula:

$$\text{CYCLE} = \frac{\text{DIVISOR (from OWN ID)}}{2 \cdot \text{INPUT CLOCK FREQUENCY (MHz)}} \ (\mu\text{sec})$$

4

TRANSFER COUNT REGISTER

The TRANSFER COUNT register is a 24-bit register containing a preset value for the internal transfer counter. This preset value is loaded into the internal transfer counter when a Send, Receive, or Transfer command is issued. This counter is used to define command completion by decrementing as each data byte is transferred over the SCSI bus and causing a "successful completion" interrupt when the counter reaches zero. In Combination commands, this register specifies the number of bytes to be transferred during a Data phase.

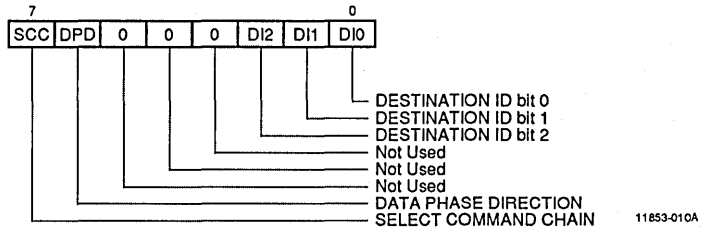
The counter function can be disabled by loading the TRANSFER COUNT register with zeros prior to issuing a command or by setting the SINGLE-BYTE TRANSFER bit in the COMMAND register concurrent with issuing the command. If the counter is disabled, the Send, Receive, or Transfer command will be completed when a single byte has been transferred.

After the completion of any successful transfer, the TRANSFER COUNT register will be zero. This includes commands issued in Single Byte Transfer mode.

When a transfer is interrupted by a halt on error condition, a SCSI bus phase change, or an abort, the TRANSFER COUNT register will contain the number of bytes NOT successfully transferred to/from the SCSI bus, including clearing the internal FIFO of any bytes left in the FIFO (see DATA register). This FIFO clearing process may cause the TRANSFER COUNT register to differ with the user's DMA controller count, because some bytes may have been transferred into the FIFO, but not to the SCSI bus; therefore, the TRANSFER COUNT should be used to determine the actual number of bytes transferred to/from the SCSI bus.

DESTINATION ID REGISTER

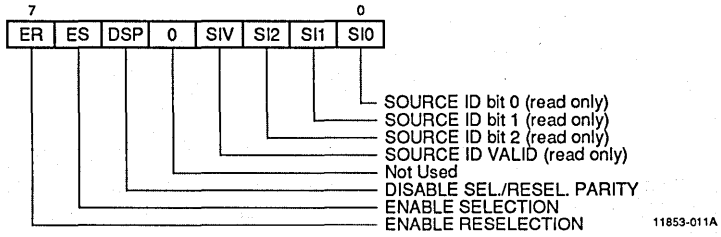
The DESTINATION ID register contains the encoded SCSI bus ID of the device which is to be selected or reselected when a Reselect or Select command is issued. This register also contains control bits that affect the operation of certain combination commands.



Bit	Name	Description
0–2	DIx	DESTINATION ID Bits DI0–DI2 contain the encoded SCSI bus ID of the device which is to be selected or reselected when a RESELECT or a SELECT command is issued.
6	DPD	DATA PHASE DIRECTION, when advanced features are enabled (see p.14), is used to specify the expected direction of the SCSI data phase, when it occurs. This allows the 33C93A to verify the direction during Select-and-Transfer commands before beginning the transfer. When this bit is zero, the expected direction is out (to the Target). When this bit is one, the expected direction is in (from the Target). An unexpected information phase error will occur if the direction does not match the setting of this bit.
7	SCC	SELECT COMMAND CHAIN is used only when the Reselect-and-Transfer command is issued with EDI=1. This bit selects which command is chained to when the data transfer is completed. When this bit is zero, a Send-Status-and-Command-Complete command begins executing. When this bit is one, a Send-Disconnect-Message command begins executing.

SOURCE ID REGISTER

The SOURCE ID register is used to report the SCSI bus ID of the device that has selected or reselected the 33C93A. It also contains bits that enable and control response to selection and reselection.



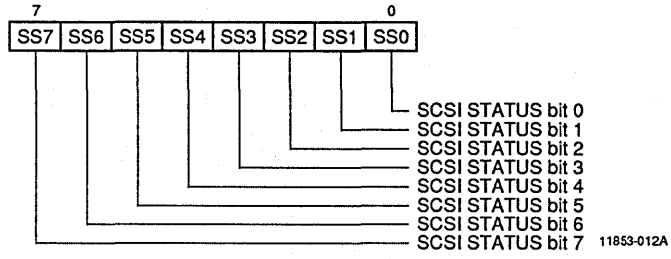
Bit	Name	Description
0–2	SIx	SOURCE ID Bits 0–2 are valid only if the SIV bit is set to one. These bits indicate the SCSI bus ID of the device that selected or reselected the 33C93A.
3	SIV	SOURCE ID VALID is set to one after the 33C93A is selected or reselected if the other SCSI bus device asserted its own bus ID bit (in addition to the bus ID bit of the 33C93A) during the select/reselect phase. This bit is zero if only the bus ID bit of the 33C93A was asserted.
5	DSP	DISABLE SELECT PARITY, when set to one, causes the 33C93A to ignore the bus parity when responding to selection or reselection. When this bit is zero, any selection or reselection with a parity error is ignored.
6	ES	ENABLE SELECTION, when set to one, enables the 33C93A to respond to a selection by another device on the SCSI bus. When this bit is zero, any selection is ignored.
7	ER	ENABLE RESELECTION, when set to one, enables the 33C93A to respond to a reselection by another device on the SCSI bus. When this bit is zero, any reselection is ignored.

SCSI STATUS REGISTER

The SCSI STATUS register is a read-only register which indicated the cause of the most recent INTRQ assertion. INTRQ is asserted whenever a condition occurs within the Am33C93A that requires intervention by the host; for example:

- The 33C93A has been reset;
- The command completed successfully;
- The bus phase changed;
- An error occurred.

Once INTRQ has been asserted, the contents of this register will not change until after the SCSI STATUS register has been read or until the 33C93A has been reset.



Bit	Name	Description
0-3	SSx	SCSI STATUS bits 0-3 are status qualifiers whose meaning depends upon which upper (4–7) status bit is set.
4–7	SSx	SCSI STATUS bits 4–7 define the type of interrupt that occurred. The possible codes are defined in the following table:

Status	Code	Group Meaning
0000	xxxx	The 33C93A is in a reset state.
0001	xxxx	A 33C93A command has completed successfully.
0010	xxxx	A 33C93A command has paused or was aborted by an Abort command.
0100	xxxx	A 33C93A command has been terminated prematurely due to an error or other unexpected condition.
1000	xxxx	An event on the SCSI bus requires service.

All other Status Code groups are currently not used and are reserved for future use.

In the following tables, the 'STATE' column indicates the current state in which the Status Code can occur. Also, the MCI field refers to the signals that define a SCSI bus information transfer phase: MSG, C/D, and I/O. A bit set to one indicates that the signal is asserted on the SCSI bus. A zero indicates negation. Whenever one of these Status Codes occurs, the REQ signal is asserted on the SCSI bus. The table below summarizes the meaning of the MCI field:

MCI CODE	MEANING
000	Data Out phase
001	Data In phase
010	Command phase
011	Status phase
100	Unspecified Info Out phase
101	Unspecified Info In phase
110	Message Out phase
111	Message In phase

Reset State Interrupts

Status	Code	State	Specific Meaning
0000	0000	DTI	33C93A Reset. The device has been reset, or a Reset command has executed successfully with no advanced features enabled. The new state of the 33C93A is disconnected.
0000	0001	DTI	33C93A Reset. The device has successfully completed a Reset command with advanced features enabled. The new state of the 33C93A is disconnected.

Successful Completion Interrupts

Status	Code	State	Specific Meaning
0001	0000	D	A Reselect command completed successfully. The new state of the 33C93A is connected as a Target.
0001	0001	D	A Select command completed successfully. The new state of the 33C93A is connected as an Initiator.
0001	0010	—	Reserved for future use.
0001	0011	DT	A Receive, Send, Reselect-and-Transfer, Wait-for-Select-and-Receive, Send-Status-and-Command-Complete, or a Send-Disconnect- Message command completed successfully ($\overline{ATN}$ is not asserted).
0001	0100	DT	A Receive, Send, Reselect-and-Transfer, Wait-for-Select-and-Receive, Send-Status-and-Command-Complete, or a Send-Disconnect-Message command completed successfully ($\overline{ATN}$ is asserted).
0001	0101	DT	A Translate Address command completed successfully.
0001	0110	DI	A Select-and-Transfer command completed successfully.
0001	0111	—	Reserved for future use.
0001	1MCI	I	A Transfer (non-MESSAGE IN phase) command completed successfully. MCI defines the new information type (SCSI bus phase) being requested.

Paused or Aborted Interrupts

Status	Code	State	Specific Meaning
0010	0000	I	A Transfer Info (MESSAGE-IN phase) command has paused with $\overline{ACK}$ asserted. This allows the host to examine the message before accepting it.
0010	0001	I	A Save-Data-Pointer message was received during a Select-and-Transfer command. The host should save its current data buffer pointer.
0010	0010	D	A Select or Reselect command was aborted.
0010	0011	T	A Receive or Send command has halted by an error or was aborted ($\overline{ATN}$ is not asserted).
0010	0100	T	A Receive or Send command has halted by an error or by assertion of $\overline{ATN}$ or was aborted ($\overline{ATN}$ is asserted).
0010	0101	—	Reserved for future use.
0010	0110	—	Reserved for future use.
0010	0111	D	The 33C93A has been reselected during a Select-and-Transfer (with IDI=0) by a Target that does not match the SCSI bus ID loaded into the DESTINATION ID register or the following Identify message did not match the LUN loaded into the TARGET LUN register. $\overline{ACK}$ has been left asserted following the Identify message, and the bus ID and LUN of the reselecting Target are available in the SOURCE ID and TARGET LUN registers. (Advanced Mode only)
0010	1MCI	I	A Transfer command was aborted. MCI define the new information type (SCSI bus phase) being requested.

Terminated Interrupts

Status	Code	State	Specific Meaning
0100	0000	DTI	An invalid command was issued.
0100	0001	I	An unexpected disconnect (SCSI bus free) by the Target caused a command to terminate. The new state of the 33C93A is disconnected.
0100	0010	D	A timeout occurred during a Select or Reselect command. The state of the 33C93A is disconnected.
0100	0011	TI	A parity error caused a command to terminate ($\overline{ATN}$ is not asserted). The transfer direction determines whether it is a SCSI or host parity error.
0100	0100	TI	A parity error caused a command to terminate ($\overline{ATN}$ is asserted). The transfer direction determines whether it is a SCSI or host parity error.
0100	0101	DT	The Logical Address exceeded the disk boundaries.
0100	0110	D	A Target whose SCSI bus device ID does not match the bus ID set in the DESTINATION ID register has reselected the 33C93A during a Select-and-Transfer command (with IDI=0). This interrupt occurs when the 33C93A is not in Advanced Mode. The new state of the 33C93A is connected as an Initiator.
0100	0111	I	An incorrect status byte was received during a Select-and-Transfer command.
0100	1MCI	I	An unexpected information phase was requested. MCI define the SCSI bus phase which is requested. This is typically caused by a phase change before the Transfer Count has reached zero or by an unexpected phase sequence occurring during a Select-and- Transfer command.

Service Required Interrupts

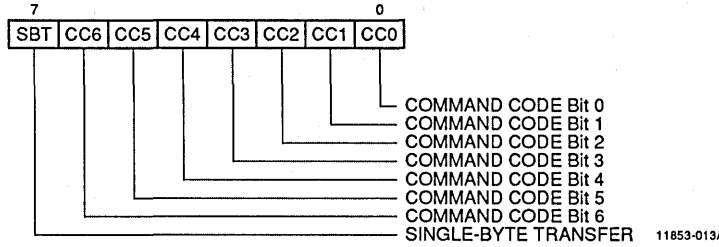
Status	Code	State	Specific Meaning
1000	0000	D	The 33C93A has been reselected. The new state of the 33C93A is connected as an Initiator. No Identify message transfer has yet occurred.
1000	0001	D	The 33C93A has been reselected in Advanced Mode. The SCSI bus ID of the Target may be read from the SOURCE ID register. The Identify message from the Target may be read from the DATA register. The $\overline{ACK}$ signal is left asserted. The new state of the 33C93A is connected as an Initiator.
1000	0010	D	The 33C93A has been selected ($\overline{ATN}$ was not asserted). The new state of the 33C93A is connected as a Target.
1000	0011	D	The 33C93A has been selected ($\overline{ATN}$ was asserted). The new state of the 33C93A is connected as a Target.
1000	0100	T	The $\overline{ATN}$ signal has been asserted.
1000	0101	I	A disconnect has occurred. The new state of the 33C93A is disconnected.
1000	0110	—	Reserved for future use.
1000	0111	T	The Wait-for-Select-and-Receive command has paused because the first byte of the incoming CDB is not a known command group. The OWN ID register must be loaded with the CDB length, and the command resumed. The CDB1 register may be examined to determine the SCSI command group from the opcode. The new state of the 33C93A is connected as a Target. (Advanced Mode only)
1000	1MCI	I	The REQ signal has been asserted following connection or when the 33C93A is in the Initiator state and no command is executing. The information phase type should be examined. MCI define the information phase (SCSI bus phase) which is being requested.

COMMAND REGISTER

The COMMAND register is used to issue the 33C93A commands. This register should never be loaded when the CIP or INT bits (in AUXILIARY STATUS) are set to one, and a Level II command should never be loaded when the BSY bit is set to one.

The SINGLE-BYTE TRANSFER (SBT) bit in the COMMAND register is only used during information transfer type commands. When this bit is set in conjunction with one of these commands, the transfer counter is disabled and exactly one byte is to be transferred, regardless of the value in the TRANSFER COUNT register. The previous contents of the TRANSFER COUNT register are not preserved.

Refer to the COMMANDS section for a description of the commands and their corresponding command codes.



DATA REGISTER

The DATA register is used to transfer data bytes between the host and the SCSI bus during the SCSI information transfer phases (command, data, status, or message phase). It may be accessed by the processor during any type of information phase (simple Level II commands) or via the DMA/DBA Bus interface during a SCSI Data In phase or Data Out phase (simple and combination Level II commands).

The DATA register is actually a port for the host interface into the internal twelve byte FIFO of the 33C93A. The FIFO is used for all transfers (synchronous and asynchronous) between the SCSI bus and the host bus, for both DMA and processor access transfers. If the 33C93A is to be halted for any reason (through ABORT, for example), then data transfers with this FIFO must continue until an interrupt occurs. This must be done so that the FIFO is returned to a ready state for subsequent transfers, and to flush incoming data to the host bus.

The DATA register is accessed by the processor during a data phase when the CONTROL register DMA mode select bits are all reset (=0), and when the DBR bit in the AUXILIARY STATUS register is true. The processor writes (reads) the DATA register by loading the ADDRESS register with a hex value of 19 and asserting the $\overline{WE}$ ($\overline{RE}$) and $\overline{CS}$ pins. This access also occurs during non-data phases.

When the CONTROL register DMA mode select bits are set for DMA mode or BURST mode, the DMA interface is enabled. In this case, the DATA register is written (read) when the $\overline{DACK}$ and $\overline{WE}$ ($\overline{RE}$) pins are asserted in response to the assertion by the 33C93A of the $\overline{DRQ}$ pin.

When the DBA Bus is selected by the DMA mode select bits, the $\overline{RCS}$ pin functions as an external buffer chip select and the $\overline{WE}$ and $\overline{RE}$ pins become outputs, allowing the 33C93A to automatically transfer data between its DATA register and the external buffer. In this mode, bus control can be returned to the external processor or any other device by negating the DRQ pin.

Reset Conditions

HARDWARE RESET

The following results occur when the 33C93A is reset by the assertion of the $\overline{MR}$ signal:

- The AUXILIARY STATUS register is reset to zero. The INT bit (and the INTRQ pin) is set to one when the hardware reset completes.
- The OWN ID register is reset to zero.
- Advanced mode is disabled.
- The ES, ER, and DSP bits in the SOURCE ID register are reset to zero.
- The SCSI STATUS register is reset to zero.
- The internal FIFO, internal transfer counter (not the host accessible register), offsets, and state machines are cleared.
- The internal clock divider circuit is set to divide by two.

The following host accessible registers are NOT affected by the $\overline{MR}$ signal:

- Registers 01 hex through 15 hex;
- SOURCE ID (16 hex) register bits 0-3;
- COMMAND register (18 hex);

Note: The SCSI Soft Reset may be implemented by using the SCSI bus reset signal to cause a reset of the 33C93A (for example, OR the host power on reset signal with the received SCSI bus reset (RST) signal). The host may examine the registers that are not affected by the $\overline{MR}$ signal to recover from the SCSI reset condition.

SOFTWARE RESET

The following results occur when the 33C93A executes the Reset command:

- The DBR bit in the AUXILIARY STATUS register is reset to zero. The INT bit (and INTRQ pin) is set to one when the Reset command is complete.
- All SCSI bus signals are reset to the negated state.
- The internal FIFO, internal transfer counter (not the host accessible register), offsets, and state machines are cleared.
- The OWN ID register is interpreted and the clock divisor, host parity, and advanced mode are configured.
- Registers 01 hex through 16 hex are reset to zero. The COMMAND register (18 hex) is also reset to zero.
- The SCSI STATUS register is set as commanded by the EAF bit in the OWN ID register.

COMMANDS

Command List

Command Code (HEX)	Valid Command	States	Level
00	Reset	D,T,I	I
01	Abort	D,T,I	I
02	Assert ATN	I	I
03	Negate ACK	I	I
04	Disconnect	T,I	I
05	Reselect	D	II
06	Select-with-ATN	D	II
07	Select-without-ATN	D	II
08	Select-with-$\overline{\text{ATN}}$ and-Transfer	D,I	II
09	Select-without-$\overline{\text{ATN}}$ and-Transfer	D,I	II
0A	Reselect-and-Receive-Data	D,T	II
0B	Reselect-and-Send-Data	D,T	II
0C	Wait-for-Select-and-Receive	D,T	II
0D	Send-Status-and-Command-Complete	T,I	I
0E	Send-Disconnect-Message	T	II
0F	Set IDI	D,T,I	I
10	Receive Command	T	II
11	Receive Data	T	II
12	Receive Message Out	T	II
13	Receive Unspecified Info Out	T	II
14	Send Status	T	II
15	Send Data	T	II
16	Send Message In	T	II
17	Send Unspecified Info In	T	II
18	Translate Address	D,T	II
20	Transfer Info	I	II

33C93A states:
D = Disconnected
T = Connected as a Target
I = Connected as an Initiator

Command Levels:
I = Level I command
II = Level II command

33C93A Command Types

There are two basic types of 33C93A commands: Level I and Level II. Level I commands may be issued while a Level II command is in progress (indicated by an AUXILIARY STATUS of BSY=1,CIP=0) and, except for the "Abort" and "Reset" commands, do not generate an interrupt upon their completion. Level II command execution will always result in an interrupt. If a Level II command is issued while another Level II command is executing, unpredictable results may occur.

There are two types of Level II commands. 'Simple' Level II commands are associated with a single operation or phase (for example, selection or information transfer). 'Combination' Level II commands combine multiple phases into a single 33C93A command to minimize interrupt overhead. The Initiator combination commands 'expect' certain SCSI bus phases at certain times during a sequence. These expected phases are based on common sequences performed by a Target on the SCSI bus; any deviation causes an interrupt. Target combination commands can be chained together to further minimize interrupt overhead by creating longer phase sequences.

NOTE: When using command chaining, care must be taken to ensure that all commands in the chain are initialized prior to issuing the command.

The 33C93A will be in one of three "states" during operation: Disconnected, Connected as a Target, or Connected as an Initiator. Certain commands are valid only in particular states as indicated in the COMMAND LIST. An attempt to issue a Level II command which is invalid for the present 33C93A state will cause an "invalid command" interrupt. Level I commands issued in invalid states will be ignored.

Advanced Mode Features

The 33C93A has several new features included which add new functions to the original 33C93 design. Some of these features cause the 33C93A to be incompatible with the 33C93. These features have been grouped together under the heading of 'Advanced Mode' features. These features are disabled when the 33C93A is reset by the $\overline{MR}$ signal (hardware reset). They must be enabled by the host by issuing the 'Reset' command with the 'Enable Advanced Features' (EAF) bit set in the OWN ID register. The host can determine if advanced features have been enabled (thereby implying that a 33C93A is installed) by examining the SCSI STATUS register after issuing the 'Reset' command.

The features enabled by this bit are described in the following.

UNEXPECTED RESELECTION

When in normal (33C93A) mode, a reselection when idle (ER=1) or when disconnected during a Select-and-Transfer command (and the Target bus ID does not match the DESTINATION ID register) causes an immediate interrupt after the reselection handshake is complete. In Advanced Mode, the 33C93A will continue to the Message In phase to fetch the Identify message. If the 33C93A was idle, the SCSI STATUS register will be set to 81 hex, and the Identify message will be in the DATA register. If the 33C93A was executing a Select-and-Transfer command, the SCSI STATUS register will be set to 27 hex, and the Identify message will be in the TARGET LUN register. In either case, the SOURCE ID register will contain the SCSI bus ID of the reselecting Target, and the ACK signal remains asserted so that the Identify message may be rejected.

UNKNOWN SCSI COMMAND GROUPS

When a SCSI Command Descriptor Block is transferred on the SCSI bus, the command length in bytes is determined by the group code, which is found in bits 7–5 of the first command byte, or opcode. Group 0 (opcodes 00 to 1F hex), group 1 (opcodes 20 to 3F hex), and group 5 (opcodes A0 to BF hex) commands are defined by the SCSI standard (X3.131-1986) as six, ten, and twelve byte commands, respectively. All other command groups are undefined by that standard. In normal mode, the 33C93A will assume that these undefined groups are six byte commands when executing Select-and-Transfer or Wait-for-Select-and-Receive commands. In Advanced Mode, the following events will occur:

Select-and-Transfer: When loading the CDB into the CDB registers prior to issuing the command, the host also loads the expected command length into the OWN ID register. The 33C93A uses this value to make sure the correct number of bytes are then transferred in the command phase.

Wait-for-Select-and-Receive: When receiving the CDB from the Initiator, the 33C93A will check the first CDB byte as soon as it is received. If the group is undefined, an interrupt will occur so that the host can examine the first command byte in the CDB 1ST register, and then load the TOTAL command length into the OWN ID register. The SCSI STATUS register is set to 87 hex, and the COMMAND PHASE register is set to 31 hex, when this interrupt occurs.

After the interrupt, the 33C93A will only accept a Resume Wait-for-Select-and-Receive command, Abort, Disconnect, or Reset command. All other commands are invalid; during the interrupt processing, the 33C93A will continue to transfer the first six bytes of the command into its internal FIFO.

DATA PHASE DIRECTION
During a Select-and-Transfer command in normal mode, the Data phase direction is determined solely by the Target; if this direction does not match the direction expected by the host, the 33C93A will not detect this error but expects that the transfer will continue. In Advanced Mode, the DPD bit in the DESTINATION ID register is compared with the state of the I/O signal on the SCSI bus. If the expected and actual directions do not match, an interrupt will occur with 'unexpected phase' status in the SCSI STATUS register.

Level I Commands

RESET (00 HEX)
The Reset command performs a similar function to the hardware reset caused by asserting the $\overline{MR}$ pin except that the OWN ID register is sampled for information concerning the operating configuration of the 33C93A. The 33C93A is also initialized as described in the RESET CONDITIONS section. The Reset command may be executed in any 33C93A state and will force the 33C93A into the Disconnected state, aborting any previously issued command in progress. Upon completion of the Reset command, an interrupt is generated the SCSI STATUS will be 00 hex or 01 hex, depending on the contents of the OWN ID register.

ABORT (01 HEX)
The Abort command is valid in the Disconnected and Connected-as-a-Target states.The Abort command has different effects depending on the state and the command that is currently executing, as described below:

Disconnected State: In the Disconnected state, the Abort command may be used to halt an attempted Select, Select-and-Transfer, Reselect, or Reselect-and-Transfer command. If the Abort command is issued following a Select or Reselect command and the Am33C93A has won arbitration, the Am33C93A releases the SCSI bus by removing the Bus ID bits while $\overline{SEL}$ is asserted and checking for a negated $\overline{BSY}$ signal. If after at least 200 µs, there is no $\overline{BSY}$ response, the Am33C93A goes to a Bus Free condition and generates a "paused/aborted" interrupt. If there is a response within this time period, then a "successful completion" interrupt will result instead.

Note that the Am33C93A will neglect Abort command unless the command is completed. After the completion of a command, the Am33C93A will accept an Abort command and will go to the Bus Free phase and generate a "Pause/Aborted Interrupt".

DISCONNECT (04 HEX)
The Disconnect command may be used in either the Target or the Initiator connected states. In the Target role, the Disconnect command is the normal procedure for disconnecting from the SCSI bus following the information transfer phase. In the Initiator role, Disconnect can be used to release the bus following a timeout condition. The Disconnect command causes the immediate release of all bus signals and, in Target mode, returns the SCSI bus to the Bus Free phase. If the Disconnect command is issued during an active Level II command, the Level II command is immediately terminated and the 33C93A transitions to the Disconnected state.

ASSERT ATN (02 HEX)
The Assert ATN command is only valid when Connected as an Initiator. It is normally used to allow the Initiator to inform a Target that it has a message pending (The Target is expected to respond by performing a Message Out Phase).

$\overline{ATN}$ is automatically negated:

- Before the last byte of a Transfer Info command issued in response to the Message Out phase;
- When the Identify message out is transferred to the Target during a Select-and-Transfer command;
- When a SCSI Bus Free phase occurs.

The Select-with-ATN and Select-with-ATN-and-Transfer commands will cause the 33C93A to automatically assert $\overline{ATN}$ prior to the release of $\overline{SEL}$ providing the bus arbitration is won.

NEGATE ACK (03 HEX)
The Negate ACK command causes $\overline{ACK}$ to be negated. It may be used in the following situations:

- after successful completion of a Message-In Transfer Info commands;
- after the 33C93A has detected a parity error on any received SCSI information and the HALT on SCSI PARITY ERROR (HSP) bit is set;
- after unexpected reselection in advanced mode; and
- after a save-data-pointer message is received during a select-and-transfer command.

Host parity errors do not affect the $\overline{ACK}$ signal. For all other Initiator transfers, $\overline{ACK}$ negation is automatic.

In the case of a Message-In transfer, incoming messages may be rejected and the Initiator may indicate its intent to send either a "MESSAGE REJECT" or a "MESSAGE PARITY ERROR" Message by issuing the Assert ATN command prior to issuing the Negate Ack command. If the incoming message is to be accepted, only the Negate Ack command should be issued.

During non-Message-In transfers, if the Transfer command is terminated by a parity error, the Assert ATN command can again be issued prior to Negate ACK, this time indicating the Initiator's intent to send an "INITIATOR DETECTED ERROR" Message.

SET IDI (0F HEX)
The Set IDI command is used in the Initiator role to support overlapped SCSI operations. If a SCSI command is executing via a Select-and-Transfer command, then the Set IDI command may be used to set the IDI bit in the CONTROL register, which then causes an interrupt to occur upon a Target disconnection. This ability allows the IDI bit to be left reset when the first SCSI operation is started, which may reduce the number of 33C93A interrupts, yet also allows a second operation to be started when needed without waiting for the first operation to be completed.

Simple Level II Commands

SELECT-WITH-ATN (06 HEX)
Select-with-ATN is valid only in the Disconnected state and when issued will cause the 33C93A to select a Target. Before issuing this command, the SCSI Bus ID of the Target device should be written into the DESTINATION ID register. When the Select- with-ATN command is issued, the 33C93A begins bus arbitration. If the 33C93A is selected or reselected by another device during the arbitration, the Select-with-ATN command is aborted and a "service required" interrupt (8x hex) is generated.

Should the 33C93A win the arbitration, $\overline{SEL}$ and $\overline{ATN}$ are asserted, the Target and Initiator Bus IDs are placed on the SCSI data bus, and then $\overline{BSY}$ is deasserted. At this time, a timeout sequence whose length is determined by the value in the TIMEOUT PERIOD register begins. If $\overline{BSY}$ is not asserted by the Target before a timeout occurs, the 33C93A begins its selection abort sequence (as described in the Abort command description), and if there is no Target response the Select-with-ATN command is terminated and a "terminated" interrupt is generated. If the Target responds before the timeout period has elapsed or before the selection abort sequence is complete, the 33C93A negates the $\overline{SEL}$ signal, putting the 33C93A in a Connected-as-an-Initiator state. A "successful completion" interrupt indicates that the Select-with-ATN command has been completed successfully.

If the 33C93A does not win the arbitration or there is no response from the Target and the timeout feature is disabled, the Select-with-ATN command can be aborted with an Abort command. When the Abort command is successfully executed under these circumstances, the 33C93A is disconnected from the bus and a "paused/aborted" interrupt is generated.

SELECT-WITHOUT-ATN (07 HEX)
The Select-without-ATN command is identical to the Select-with-ATN command except that $\overline{ATN}$ is not set during the Selection Phase.

RESELECT (05 HEX)
The Reselect command is identical to the Select-without-ATN command except that the I/O signal is asserted upon completion of the Arbitration Phase. Successful completion of the Reselect command results in the 33C93A being Connected as a Target.

RECEIVE (10-13 HEX)
There are four Receive commands which are distinguished from each other only by the state of three SCSI interface signals and the type of data that is transferred. These commands, consisting of the Receive Command, Receive Data, Receive Message Out, and Receive Unspecified Info Out commands are valid only in the Connected-as-a-Target state. The type of the Receive command selected determines the state of the $\overline{I/O}$, C/$\overline{D}$, and $\overline{MSG}$ outputs during the command according to the following chart (1=asserted):

Receive Command Type	OPCODE	MSG	C/D	I/O
Receive Command	10	0	1	0
Receive Data	11	0	0	0
Receive Message Out	12	1	1	0
Receive Unspecified Info Out	13	1	0	0

The Receive commands are information transferring commands and are therefore dependent on the SBT bit in the COMMAND register for determination of a successful completion. In addition to a termination caused by reset (via either a Reset command being issued or assertion of the $\overline{MR}$ pin), a Receive command completion or termination will occur under any of these conditions: (1) The internal transfer counter is disabled (SBT=1 or the TRANSFER COUNT register is loaded with zero) and a single byte has been read from the DATA register; (2) The counter has decremented to zero (with SBT=0) indicating that the specified number of bytes have been transferred; (3) A parity error has

been detected on one of the received data bytes (and HSP=1); (4) The $\overline{ATN}$ pin is asserted (and HA=1); (5) The Abort command is issued; or (6) A Disconnect command is issued.

When the Receive command is completed as a result of receiving the correct number of bytes, a "successful completion" interrupt will be generated. If a parity error has caused termination, a "terminated" interrupt will instead be generated. In this case, the TRANSFER COUNT register will contain the number of bytes yet to be transferred. After any completion or termination of the Receive commands except those due to a subsequent Disconnect command or reset, the 33C93A is in the Connected-as-a-Target state.

As data transfer commands, the Receive commands are dependent on the DMA mode select bits in the CONTROL register for the DATA register accessing mode. These bits determine whether the DATA register accesses will be handled by the processor or through a DMA/DBA interface. When the processor is required to read the DATA register (i.e. DMA mode select bits=0), it must monitor the DBR status bit (in AUXILIARY STATUS) to determine when a byte is available for reading. During Receive commands, this status bit will be reset when a byte is read from the DATA register and set when a byte is loaded into the DATA register via the SCSI interface. DBR is also reset when a Receive command is issued.

All information transfers involving other than data information are asynchronous. However, if the information phase involves data transfers, the SYNCHRONOUS TRANSFER register will be evaluated. In this case, any selected offset other than zero results in synchronous transfers. The minimum Transfer Period for both types of transfers is determined by the transfer period bits in this same register.

SEND (14-17 HEX)

As in the case of the Receive commands, there are four Send commands which are distinguished only by the state of the $\overline{I/O}$, C/$\overline{D}$, and $\overline{MSG}$ pins and the type of data that is transferred. The four Send commands, also valid in the Connected-as-a-Target state only, are the Send Status, Send Data, Send Message In, and Send Unspecified Info In commands. The SCSI pin states during the Send commands are determined by the particular command as follows (asserted=1):

Send Command Type	OPCODE	MSG	C/D	I/O
Send Status	14	0	1	1
Send Data	15	0	0	1
Send Message In	16	1	1	1
Send Unspecified Info In	17	1	0	1

The Send commands are also information transferring commands and as such are also dependent upon the SBT bit in the COMMAND register for command completion. In addition to that caused by reset (via either a Reset command being issued or assertion of the $\overline{MR}$ pin), a Send command completion or termination will occur under any of these conditions: (1) The internal transfer counter is disabled (SBT=1 or the TRANSFER COUNT register is loaded with zero) and a single byte has been read from the DATA register; (2) The counter has decremented to zero (with SBT=0) indicating that the specified number of bytes have been transferred; (3) A parity error has been detected on one of the data bytes from the host (and HHP=1); (4) The $\overline{ATN}$ pin is asserted (and HA=1); (5) The Abort command is issued; or (6) A Disconnect command is issued. The 33C93A remains Connected-as-a-Target following the Send command completion/termination unless the Disconnect command or reset was used to force a termination.

During a Send command, DATA register accessing is controlled by the DMA mode select bits in the CONTROL register. When these bits are set to the appropriate mode, loading of the DATA register is accomplished by a DMA controller or through the Am-Bus interface. If the DMA mode select bits are zero, the processor must poll the AUXILIARY STATUS register and can write to the DATA register only when the DATA BUFFER READY bit is set (DBR=1). Send commands cause the DBR bit to be reset every time the processor loads a byte into the DATA register and set when a byte is transferred from the DATA register onto the SCSI data bus. The DBR bit will also be set upon issuing a Send command.

As in the case of Receive commands, synchronous transfers will occur only when data transfers are involved and an offset other than zero is selected.

TRANSFER INFO (20 HEX)

The Transfer Info command is valid only when Connected as an Initiator and is used to send and receive data, command, status, and message information.

The first $\overline{REQ}$ assertion following connection as an Initiator results in a "service required" interrupt. The processor should examine the SCSI STATUS register to determine the type and direction of information transfer requested by the Target, and then issue a Transfer Info command in response. While an Initiator, the 33C93A will also generate an interrupt each time the Target device requests a new type of information transfer phase.

As in the case of the Send and Receive commands, when completion of the Transfer Info command depends upon the internal transfer counter, the processor should load the TRANSFER COUNT register prior to issuing this command. The DMA mode select bits in the CONTROL register, the offset and transfer period bits in the SYNCHRONOUS TRANSFER register, and the SBT bit in the COMMAND register are used during Transfer Info commands just as they are during the Send and Receive commands. However, for processor access of the DATA register during Transfer Info commands (when the DMA mode select bits are zero or the bus phase is other than Data phase), behavior of the DATA BUFFER READY (DBR) status bit is determined by the direction of information transfer as defined by the $\overline{I/O}$ pin. When the transfer is from Initiator to Target, the DBR bit is reset by writing to the DATA register and is set when the byte is transferred from the DATA register onto the SCSI data bus. When the transfer is from Target to Initiator, DBR is set when a byte is received over the SCSI data bus and transferred into the DATA register and is reset by reading the DATA register. DBR is also reset whenever a Transfer Info command is issued.

There are several causes of a Transfer Info command completion/termination in addition to a reset. Just as for a Send or Receive command, the Transfer Info command can be terminated by issuing a subsequent Disconnect or Abort command. The Abort command will cause a "paused/aborted" interrupt to be generated after execution (leaving the 33C93A in a connected state), while the Disconnect command causes an immediate disconnect and does not generate an interrupt.

A Transfer Info command will either complete or pause when the specified number of bytes (either a single byte or multiple bytes as defined by the SINGLE-BYTE TRANSFER bit in the COMMAND register) has been sent or received. The 33C93A generates a "successful completion" interrupt only after receiving another $\overline{REQ}$ from the Target during non-Message-In information phases but generates a "paused/aborted" interrupt for Message-In phases without waiting for an additional $\overline{REQ}$ (Note that when the completed Transfer Info command was a Message-In transfer phase, the $\overline{ACK}$ pin will be left asserted by the 33C93A in the last REQ-ACK cycle of the command, and the processor is required to issue a Negate ACK or an Assert ATN followed by a Negate ACK command to accept or reject the message).

If a parity error is detected on a data byte received from the SCSI bus (and HSP=1) or on a data byte byte received from the host (and HHP=1), then the 33C93A will terminate the command and, for SCSI parity errors, will leave $\overline{ACK}$ asserted (to also halt the Target). In this case a "terminated" interrupt is generated. Finally, a negation of the $\overline{BSY}$ signal (i.e. the Target suddenly disconnects) or a transition in the $\overline{I/O}$, C/$\overline{D}$, and/or $\overline{MSG}$ pins during a Transfer command will also terminate the command and generate a "terminated" interrupt.

If a parity error is detected on a received byte but parity error command termination is disabled (HSP=0 or HHP=0, as appropriate), the 33C93A will still set the PARITY ERROR status bit in the AUXILIARY STATUS register but will not terminate the command as a result of this error.

TRANSLATE ADDRESS (18 HEX)

The Translate Address Command performs a logical-address to physical-address translation. Certain SCSI commands involve a logical address which may be up to 32 bits in length. When a command is detected which requires address translation, the processor can reload the logical address into the 33C93A LOGICAL ADDRESS register and then issue the Translate Address command to have the 33C93A do the conversion. Upon receiving a "successful completion" interrupt, the processor can read the CYLINDER NUMBER, HEAD NUMBER, and SECTOR NUMBER registers to extract the logical address. The disk parameters contained in the TOTAL SECTORS, TOTAL HEADS, and TOTAL CYLINDERS registers must also be valid before issuing a Translate Address command.

If automatic compensation for spare sectors is to be performed by the Am33C93A, then the number of spare sectors per cylinder and total number of sectors per cylinder must also be loaded, respectively, into the HEAD NUMBER and CYLINDER NUMBER registers. A "terminated" interrupt will occur if any division operation performed during this command results in an overflow.

Combination Level II Commands

SELECT-AND-TRANSFER (08 AND 09 HEX)

The Select-and-Transfer commands greatly reduce the host or local processor interrupt-handling burden by enabling the 33C93A's internal microprocessor to manage the low-level SCSI protocol, resulting in as few as one interrupt per SCSI operation. Select-and-Transfer commands are used when in an Initiator role, and typically consist of at least the following SCSI phases: (1) Selection of a Target device; (2) Sending of a command; (3) Reception of status information; and (4) Reception of a COMMAND COMPLETE Message. These commands optionally consist of a Data Transfer phase and additional Message Transfer phases.

The 33C93A will update the COMMAND PHASE register as the Select-and-Transfer command executes. Upon completion or termination of the command, the local processor can read this register to determine where the SCSI operation stopped.

The two Select-and-Transfer commands differ from each other only by whether or not the $\overline{ATN}$ pin is asserted during the Selection phase. The ability to assert $\overline{ATN}$ during Selection supports the SCSI Message Protocol which calls for an IDENTIFY Message Out phase following the Selection. When executing a Select W/ATN-and-Transfer commands, the 33C93A expects the Target to request a Message Out phase immediately following selection, whereas for a Select W/O ATN-and-Transfer command, it expects the Target to directly enter Command phase. The Select-and-Transfer commands, moreover, support Group 0 (6-byte CDB), Group 1 (10-byte CDB), and Group 5 (12-byte CDB) SCSI commands.

When a Select-and-Transfer command is issued, the 33C93A arbitrates for the bus and selects a Target just as during a Select command. If the Target does not respond before a timeout occurs, the Select-and-Transfer command halts and a "terminated" interrupt is generated. Failure to complete the Selection phase is also indicated by the fact that the COMMAND PHASE register contains all zeros. If the Selection is successful, no interrupt is generated, but the COMMAND PHASE register will be set to a hex 10.

After completing the Selection phase, the 33C93A begins an information transfer phase. If $\overline{ATN}$ has been asserted (i.e. a Select W/ATN-and-Transfer command was issued), the 33C93A expects the Target to respond with a Message Out phase. If the first information phase request is other than a Message Out request, the 33C93A will terminate the command and generate a "terminated" interrupt. However, when the Target does request a Message Out phase, the 33C93A will respond by automatically sending an IDENTIFY Message. This single byte message is of the binary form: 1r000ttt, where r=1 if the ENABLE RESELECTION bit in the SOURCE ID register is equal to 1, and ttt is the encoded Target LOGICAL UNIT NUMBER contained in the TARGET LUN register. Once the IDENTIFY Message has been sent, the 33C93A will set the COMMAND PHASE register to hex 20.

Following the Message Out phase (or Selection phase when $\overline{ATN}$ was not asserted during Selection), a Command phase is expected by the 33C93A. Again, and throughout the entire Select-and-Transfer command execution, if the Target requests an unexpected information phase type, the 33C93A terminates the command and generates a "terminated" interrupt. If the Command phase is requested in this situation, the 33C93A will extract the SCSI command from the internal COMMAND DESCRIPTOR BLOCK registers and send the 6-, 10-, or 12-bytes of command information as determined by its evaluation of the SCSI command code in the CDB1 register. The COMMAND PHASE register is set to hex 30 before the first Command byte is sent and then increments with each byte transferred, so that for a 12-byte CDB command the COMMAND PHASE register will contain hex 3C when all bytes of the CDB have been transferred.

After the Command phase, the 33C93A expects either a Data In phase, Data Out phase, Status phase, or Message In phase. If the Target is requesting a Message In phase, a pending disconnection is assumed. The 33C93A therefore expects to receive either a Save-Data-Pointer message (hex 02) or a Disconnect message (hex 04). If either message is incorrect, or if a different message is received, a "terminated" interrupt will be generated to alert the processor of that fact and to allow the message to be read from the DATA register. A "terminated" interrupt will also be generated if the Target disconnects before sending the Disconnect message. When a correct Save-Data-Pointer message is received, a "paused/aborted" interrupt is generated and the Select-and-Transfer command terminated to allow the processor to save the SCSI data pointer. However, if a Disconnect message is received, the COMMAND PHASE register will be updated to hex 42 and command execution continues.

When the actual Target-disconnection does occur, the COMMAND PHASE register is updated to hex 43 and if the IDI bit is set, the Am33C93A terminates the Select-and-Transfer command by generating an 85H interrupt. However, if the IDI bit is reset, then instead the Am33C93A sits in an idle state, waiting for the Target to reconnect. If a different Target device Reselects the 33C93A, a "terminated" interrupt is generated. However, if the original Target Reselects the 33C93A, no interrupt is generated and the COMMAND PHASE register is set to hex 44.

Following the original Target Reselection, the 33C93A expects a Message In phase which should consist of the Target sending an IDENTIFY Message. This single-byte message should be of the binary form: 10000ttt, where ttt is the Target LUN. If the data received by the 33C93A is different or the Target LUN specified in this byte does not match the contents of the TARGET LUN register, a "terminated" interrupt is generated and the Message byte may be examined by the processor. A correct IDENTIFY Message In phase results in the COMMAND PHASE register being updated to hex 45.

After the IDENTIFY Message is received from the Target or immediately after the Command Out phase (when there is no disconnection), a Data In phase, Data Out phase, or Status phase should occur. If the TRANSFER COUNT register contains any non-zero value, then the 33C93A will expect a Data Transfer phase. If Advanced Features are enabled, then the DPD bit will be examined to verify the correct data direction. If the data direction is incorrect, then a "terminated" interrupt is generated. In this phase, the 33C93A will use the TRANSFER COUNT register to determine the number of bytes to be transferred, and all host-side DATA register accesses will be accomplished via the

method selected by the DMA mode select bits in the CONTROL register. When the internal counter reaches zero, the Data Transfer phase is complete and the COMMAND PHASE register is set to hex 46.

Note that any number of disconnection/reconnection cycles may occur during the Data Transfer phase so long as they are accomplished according to the defined message protocol. The COMMAND PHASE register will cycle through the disconnect phases (41–45) with each disconnection and subsequent reconnection until all of the data has been transferred and the Data Transfer phase is complete.

A Status phase is expected by the 33C93A following the Data Transfer phase (or instead of the Data Transfer phase when the TRANSFER COUNT register contains a value of zero). At the start of the Status phase, the COMMAND PHASE register is loaded with hex 47. Upon completion of the Status phase, the COMMAND PHASE register will be updated to hex 50, and the received status byte is stored in the TARGET LUN register where it can be read upon completion of the command.

Following completion of the status-byte transfer, a Message In phase is expected. The 33C93A expects the Target to send a COMMAND COMPLETE Message (hex 00) to indicate that the SCSI command operation has been completed. After the 33C93A receives this COMMAND COMPLETE Message, the COMMAND PHASE register advances to hex 60, and if the EDI bit is reset, a "successful completion" interrupt is generated. The processor should then read the TARGET LUN register to examine the Target status. An additional interrupt will then occur when the SCSI bus goes to the Bus Free state, or when another $\overline{REQ}$ is asserted to begin an information transfer phase (as in SCSI linked commands). If the EDI bit is set, the "successful completion" interrupt will be suppressed until the Target disconnects from the SCSI bus.

At any time during execution of the Select-and-Transfer commands, an abnormal or unexpected condition will cause the 33C93A to terminate the command, set the appropriate status qualifiers, and generate a "terminated" interrupt. If the termination occurred during an information transfer phase, the 33C93A will be left in a Connected-as-an-Initiator state (unless termination was due to a sudden Target disconnection). Command termination during any other phase will result in the 33C93A being in a Disconnected state. Transfer commands may be used to handle the exception by transferring messages with the Target.

The following table summarizes the possible values that the COMMAND PHASE register can take during the Select-and-Transfer commands, and their meanings relative to command termination:

Command Phase	Meaning
00	No SCSI bus device has been selected. The 33C93A is in the disconnected state.
10	The Target has been selected. The 33C93A is now in the connected as an Initiator state.
20	An Identify message has been sent to the Target.
30	Command phase has started, no bytes transferred.
3x	Command phase, x bytes have been transferred.
41	Save-Data-Pointer message received.
42	Disconnect message received, bus not free.
43	Target has disconnected (SCSI bus free) following a successful transfer of a Disconnect message. The 33C93A is now in the disconnected state.
44	The 33C93A has been reselected by the Target whose SCSI bus ID matches the value in the DESTINATION ID register. The 33C93A is now in the connected as an Initiator state.
45	The 33C93A has received an Identify message from the Target whose Logical Unit Number matches the value in the TARGET LUN register.
46	The number of bytes specified in the TRANSFER COUNT register have been transferred to/from the Target during a Data Out/In phase.
47	The Target has begun a Receive Status phase.
50	The 33C93A has successfully received a Status byte from the Target and stored it in the TARGET LUN register.
60	The 33C93A has successfully received a Command Complete message from the Target.

A "Resume Select-and-Transfer" command is assumed whenever a normal "Select-and-Transfer" command is issued while the 33C93A is in the Connected-Initiator state. When the "Resume" is issued, the 33C93A examines the COMMAND PHASE register to determine where to restart the Select-and-Transfer command execution. This feature, in conjunction with the INTERMEDIATE DISCONNECT INTERRUPT enabled, allows support of multi-threaded or overlapped I/O on the SCSI bus.

The following table briefly describes the valid settings of the COMMAND PHASE register when resuming a Select-and-Transfer command:

Command Phase	Meaning
10	Resume after Target selection is complete.
20	Resume after Identify message out. Command phase is expected; an implied Negate ACK occurs.
30	Resume when Command phase has begun ($\overline{REQ}$ asserted).
41	Resume after Command phase or after Save-Data-Pointer message. Data, Status, or Message In phases are expected. An implied Negate ACK occurs.
42	Resume to complete Disconnect Message In; an implied Negate ACK occurs.
44	Resume after reselection by a Target.
45	Resume to transfer more data in a data transfer phase. May expect Status or Message In as well. An implied Negate ACK occurs.
46	Resume after the data phase has been completed, expecting Status phase or a Save-Data-Pointer/Disconnect Message In phase. An implied Negate ACK does NOT occur.
50	Resume to complete a Status phase; an implied Negate ACK occurs.
60	Resume to complete a Command Complete message from the Target; an implied Negate ACK occurs.

RESELECT-AND-TRANSFER (0A AND 0B HEX)

The Reselect-and-Transfer commands include the Reselect-and-Receive-Data and the Reselect-and-Send-Data commands. These commands cause the 33C93A to execute certain common SCSI bus phase sequences as a Target following a Reselection phase. These phases are determined by which command is sent, and the setting of two bits: the EDI bit in the CONTROL register; and the SCC bit in the DESTINATION ID register. The SCSI bus phase sequences are summarized below. Refer to the command descriptions of the Send-Status-and-Command-Complete and Send-Disconnect-Message commands for details on those sequences.

1. Reselect-and-Receive command, EDI=0, and SCC=don't care:
   - Reselection phase;
   - Send Identify Message In;
   - Receive Data Out phase;
   - Completion interrupt.
2. Reselect-and-Send command, EDI=0, and SCC=don't care:
   - Reselection phase;
   - Send Identify Message In;
   - Send Data In phase;
   - Completion interrupt.
3. Reselect-and-Receive command, EDI=1, and SCC=0:
   - Reselection phase;
   - Send Identify Message In;
   - Receive Data Out phase;
   - Chain to Send-Status-and-Command Complete;
4. Reselect-and-Send command, EDI=1, and SCC=0:
   - Reselection phase;
   - Send Identify Message In;
   - Send Data In phase;
   - Chain to Send-Status-and-Command-Complete;
5. Reselect-and-Receive command, EDI=1, and SCC=1:
   - Reselection phase;
   - Send Identify Message In;
   - Receive Data Out phase;
   - Chain to Send-Disconnect-Message;
6. Reselect-and-Send command, EDI=1, and SCC=1:
   - Reselection phase;
   - Send Identify Message In;
   - Send Data In phase;
   - Chain to Send-Disconnect-Message.

If the reselection attempt times out during a Reselect-and-Transfer command, $\overline{ATN}$ is asserted and HA=1, or if a parity error is detected on a incoming data byte (and HSP=1 or HHP=1, depending on data direction), the command will be terminated and the appropriate status will be set. In this case, the COMMAND PHASE register should be evaluated to determine the last successfully completed phase. If none of these conditions occurs, all phases complete normally, and if EDI=0, then a "successful completion" interrupt would be generated at this point. However, if EDI=1, no interrupt is generated and command chain occurs (as described above).

The following table summarizes the possible values that the COMMAND PHASE register can take during the Reselect-and-Transfer commands, and their meanings relative to command termination. See other command descriptions for additional values that can occur when command chaining is used.

Command Phase	Meaning
00	No SCSI bus device has been reselected. The 33C93A is in the disconnected state.
10	The 33C93A has successfully reselected the Initiator. The 33C93A is now in the connected as a Target state.
20	The Identify message has been successfully sent to the Initiator.
46	The requested data transfer has been completed.

A "Resume Reselect-and-Transfer" command is assumed whenever a normal "Reselect-and-Transfer" command is issued while the 33C93A is in the Connected-as-a-Target state. When the "Resume" is issued, the 33C93A examines the COMMAND PHASE register to determine where to restart the Reselect-and-Transfer command execution. This feature, in conjunction with the capability to chain to other combination commands, allows longer SCSI bus sequences to be executed by a single command.

The following table briefly describes the meaning of the COMMAND PHASE register when resuming a Reselect-and-Transfer command:

Command Phase	Meaning
10	Resume after Initiator reselection is complete; start with Identify Message Out.
20	Resume after Identify message out; start with data transfer phase. If TRANSFER COUNT is zero, no data transfer phase occurs. In either case, a chain to another combination command can occur if enabled.

WAIT-FOR-SELECT-AND-RECEIVE (0C HEX)

The Wait-for-Select-and-Receive causes the 33C93A to idle until it is selected by an Initiator, at which time the 33C93A will enter the Target mode and message and command information will automatically be requested. As an option, the 33C93A may be programmed to disconnect when a SCSI read command is received while executing a Wait-for-Select-and-Receive command. Use of this command therefore eliminates the interrupts which normally occur after selection and after each subsequent SCSI bus phase, and results in very short bus-connect time during SCSI read commands.

If $\overline{ATN}$ was asserted by the Initiator during the selection phase, the 33C93A will first execute an implied "Receive Message Out" command to get the Identify message from the Initiator, before continuing on with the implied "Receive Command" to receive the SCSI command information. The SCSI command information (CDB) will be stored in the CDB registers (hex addresses 03 to 0E), and if a valid IDENTIFY message is received, it will be saved in the TARGET LUN register (hex address 0F). The number of command bytes requested by the 33C93A is determined by the SCSI group code in the first byte of the CDB.

After the 33C93A is selected and receives all valid command and message information, a "successful completion" interrupt will normally be generated to allow the local processor to read out and interpret the SCSI CDB. However, by setting the EDI bit prior to issuing a Wait-for-Select-and-Receive command, the 33C93A is enabled to perform an automatic disconnect when a SCSI read command is received. Therefore, when EDI=1 and the 1st CDB byte received contains a 6-, 10-, or 12-byte read command code, then the 33C93A will temporarily suppress the interrupt and chain to begin execution of a Send-Disconnect-Message command. An interrupt will then be generated after completion of this command, which normally would indicate a transition to the bus free condition. Refer to the Send-Disconnect-Message command description for more details.

If during execution the message or command information received from the Initiator is invalid, the implied receive command will be terminated and the appropriate status reported. In this case, the COMMAND PHASE register should be read to determine which phase of the Wait-for-Select-and-Receive command was last completed before the error condition occurred. A COMMAND PHASE hex value of hex 10 indicates that the 33C93A was successfully selected. A hex value of 20 indicates that a message was received from the Initiator, and when the 33C93A begins receiving command bytes, the COMMAND PHASE is set to hex 30 and increments with each byte received (to a maximum of 3C for a 12-byte CDB command).

The following table summarizes the possible values that the COMMAND PHASE register can take during the Wait-for-Select-and-Receive command, and their meanings relative to command termination. See other command descriptions for additional values that can occur when command chaining is used.

Command Phase	Meaning
00	The 33C93A has not been selected. The 33C93A is in the disconnected state.
10	The 33C93A has been successfully selected by the Initiator. The 33C93A is now in the connected as a Target state.
20	The Identify message has been successfully received from the Initiator.
30	The 33C93A has begun command phase by setting the SCSI bus phase signals and asserting $\overline{REQ}$.
31	The 33C93A has transferred one command byte from the Initiator. The SCSI STATUS may indicate the need for the host to load the command size into the OWN ID register.
3x	The 33C93A has transferred x command bytes from the Initiator.

A "Resume Wait-for-Select-and-Receive" command is assumed whenever a normal "Wait-for-Select-and-Receive" command is issued while the 33C93A is in the Connected-as-a-Target state. When the "Resume" is issued, the 33C93A examines the COMMAND PHASE register to determine where to restart the Wait-for-Select-and- Receive command execution. This feature, in conjunction with the capability to chain to other combination commands, allows longer SCSI bus sequences to be executed by a single command.

The following table briefly describes the meaning of the COMMAND PHASE register when resuming a Wait-for-Select-and-Receive command:

Command Phase	Meaning
10	Resume after selection by the Initiator is complete; start with Identify Message Out if ATN is asserted, otherwise, start with command phase.
20	Resume after a message out; check the received message in the TARGET LUN register for a valid Identify message.
30	Resume after Identify message out. Start with command phase.
31	Resume after the 33C93A has transferred 1 command byte from the Initiator. This resume point is used only when an unknown group code has been detected in Advanced Mode, and the command size has been loaded into the OWN ID register.

SEND-STATUS-AND-COMMAND-COMPLETE (0D HEX)

The Send-Status-and-Command-Complete command is valid in the Target role, and is used to complete a SCSI operation by transferring the appropriate status information to the Initiator prior to disconnection from the SCSI bus. This command also supports linked SCSI operations by optionally allowing a linked command-complete message to be sent after the status is transferred. Linked command complete messages are controlled by the CDB12 register with bits that correspond to the standard linked command control bits in the CDB.

Before a Send-Status-and-Command-Complete command is issued, the CDB11 register must be loaded with a status byte which will then be transferred across the SCSI bus. Also, the link control bits from the current CDB must be loaded into the CDB12 register to ensure that the correct sequence occurs. Note that the bits used by the 33C93A are identical in meaning to the SCSI standard link control bits. The host processor may simply load the control byte from the current SCSI command into CDB12 to get the correct function. As the command execution progresses, the COMMAND PHASE register will be updated to indicate the last phase completed.

The possible sequences caused by this command are as follows:

1. CDB12 bit0=0, bit1=don't care: The status byte in CDB11 is sent, followed by a Command Complete message (00 hex). A "successful completion" interrupt now occurs.
2. CDB12 bit0=1, bit1=0: The status byte in CDB11 is sent, followed by a Linked Command Complete message (0A hex). A chain to the command fetch portion of Wait-for-Select-and-Receive then occurs to fetch the next CDB from the Initiator. 33C93A command execution proceeds as described for that command.
3. CDB12 bit0=1, bit1=1: The status byte in CDB11 is sent, followed by a Linked Command Complete with Flag message (0B hex). A chain to the command fetch portion of Wait-for-Select-and-Receive then occurs to fetch the next CDB from the Initiator. 33C93A command execution proceeds as described for that command.

A Send-Status-and-Command-Complete command may be terminated by $\overline{ATN}$ asserted when HA=1, or when a Disconnect or Reset command is issued.

The following table summarizes the possible values that the COMMAND PHASE register can take during the Send-Status-and-Command-Complete command, and their meanings relative to command termination. See other command descriptions for additional values that can occur when command chaining is used.

Command Phase	Meaning
00	No operation occurred; typically, $\overline{ATN}$ was found to be asserted.
50	Status phase transfer completed.
60	Command Complete message transfer completed.
61	Linked Command Complete message transfer completed.

A "Resume Send-Status-and-Command-Complete" command is assumed whenever a normal "Send-Status-and-Command-Complete" command is issued while the 33C93A is in the Connected-as-a-Target state. When the "Resume" is issued, the 33C93A examines the COMMAND PHASE register to determine where to restart the Send-Status-and-Command-Complete command execution. This feature, in conjunction with the capability to chain to other combination commands, allows longer SCSI bus sequences to be executed by a single command.

The following table briefly describes the meaning of the COMMAND PHASE register when resuming a Send-Status-and-Command-Complete command:

Command Phase	Meaning
50	Resume after status phase. Start with command complete message. May chain to command fetch if commanded to do so.

SEND-DISCONNECT-MESSAGE (0E HEX)

The Send-Disconnect-Message command is a Target-role command which may be used to disconnect from the SCSI bus at any time during a SCSI command sequence. This command consists of sending a Disconnect message byte, followed by physical disconnection from the bus (SCSI bus free). An interrupt is generated only after transition to bus free occurs. As an option, a Save-Data-Pointer message will automatically be sent before the Disconnect message whenever the IDI bit is set prior to issuing this command.

The COMMAND PHASE register is updated during execution of the Send-Disconnect-Message command to indicate bus phase status. After a Save-Data-Pointer message is sent, the COMMAND PHASE will be set to 41H. After the Disconnect message transfer, this register will be updated to 42H, and after disconnection the COMMAND PHASE register will contain a 43H.

A Send-Disconnect-Message command may be terminated by $\overline{ATN}$ asserted when HA=1, or when a Disconnect or Reset command is issued.

The following table summarizes the possible values that the COMMAND PHASE register can take during the Send-Disconnect-Message, and their meanings relative to command termination. See other command descriptions for additional values that can occur when command chaining is used.

Command Phase	Meaning
00	No operation occurred; typically, $\overline{ATN}$ was found to be asserted.
41	The Save-Data-Pointer message was transferred.
42	The Disconnect message was transferred.
43	The bus free state occurred after the Disconnect message was transferred. The 33C93A is now in the disconnected state.

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to GND	–0.5 V to +7.0 V
Operating temperature	0 to 70 deg. C
Storage temperature	–55 to +125 deg. C
Power dissipation	500 mW
Input Static Discharge Protection	2000 V pin to pin

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolutely maximum ratings for extended periods may affect device reliability.

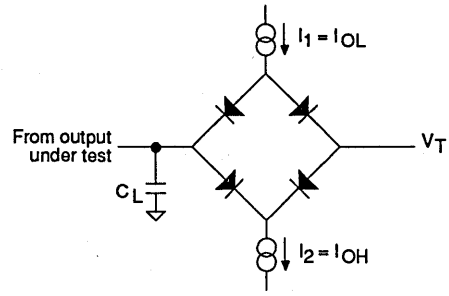
DC OPERATING CHARACTERISTICS

Ta = 0 to 70°C, VCC = +5 V ±0.25 V, GND = 0 V

Symbol	Characteristic	Min	Max	Units	Conditions
IIL	Input Leakage		10	µA	VIN =. 4 to VCC
IOL1	SCSI Output Leakage (Inactive)		50	µA	VOUT = .5 to VCC
IOL2	Output Leakage (Tri-State)		10	µA	VOUT = .4 to VCC
VIH	Input High Voltage	2.0		V	
VIL	Input Low Voltage		0.8	V	
VIHYS	Schmitt Trigger Input Hysteresis (All SCSI Pins)	0.2		V	
VOH	Output High Voltage	2.4		V	IO = –400 µA
VOL1	SCSI Output Low Voltage		0.5	V	IO = 48.0 mA
VOL2	Output Low Voltage (All Others)		0.4	V	IO = 4.0 mA
ICC	Supply Current		20	mA	Ta = +25°C

PRELIMINARY

SWITCHING TEST CIRCUIT

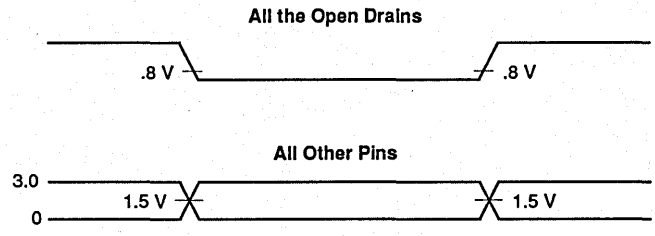


Pins of the Device	*C_L	I_1	I_2
INTRQ	100 pF	4 mA	400 µA
D0–D7, DP	100 pF	4 mA	400 µA
DACK, DRQ	100 pF	4 mA	—
SD0–SD7, SDP			
BSY, SEL, I/O, C/D	100 pF	20 mA	—
MSG, ATN, REQ, ACK			

PRELIMINARY

* Actual capacitance may vary ±20%.

SWITCHING TEST WAVEFORM



TIMING CHARACTERISTICS

Timing characteristics are valid over the entire operating temperature (0 to 70°C) and voltage (4.75 to 5.25 V) ranges, and are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 Volts. All outputs are assumed to have a load capacitance of 100 picofarads.

Many of the timing parameters that follow are defined in terms of an internal clock cycle time that is determined by the input clock and the clock divisor selected in the OWN ID register. This cycle time is calculated as follows:

$$Tcyc = \frac{Tckin \bullet DIVISOR}{2}$$

Where:

Tcyc is the internal clock cycle time;

Tckin is the period of the clock at the CLK input;

DIVISOR is the clock divisor selected in the OWN ID register.

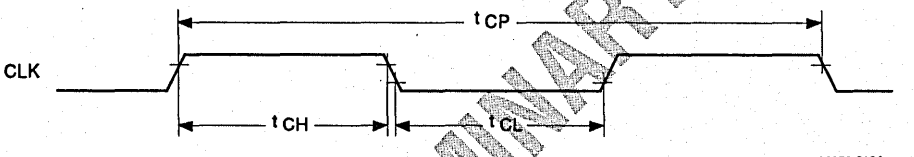
For example, with a 16 MHz clock input to the 33C93A, the clock divisor selected would be 4. Therefore, the value of Tcyc would be:

$$Tcyc = \frac{62.5\ ns \bullet 4}{2} = 125\ ns$$

Processor/DMA Interface

CLK

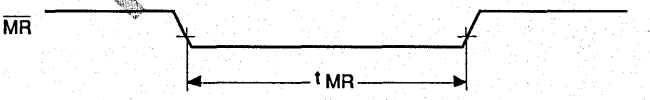
Symbol	Characteristic		Min	Max	Units
t_{CP}	Clock Period	16 MHz	62.5	125	ns
		20 MHz	50.0		ns
t_{CH}	Clock High	16 MHz	28.0		ns
		20 MHz	20.0		ns
t_{CL}	Clock Low	16 MHz	28.0		ns
		20 MHz	20.0		ns



11853-013A

$\overline{MR}$

Symbol	Characteristic	Min	Max	Units
t_{MR}	$\overline{MR}$ Pulse Width	1		µs



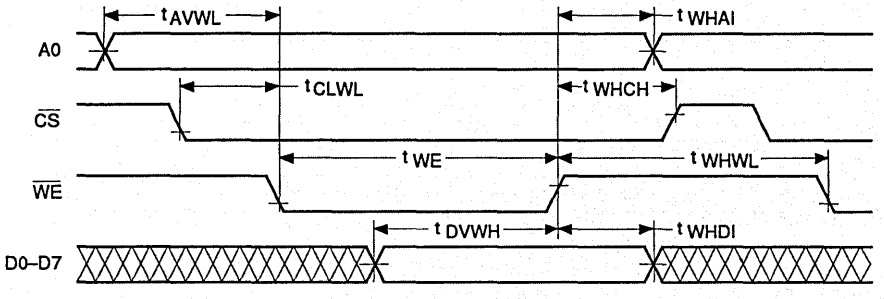
11853-014A

PRELIMINARY

PROCESSOR WRITE—INDIRECT ADDRESSING MODE

Symbol	Characteristic	Min	Max	Units
t_{AVWL}	A0 Valid to $\overline{WE}$ Low	0		ns
t_{CLWL}	$\overline{CS}$ Low to $\overline{WE}$ Low	0		ns
t_{WE}	$\overline{WE}$ Pulse Width	120		ns
t_{DVWH}	Data Valid to $\overline{WE}$ High	70		ns
t_{WHAI}	$\overline{WE}$ High to A0 Invalid	0		ns
t_{WHCH}	$\overline{WE}$ High to $\overline{CS}$ High	0		ns
t_{WHDI}	$\overline{WE}$ High to Data Invalid	0		ns
t_{WHWL}	$\overline{WE}$ High to $\overline{WE}$ or $\overline{RE}$ Low	100		ns

PRELIMINARY

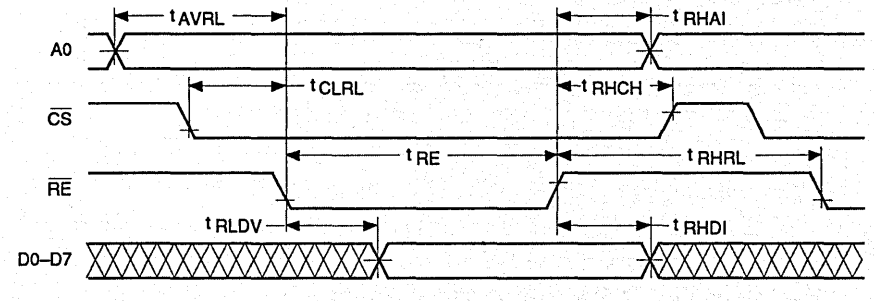


11853-015A

PROCESSOR READ—INDIRECT ADDRESSING MODE

Symbol	Characteristic	Min	Max	Units
t_{AVRL}	A0 Valid to $\overline{RE}$ Low	0		ns
t_{CLRL}	$\overline{CS}$ Low to $\overline{RE}$ Low	0		ns
t_{RE}	$\overline{RE}$ Pulse Width	180	10000	ns
t_{RLDV}	$\overline{RE}$ Low to Data Valid		180	ns
t_{RHCH}	$\overline{RE}$ High to $\overline{CS}$ High	0		ns
t_{RHDI}	$\overline{RE}$ High to Data Invalid	5	40	ns
t_{RHRL}	$\overline{RE}$ High to $\overline{RE}$ or $\overline{WE}$ Low	100		ns
t_{RHAI}	$\overline{RE}$ High to A0 Invalid	0		ns

PRELIMINARY

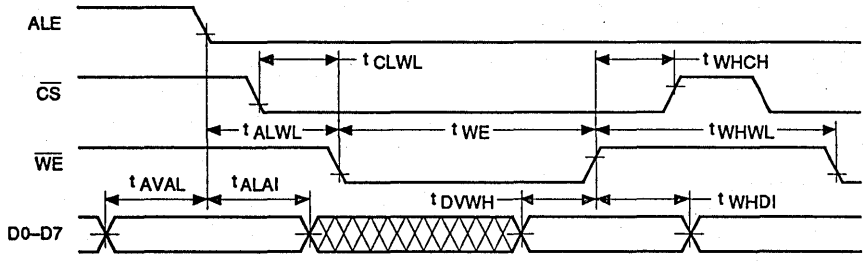


11853-016A

PROCESSOR WRITE—DIRECT ADDRESSING MODE

Symbol	Characteristic	Min	Max	Units
t_{AVAL}	ADDR Valid to ALE Low	40		ns
t_{ALAI}	ALE Low to ADDR Invalid	0		ns
t_{AHAL}	ALE High to ALE Low	50	1000	ns
t_{ALWL}	ALE Low to $\overline{WE}$ Low	90		ns
t_{CLWL}	$\overline{CS}$ Low to $\overline{WE}$ Low	0		ns
t_{WE}	$\overline{WE}$ Pulse Width	120		ns
t_{DVWH}	Data Valid to $\overline{WE}$ High	70		ns
t_{WHCH}	$\overline{WE}$ High to $\overline{CS}$ High	0		ns
t_{WHDI}	$\overline{WE}$ High to Data Invalid	0		ns
t_{WHWL}	$\overline{WE}$ High to $\overline{WE}$ or $\overline{RE}$ Low	100		ns

PRELIMINARY

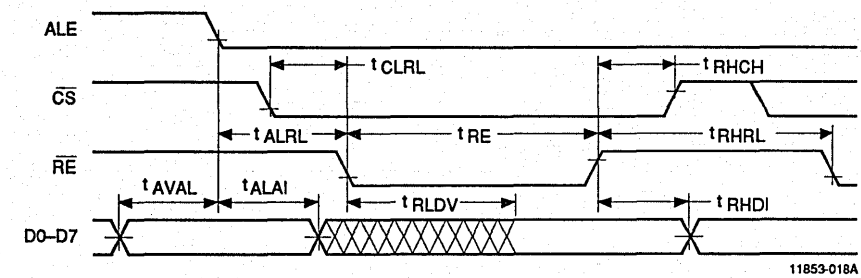


11853-017A

PROCESSOR READ—DIRECT ADDRESSING MODE

Symbol	Characteristic	Min	Max	Units
t_{AVAL}	ADDR Valid to ALE Low	40		ns
t_{ALAI}	ALE Low to ADDR Invalid	0		ns
t_{AHAL}	ALE High to ALE Low	50	1000	ns
t_{ALRL}	ALE Low to $\overline{RE}$ Low	30		ns
t_{CLRL}	$\overline{CS}$ Low to $\overline{RE}$ Low	0		ns
t_{RE}	$\overline{RE}$ Pulse Width	180	10000	ns
t_{RLDV}	$\overline{RE}$ Low to Data Valid		180	ns
t_{RHCH}	$\overline{RE}$ High to $\overline{CS}$ High	0		ns
t_{RHDI}	$\overline{RE}$ High to Data Invalid	5	40	ns
t_{RHRL}	$\overline{RE}$ High to $\overline{RE}$ or $\overline{WE}$ Low	100		ns

PRELIMINARY

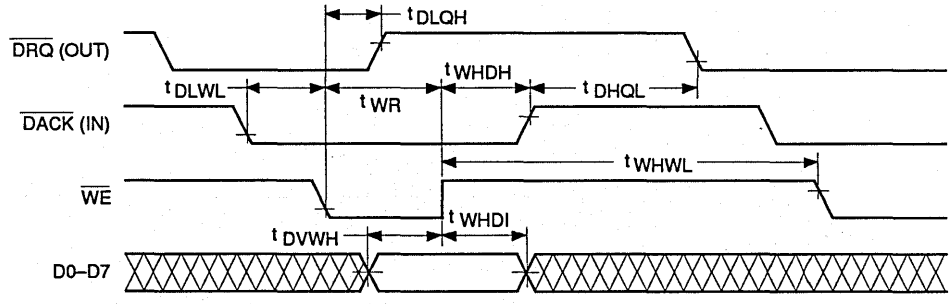


11853-018A

DMA WRITE

Symbol	Characteristic	Min	Max	Units
t_{DLWL}	$\overline{DACK}$ Low to $\overline{WE}$ Low	0		ns
t_{DLQH}	$\overline{DACK}$ Low to $\overline{DRQ}$ High		75	ns
t_{WR}	$\overline{WE}$ Pulse Width	50		ns
t_{WHWL}	$\overline{WE}$ High to $\overline{WE}$ Low	100		ns
t_{DVWH}	Data Valid to $\overline{WE}$ High	25		ns
t_{WHDH}	$\overline{WE}$ High to $\overline{DACK}$ High	0		ns
t_{WHDI}	$\overline{WE}$ High to DATA Invalid	0		ns
t_{DHQL}	$\overline{DACK}$ High to $\overline{DRQ}$ Low	0		ns

PRELIMINARY



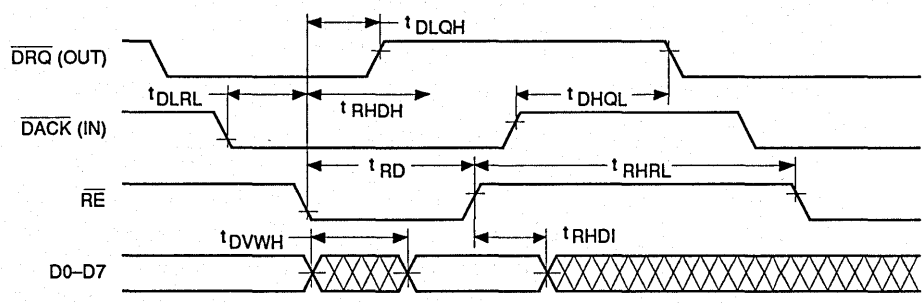
NOTE: External load on $\overline{DRQ}$ & $\overline{DACK}$ is assumed to be 1K Ω.

11853-019A

DMA READ

Symbol	Characteristic	Min	Max	Units
t_{DLRL}	$\overline{DACK}$ Low to $\overline{RE}$ Low	0		ns
t_{DLQH}	$\overline{DACK}$ Low to $\overline{DRQ}$ High		75	ns
t_{RD}	$\overline{RE}$ Pulse Width	80		ns
t_{RHRL}	$\overline{RE}$ High to $\overline{RE}$ Low	100		ns
t_{RLDV}	$\overline{RE}$ Low to Data Valid		70	ns
t_{RHDH}	$\overline{RE}$ High to $\overline{DACK}$ High	0		ns
t_{RHDI}	$\overline{RE}$ High to DATA Invalid	5	40	ns
t_{DHQL}	$\overline{DACK}$ High to $\overline{DRQ}$ Low	0		ns

PRELIMINARY

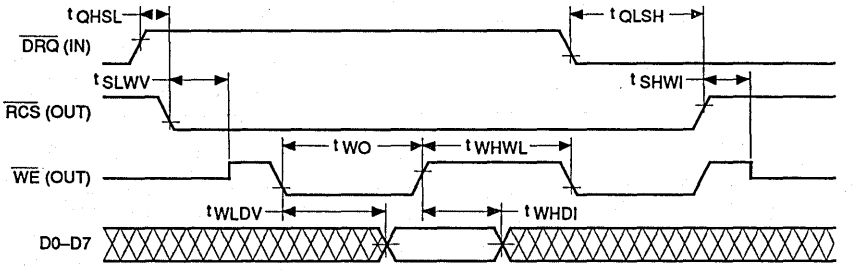


NOTE: External load on $\overline{DRQ}$ & $\overline{DACK}$ is assumed to be 1K Ω.

11853-020A

DIRECT BUFFER ACCESS WRITE

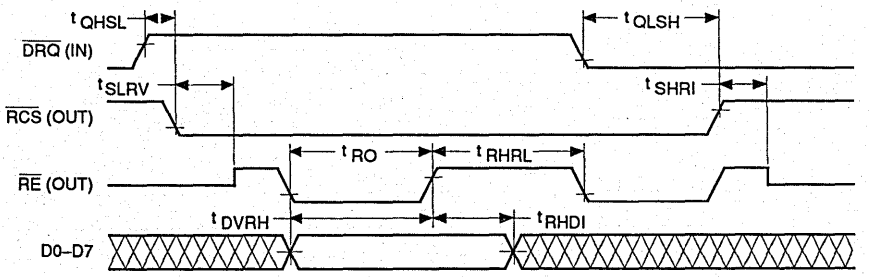
Symbol	Characteristic	Min	Max	Units
t_{QHSL}	DRQ High to $\overline{RCS}$ Low	0	40	ns
t_{SLWV}	$\overline{RCS}$ Low to $\overline{WE}$ Valid	0	20	ns
t_{WO}	$\overline{WE}$ Pulse Width	Tcyc-20		ns
t_{WLDV}	$\overline{WE}$ Low to Data Valid		50	ns
t_{WHDI}	$\overline{WE}$ High to Data Invalid	20		ns
t_{WHWL}	$\overline{WE}$ High to $\overline{WE}$ Low	Tcyc-20		ns
t_{QLSH}	DRQ Low to $\overline{RCS}$ High	8*Tcyc	10*Tcyc	ns
t_{SHWI}	$\overline{RCS}$ High to $\overline{WE}$ Invalid		100	ns
t_{WHSH}	$\overline{WE}$ High to $\overline{RCS}$ High	0		ns
t_{SLWL}	$\overline{RCS}$ Low to $\overline{WE}$ Low	60		ns
t_{QLWL}	DRQ Low to $\overline{WE}$ Low (1)	0		ns
t_{SLQL}	$\overline{RCS}$ Low to DRQ Low (2)		100	ns



11853-021A

DIRECT BUFFER ACCESS READ

Symbol	Characteristic	Min	Max	Units
t_{QHSL}	DRQ High to $\overline{RCS}$ Low	0	40	ns
t_{SLRV}	$\overline{RCS}$ Low to $\overline{RE}$ Valid	0	20	ns
t_{RO}	$\overline{RE}$ Pulse Width	Tcyc-20		ns
t_{DVRH}	Data Valid to $\overline{RE}$ High	20		ns
t_{RHDI}	$\overline{RE}$ High to Data Invalid	0		ns
t_{RHRL}	$\overline{RE}$ High to $\overline{RE}$ Low	Tcyc-20		ns
t_{QLSH}	DRQ Low to $\overline{RCS}$ High	8*Tcyc	10*Tcyc	ns
t_{SHRI}	$\overline{RCS}$ High to $\overline{RE}$ Invalid		100	ns
t_{RHSH}	$\overline{RE}$ High to $\overline{RCS}$ High	0		ns
t_{SLRL}	$\overline{RCS}$ Low to $\overline{RE}$ Low	60		ns
t_{QLRL}	DRQ Low to $\overline{RE}$ Low (1)	0		ns
t_{SLQL}	$\overline{RCS}$ Low to DRQ Low (2)		100	ns



11853-022A

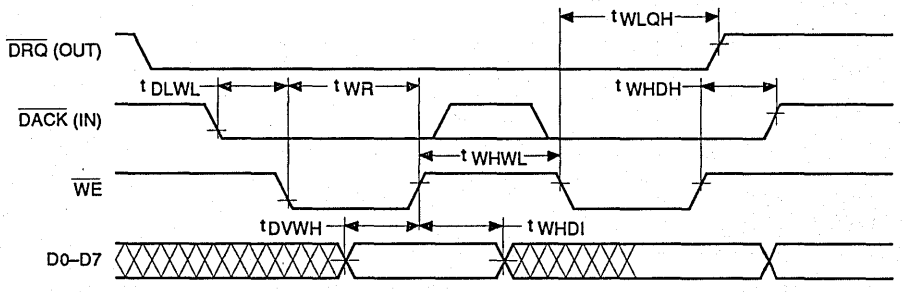
Note 1: Guaranteed that one more byte will be transferred.
Note 2: Guaranteed that only one byte will be transferred.

PRELIMINARY

BURST DMA WRITE

Symbol	Characteristic	Min	Max	Units
t_{DLWL}	$\overline{DACK}$ Low to $\overline{WE}$ Low	0		ns
t_{WLQH}	$\overline{WE}$ Low to $\overline{DRQ}$ High		75	ns
t_{WR}	$\overline{WE}$ Pulse Width	50		ns
t_{WHWL}	$\overline{WE}$ High to $\overline{WE}$ Low	80		ns
t_{DVWH}	Data Valid to $\overline{WE}$ High	25		ns
t_{WHDH}	$\overline{WE}$ High to $\overline{DACK}$ High	0		ns
t_{WHDI}	$\overline{WE}$ High to Data Invalid	0		ns

PRELIMINARY

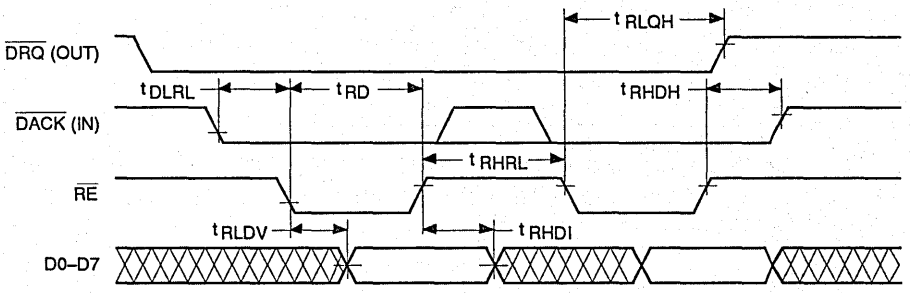


11853-023A

BURST DMA READ

Symbol	Characteristic	Min	Max	Units
t_{DLRL}	$\overline{DACK}$ Low to $\overline{RE}$ Low	0		ns
t_{RLQH}	$\overline{RE}$ Low to $\overline{DRQ}$ High		75	ns
t_{RD}	$\overline{RE}$ Pulse Width	80		ns
t_{RHRL}	$\overline{RE}$ High to $\overline{RE}$ Low	80		ns
t_{RLDV}	$\overline{RE}$ Low to Data Valid		50	ns
t_{RHDH}	$\overline{RE}$ High to $\overline{DACK}$ High	0		ns
t_{RHDI}	$\overline{RE}$ High to Data Invalid	5	40	ns

PRELIMINARY

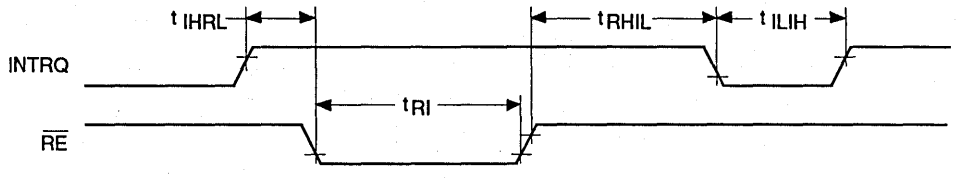


11853-024A

INTRQ

Symbol	Characteristic	Min	Max	Units
t_{IHRL}	INTRQ High to $\overline{RE}$ Low	0		ns
t_{RI}	$\overline{RE}$ Pulse Width	180		ns
t_{RLIL}	$\overline{RE}$ Low to INTRQ Low	0	100	ns
t_{ILIH}	INTRQ Low to INTRQ High	100		ns

PRELIMINARY



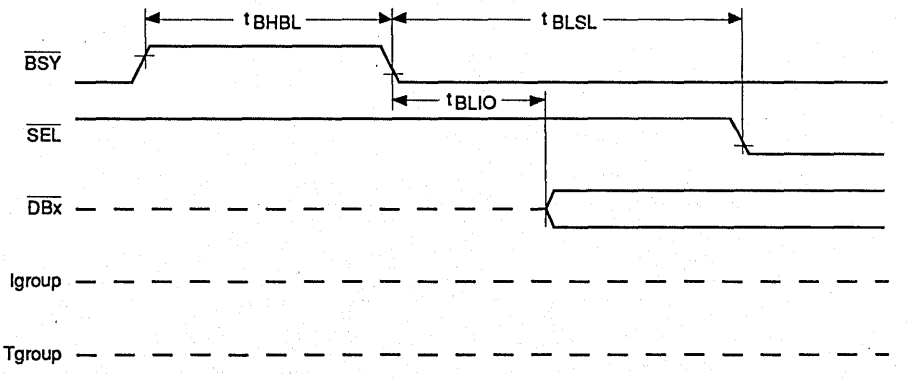
11853-025A

SCSI INTERFACE

ARBITRATION

Symbol	Characteristic	Min	Max	Units
t_{BHBL}	$\overline{BSY}$, $\overline{SEL}$ In High to $\overline{BSY}$ Out Low	12*Tcyc	16*Tcyc	ns
t_{BLIO}	$\overline{BSY}$ Out Low to BUS ID Out	50	50	ns
t_{BLSL}	$\overline{BSY}$ Out Low to $\overline{SEL}$ Out Low	2.2		µs

PRELIMINARY



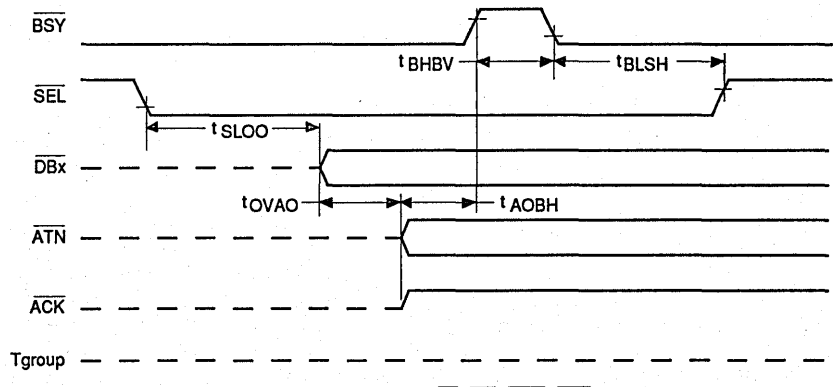
NOTE: Tgroup = signals driven by a Target = $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$
Igroup = signals driven by an Initiator = $\overline{ATN}$, $\overline{ACK}$

11853-026A

SELECTING A TARGET (AS AN INITIATOR)

Symbol	Characteristic	Min	Max	Units
t_{SLOO}	$\overline{SEL}$ Out Low to "OR-ED" ID Out	1.2		us
t_{OVAO}	"OR-ED" ID Out Valid to $\overline{ACK}$, $\overline{ATN}$ Out	100		ns
t_{AOBH}	$\overline{ACK}$, $\overline{ATN}$ Out Valid to $\overline{BSY}$ Out High	100		ns
t_{BHBV}	$\overline{BSY}$ Out High to $\overline{BSY}$ In Low Valid	400		ns
t_{BLSH}	$\overline{BSY}$ In Low to $\overline{SEL}$ Out High	100		ns

PRELIMINARY



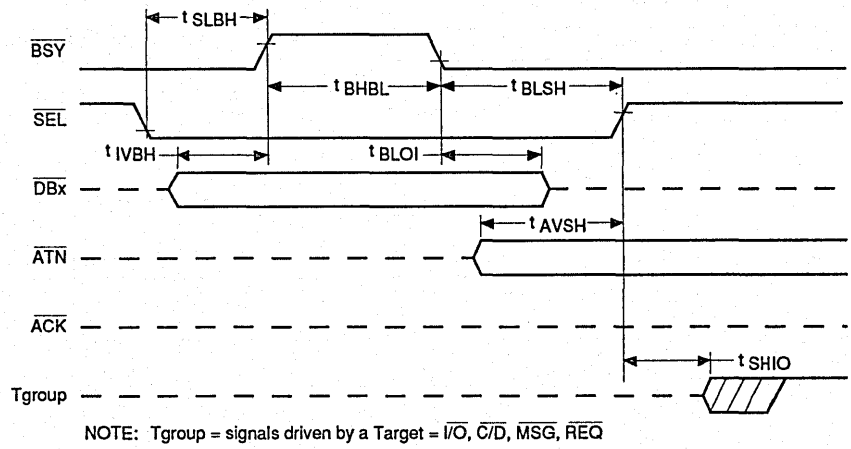
NOTE: Tgroup = signals driven by a Target = $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$

11853-027A

RESPONSE TO SELECTION (AS A TARGET)

Symbol	Characteristic	Min	Max	Units
t_{SLBH}	$\overline{SEL}$ In Low to $\overline{BSY}$ In High	0		ns
t_{IVBH}	"OR-ED" ID Valid In to $\overline{BSY}$ In High	0		ns
t_{BHBL}	$\overline{SEL}$ Low , ID Valid , $\overline{BSY}$ High to $\overline{BSY}$ Low	0.4	200	us
t_{BLOI}	$\overline{BSY}$ Out Low to "OR-ED" ID Invalid In	0		ns
t_{BLSH}	$\overline{BSY}$ Out Low to $\overline{SEL}$ In High	0		ns
t_{AVSH}	$\overline{ATN}$ Valid In to $\overline{SEL}$ In High	0		ns
t_{SHIO}	$\overline{SEL}$ In High to Tgroup Out	100		ns

PRELIMINARY



NOTE: Tgroup = signals driven by a Target = $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$

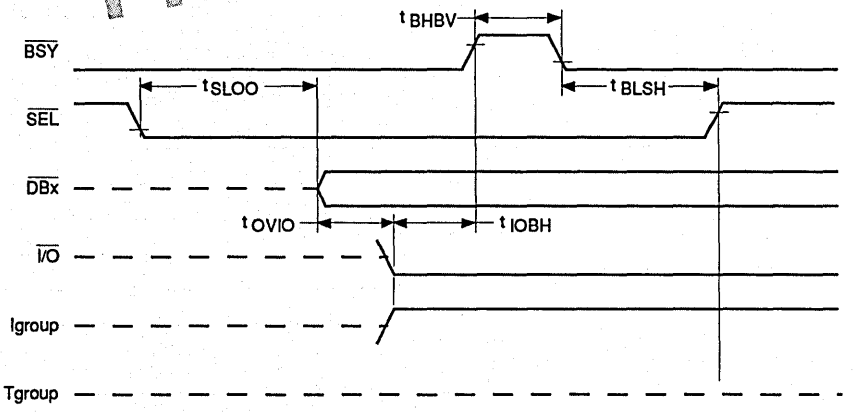
11853-028A

4

RESELECTING AN INITIATOR (AS A TARGET)

Symbol	Characteristic	Min	Max	Units
t_{SLOO}	$\overline{SEL}$ Out Low to "OR-ED" ID Out	1.2		µs
t_{OVIO}	"OR-ED" ID Out Valid to I/O & Tgroup Out Valid	100		ns
t_{IOBH}	I/O & Tgroup Out Valid to $\overline{BSY}$ Out High	100		ns
t_{BHBV}	$\overline{BSY}$ Out High to $\overline{BSY}$ In Low Valid	400		ns
t_{BLSH}	$\overline{BSY}$ In Low to $\overline{SEL}$ Out High	100		ns

PRELIMINARY



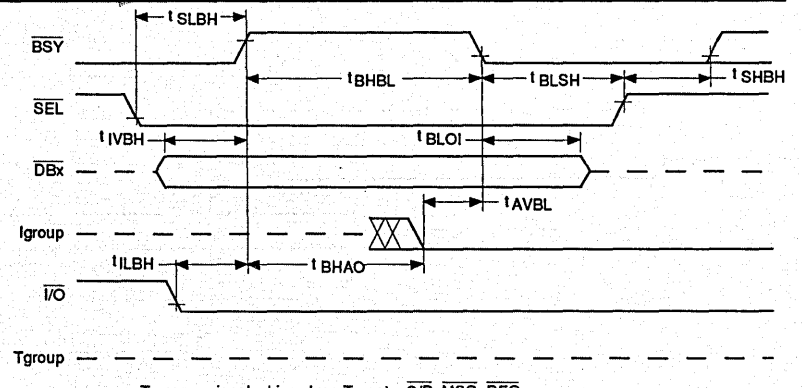
NOTE: Tgroup = signals driven by a Target = $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$
Igroup = signals driven by an Initiator = $\overline{ATN}$, $\overline{ACK}$

11853-029A

RESPONSE TO RESELECTION (AS AN INITIATOR)

Symbol	Characteristic	Min	Max	Units
t_{SLBH}	$\overline{SEL}$ In Low to $\overline{BSY}$ In High	0		ns
t_{IVBH}	"OR-ED" ID Valid In to $\overline{BSY}$ In High	0		ns
t_{ILBH}	$\overline{I/O}$ In Low to $\overline{BSY}$ In High	0		ns
t_{BHAO}	$\overline{SEL}$ Low , ID Valid , $\overline{BSY}$ High to Igroup Out	100		ns
t_{AVBL}	Igroup Valid Out to $\overline{BSY}$ Out Low	100		ns
t_{BHBL}	$\overline{BSY}$ In High to $\overline{BSY}$ Out Low	0.4	200	µs
t_{BLOI}	$\overline{BSY}$ Out Low to "OR-ED" ID Invalid In	0		ns
t_{BLSH}	$\overline{BSY}$ Out Low to $\overline{SEL}$ In High	0		ns
t_{SHBH}	$\overline{SEL}$ In High to $\overline{BSY}$ Out High	0		ns

PRELIMINARY



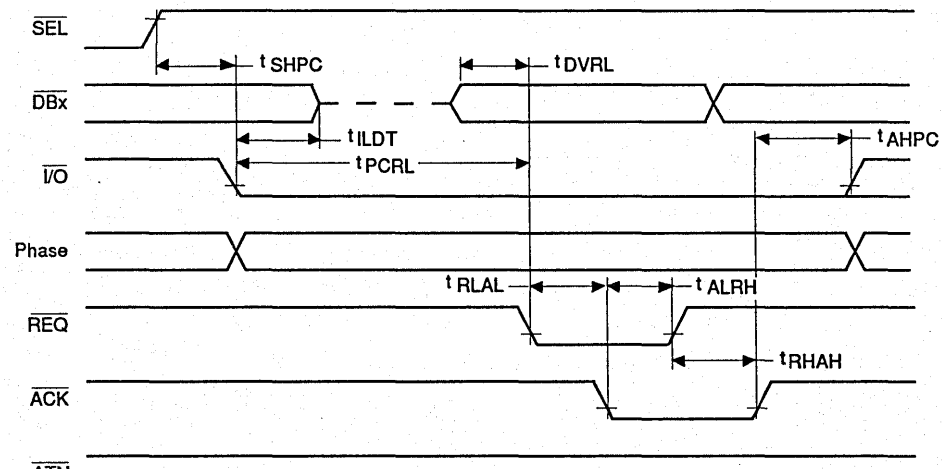
Tgroup = signals driven by a Target = $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$
Igroup = signals driven by an Initiator = $\overline{ATN}$, $\overline{ACK}$
*** $\overline{BSY}$ will still be driven by the reselecting target.

11853-030A

RECEIVE ASYNCHRONOUS INFORMATION TRANSFER IN (ACTING AS AN INITIATOR)

Symbol	Characteristic	Min	Max	Units
t_{SHPC}	$\overline{SEL}$ In High to Phase Change In	0		ns
t_{ILDT}	$\overline{I/O}$ In Low to Data Bus TRISTATE	0	125	ns
t_{PCRL}	Phase Change In to $\overline{REQ}$ In Low	400		ns
t_{DVRL}	Data Valid In to $\overline{REQ}$ In Low	0		ns
t_{RLAL}	$\overline{REQ}$ In Low to $\overline{ACK}$ Out Low	0	175	ns
t_{ALDI}	$\overline{ACK}$ Out Low to Data Invalid In	0		ns
t_{ALRH}	$\overline{ACK}$ Out Low to $\overline{REQ}$ In High	0		ns
t_{RHAH}	$\overline{REQ}$ In High to $\overline{ACK}$ Out High	0	175	ns
t_{AHPC}	$\overline{ACK}$ Out High to Phase Change In	0		ns

PRELIMINARY



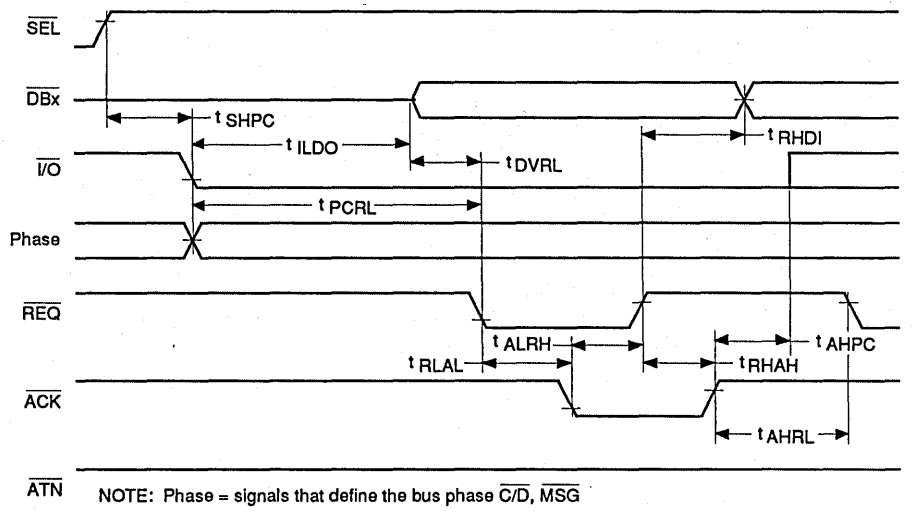
NOTE: Phase = signals that define the bus phase $\overline{C/D}$, $\overline{MSG}$

11853-031A

SEND ASYNCHRONOUS INFORMATION TRANSFER IN (ACTING AS A TARGET)

Symbol	Characteristic	Min	Max	Units
t_{SHPC}	$\overline{SEL}$ In High to Phase Change Out	100		ns
t_{ILDO}	$\overline{I/O}$ Out Low to Data Out	800		ns
t_{DVRL}	Data Out Valid to $\overline{REQ}$ Out Low	55		ns
t_{PCRL}	Phase Change Out to $\overline{REQ}$ Out Low	500		ns
t_{RLAL}	$\overline{REQ}$ Out Low to $\overline{ACK}$ In Low	0		ns
t_{ALRH}	$\overline{ACK}$ In Low to $\overline{REQ}$ Out High	0	175	ns
t_{ALDI}	$\overline{ACK}$ In Low to Data Out Invalid	0		ns
t_{RHAH}	$\overline{REQ}$ Out High to $\overline{ACK}$ In High	0		ns
t_{AHPC}	$\overline{ACK}$ In High to Phase Change Out	100		ns
t_{AHRL}	$\overline{ACK}$ In High to $\overline{REQ}$ Out Low	0	175	ns

PRELIMINARY



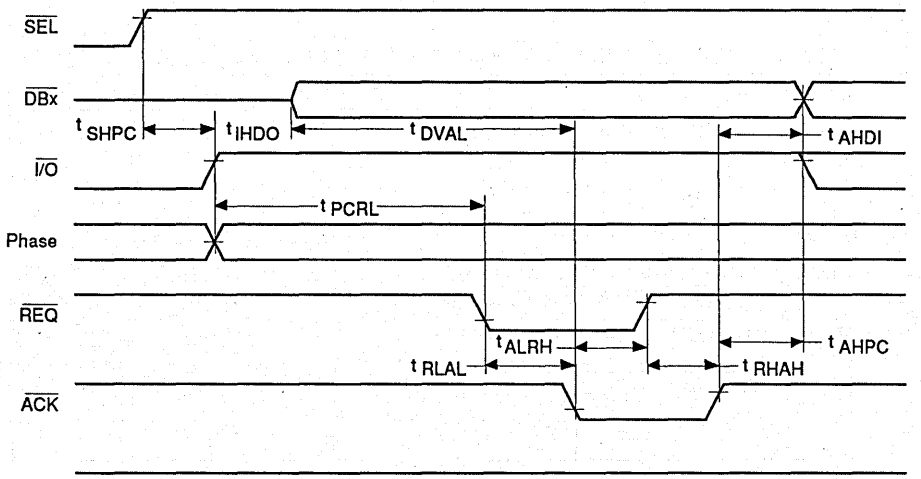
NOTE: Phase = signals that define the bus phase C/D, MSG

11853-032A

SEND ASYNCHRONOUS INFORMATION TRANSFER OUT (ACTING AS AN INITIATOR)

Symbol	Characteristic	Min	Max	Units
t_{SHPC}	SEL In High to Phase Change In	0		ns
t_{IHDO}	I/O In High to Data Out	0		ns
t_{PCRL}	Phase Change In to REQ In Low	400		ns
t_{RLAL}	REQ In Low to ACK Out Low	0	175	ns
t_{DVAL}	Data Out Valid to ACK Out Low	55		ns
t_{ALRH}	ACK Out Low to REQ In High	0		ns
t_{RHAH}	REQ In High to ACK Out High	0	175	ns
t_{RHDI}	REQ In High to Data Out Invalid	0		ns
t_{AHPC}	ACK Out High to Phase Change In	0		ns

PRELIMINARY



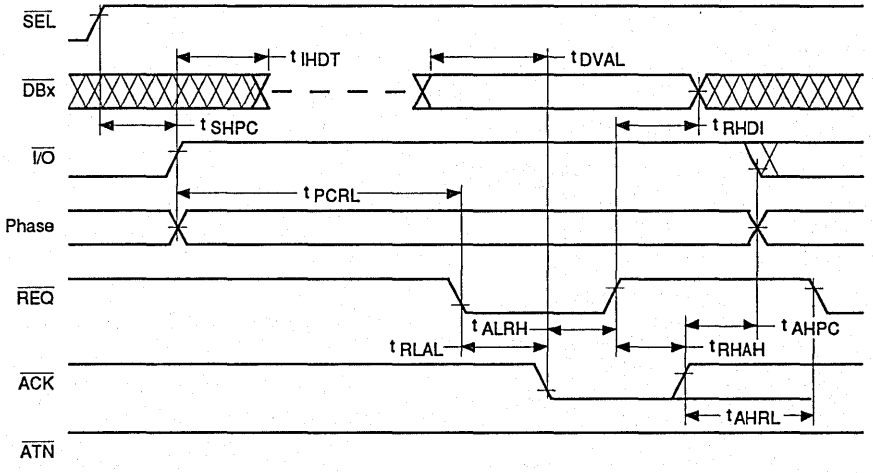
NOTE: Phase = signals that define the bus phase C/D, MSG

11853-033A

RECEIVE ASYNCHRONOUS INFORMATION TRANSFER OUT (ACTING AS A TARGET)

Symbol	Characteristic	Min	Max	Units
t_{SHPC}	$\overline{SEL}$ In High to Phase Change Out	100		ns
t_{IHDT}	$\overline{I/O}$ Out High to Data Bus TRISTATE	0		ns
t_{PCRL}	Phase Change to $\overline{REQ}$ Out Low	500		ns
t_{RLAL}	$\overline{REQ}$ Out Low to $\overline{ACK}$ In Low	0		ns
t_{DVAL}	Data In Valid to $\overline{ACK}$ In Low	0		ns
t_{ALRH}	$\overline{ACK}$ In Low to $\overline{REQ}$ Out High	0	175	ns
t_{RHDI}	$\overline{REQ}$ Out High to Data In Invalid	0		ns
t_{RHAH}	$\overline{REQ}$ Out High to $\overline{ACK}$ In High	0		ns
t_{AHPC}	$\overline{ACK}$ In High to Phase Change Out	0		ns
t_{AHRL}	$\overline{ACK}$ In High to $\overline{REQ}$ Out Low	0	175	ns

PRELIMINARY



NOTE: Phase = signals that define the bus phase $\overline{C/D}$, $\overline{MSG}$

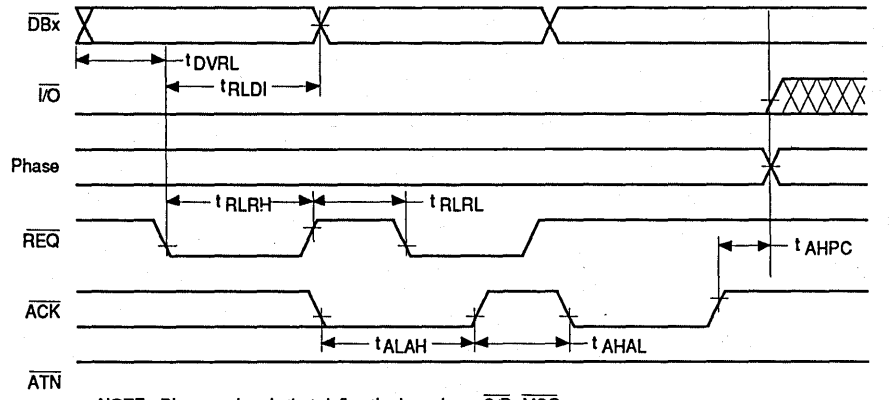
11853-034A

RECEIVE SYNCHRONOUS INFORMATION TRANSFER IN (ACTING AS AN INITIATOR)

Symbol	Characteristic	Min	Max	Units
t_{DVRL}	Data Valid In to $\overline{REQ}$ In Low	0		ns
t_{RLDI}	$\overline{REQ}$ In Low to DATA Invalid	45		ns
t_{RLRH}	$\overline{REQ}$ In Low to $\overline{REQ}$ In High	50		ns
t_{RHRL}	$\overline{REQ}$ In High to $\overline{REQ}$ In Low	50		ns
t_{ALAH}	$\overline{ACK}$ Out Low to $\overline{ACK}$ Out High	Tcyc-10		ns
t_{AHAL}	$\overline{ACK}$ Out High to $\overline{ACK}$ Out Low	Tcyc-25		ns
t_{AHPC}	$\overline{ACK}$ Out High to Phase Change	0		ns

PRELIMINARY

Parameters t_{SHPC}, t_{ILDT}, and t_{PCRL} are also applicable and are identical to those in Receive Asynchronous Information Transfer In (Acting as an Initiator), top of page 37.



NOTE: Phase = signals that define the bus phase C/D, MSG

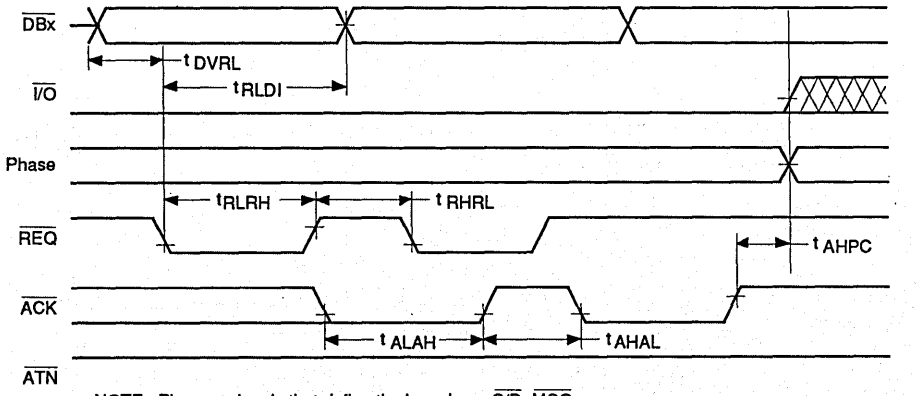
11853-035A

SEND SYNCHRONOUS INFORMATION TRANSFER IN (ACTING AS A TARGET)

Symbol	Characteristic	Min	Max	Units
t_{DVRL}	Data Valid Out to REQ Out Low	55		ns
t_{RLDI}	REQ Out Low to Data Invalid	100		ns
t_{RLRH}	REQ Out Low to REQ Out High	Tcyc-10		ns
t_{RHRL}	REQ Out High to REQ Out Low	Tcyc-25		ns
t_{ALAH}	ACK In Low to ACK In High	50		ns
t_{AHAL}	ACK In High to ACK In Low	50		ns
t_{AHPC}	ACK In High to Phase Change Out	0		ns

PRELIMINARY

Parameters t_{SHPC}, t_{ILDO}, and t_{PCRL} are also applicable and are identical to those in Receive Asynchronous Information Transfer In (Acting as a Target), bottom of page 37.



NOTE: Phase = signals that define the bus phase C/D, MSG

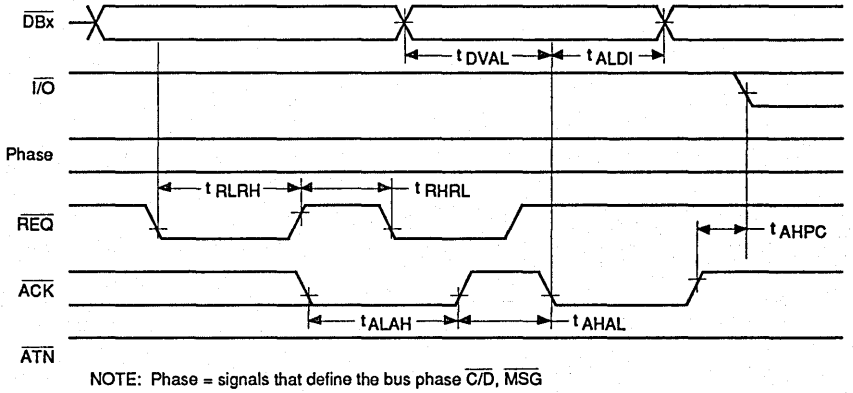
11853-036A

SEND SYNCHRONOUS INFORMATION TRANSFER OUT (ACTING AS AN INITIATOR)

Symbol	Characteristic	Min	Max	Units
t_{DVAL}	Data Valid Out to $\overline{ACK}$ Out Low	55		ns
t_{ALDI}	$\overline{ACK}$ Out Low to Data Invalid	100		ns
t_{RLRH}	$\overline{REQ}$ In Low to $\overline{REQ}$ In High	50		ns
t_{RHRL}	$\overline{REQ}$ In High to $\overline{REQ}$ In Low	50		ns
t_{ALAH}	$\overline{ACK}$ Out Low to $\overline{ACK}$ Out High	Tcyc-10		ns
t_{AHAL}	$\overline{ACK}$ Out High to $\overline{ACK}$ Out Low	Tcyc-25		ns
t_{AHPC}	$\overline{ACK}$ Out High to Phase Change In	0		ns

PRELIMINARY

Parameters t_{SHPC}, t_{IHDO}, and t_{PCRL} are also applicable and are identical to those in Send Synchronous Information Transfer In (Acting as a Target), bottom of page 40.



NOTE: Phase = signals that define the bus phase C/D, MSG

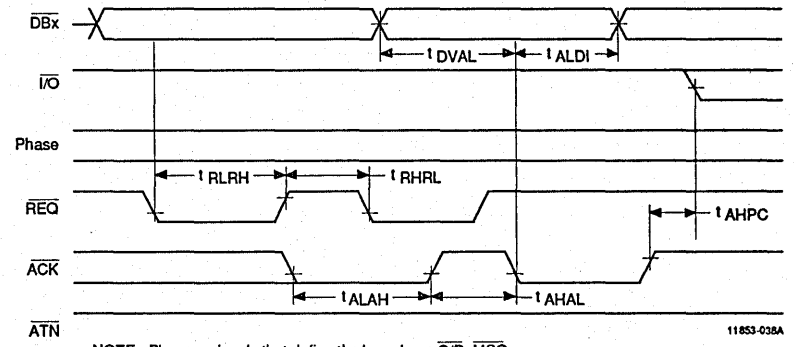
11853-037A

RECEIVE SYNCHRONOUS INFORMATION TRANSFER OUT (ACTING AS A TARGET)

Symbol	Characteristic	Min	Max	Units
t_{DVAL}	Data Valid In to $\overline{ACK}$ In Low	0		ns
t_{ALDI}	$\overline{ACK}$ In Low to Data Invalid	45		ns
t_{RLRH}	$\overline{REQ}$ Out Low to $\overline{REQ}$ Out High	Tcyc-10		ns
t_{RHRL}	$\overline{REQ}$ Out High to $\overline{REQ}$ Out Low	Tcyc-25		ns
t_{ALAH}	$\overline{ACK}$ In Low to $\overline{ACK}$ In High	50		ns
t_{AHAL}	$\overline{ACK}$ In High to $\overline{ACK}$ In Low	50		ns
t_{AHPC}	$\overline{ACK}$ In High to Phase Change Out	0		ns

PRELIMINARY

Parameters t_{shpc}, t_{ihdt}, and t_{pcrl} are also applicable and are identical to those in Send Synchronous Information Transfer Out (Acting as an Initiator), top of this page.



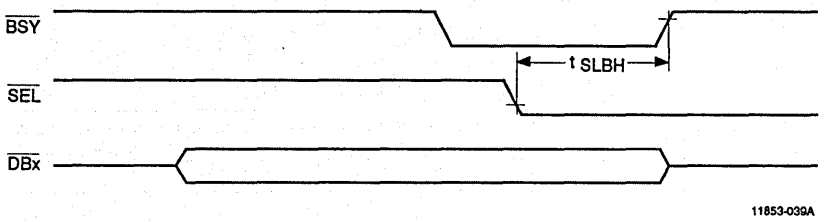
NOTE: Phase = signals that define the bus phase C/D, MSG

11853-038A

ARBITRATION TO BUS FREE

Symbol	Characteristic	Min	Max	Units
t_{SLBH}	$\overline{SEL}$ In Low to $\overline{BSY}$ High, Data TRI-STATE		8*Tcyc+75	ns

PRELIMINARY

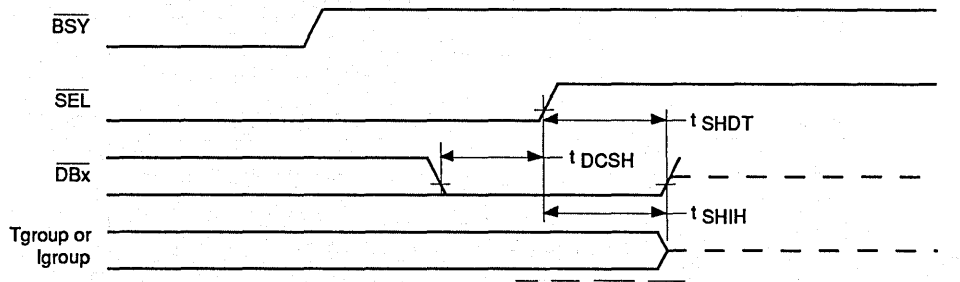


11853-039A

SELECTION (AS AN INITIATOR) OR RESELECTION (AS A TARGET) TO BUS FREE (SELECTION TIMEOUT)

Symbol	Characteristic	Min	Max	Units
t_{TADC}	Timeout or Abort to Data Bus Cleared	0		ns
t_{DCSH}	Data Bus Cleared to $\overline{SEL}$ Out High	200		µs
t_{SHDT}	$\overline{SEL}$ Out High to Data Bus TRISTATE		800	ns
t_{SHIH}	$\overline{SEL}$ Out High to CNTL TRISTATE		800	ns

PRELIMINARY



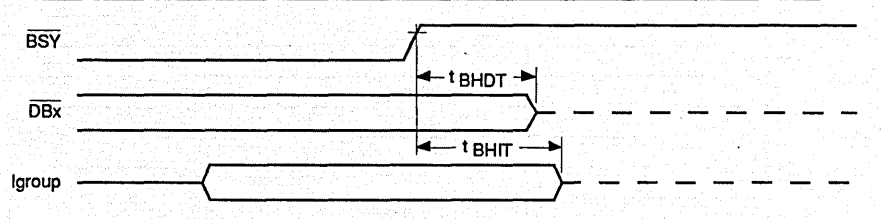
NOTE: Tgroup (signals driven by a Target) = $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, $\overline{REQ}$
Igroup (signals driven by an Initiator) = $\overline{ATN}$, $\overline{ACK}$

11853-040A

CONNECTED-AS-AN-INITIATOR TO BUS FREE

Symbol	Characteristic	Min	Max	Units
t_{BHDT}	$\overline{BSY}$ In High to Data Bus TRISTATE		8*Tcyc +75ns	ns
t_{BHIT}	$\overline{BSY}$ In High to Igroup TRISTATE		8*Tcyc +75ns	ns

PRELIMINARY

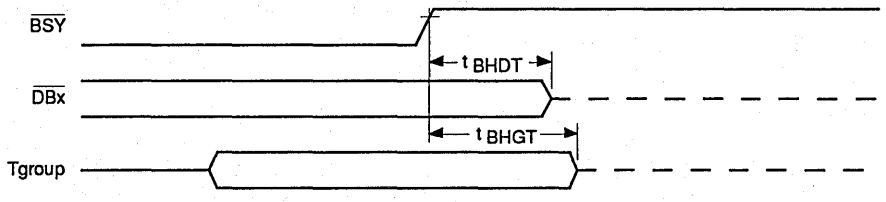


11853-041A

CONNECTED-AS-A-TARGET TO BUS FREE

Symbol	Characteristic	Min	Max	Units
t_{BHDT}	$\overline{BSY}$ Out High to Data Bus TRISTATE		8*Tcyc +75ns	ns
t_{BHGT}	$\overline{BSY}$ Out High to Tgroup TRISTATE		8*Tcyc +75ns	ns

PRELIMINARY



11853-042A

Preliminary

Am95C94

Advanced Burst Error Processor

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Implements Reed-Solomon code for the proposed X3B11 and ISO Continuous Composite Servo Standard**
- **Supports:**
  - —5-way and 10-way interleaving
  - —512-byte and 1024-byte sectors
  - —Alternative sector sizes up to 2550 bytes
  - —Layered CRC (optional)
- **Performs fast correction vector computation, entirely in hardware; there is no need for external software to reduce syndromes to vectors**
- **Parallel operation on two contiguous sectors; checkbytes for the second sector accumulate while correction vectors for first sector are computed. This allows "on-the-fly" correction**
- **Data and EDAC interface is byte-wide parallel; operates at speeds up to 10 Mb per second**
- **Microprocessor-compatible 8-bit data bus with optional parity**
- **Correction vector generation performance with a 20-MHz system clock is such that up to 80 correction vectors can be generated in under 400 μs for a 1024 byte sector**

GENERAL DESCRIPTION

The Am95C94 Advanced Burst Error Processor (ABEP) performs error detection and correction in optical disk systems. The ABEP implements the proposed ANSI X3B11 Standard Continuous Composite Servo Reed-Solomon Error Detection and Correction (EDAC) algorithm. It interfaces directly with AMD's disk data controller for optical systems, the Am95C96 Optical Disk Data Controller (ODC).

The ABEP simultaneously performs syndrome generation for the sector being read, and the correction vector generation for the previously read sector. This allows "on-the-fly" correction of data transferred from the disk at a high rate. This feature is extremely critical for optical disk systems, where read errors can be very frequent.

The ABEP works closely with the Am95C96 disk data controller. The ABEP resides on the buffer interface bus alongside the buffer memory. It functions as a slave peripheral on this bus, supporting high-speed disk data transfer operations. Parity can be generated and checked on this 8-bit bus.

During a disk write operation, the data controller transfers a sector of data bytes from the buffer memory to the disk and simultaneously to the ABEP. The ABEP generates checkbytes for this sector, and the data controller writes these checkbytes to the disk after the sector data.

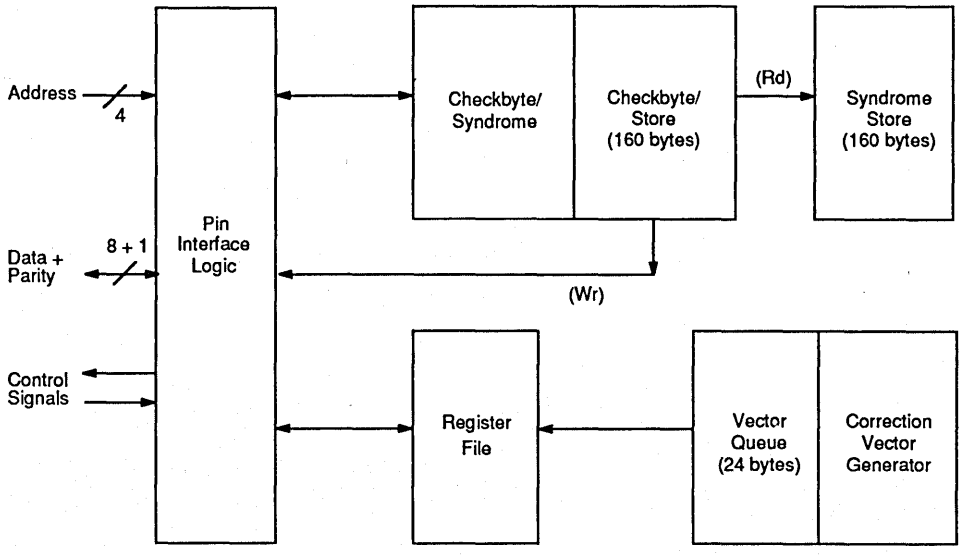
During a disk read operation, the data controller transfers a sector of data bytes from the disk to the buffer memory and simultaneously to the ABEP. Using the checkbytes read from the disk, the ABEP determines whether there are any errors, and, if so, how to correct them. If correction is required, correction vector generation takes place while the data and checkbytes of the next sector are being read from the disk and written to the ABEP. The disk data controller performs any necessary corrections to data during disk sector transfers by using the correction vectors computed by the ABEP. This ensures that errors can be corrected "on-the-fly," allowing high-speed, zero-sector-interleave read operations.

Unlike earlier error-processing chips, the Am95C94 ABEP may be treated as a black-box error processor; no software assistance is required for error checking or correction vector generation.

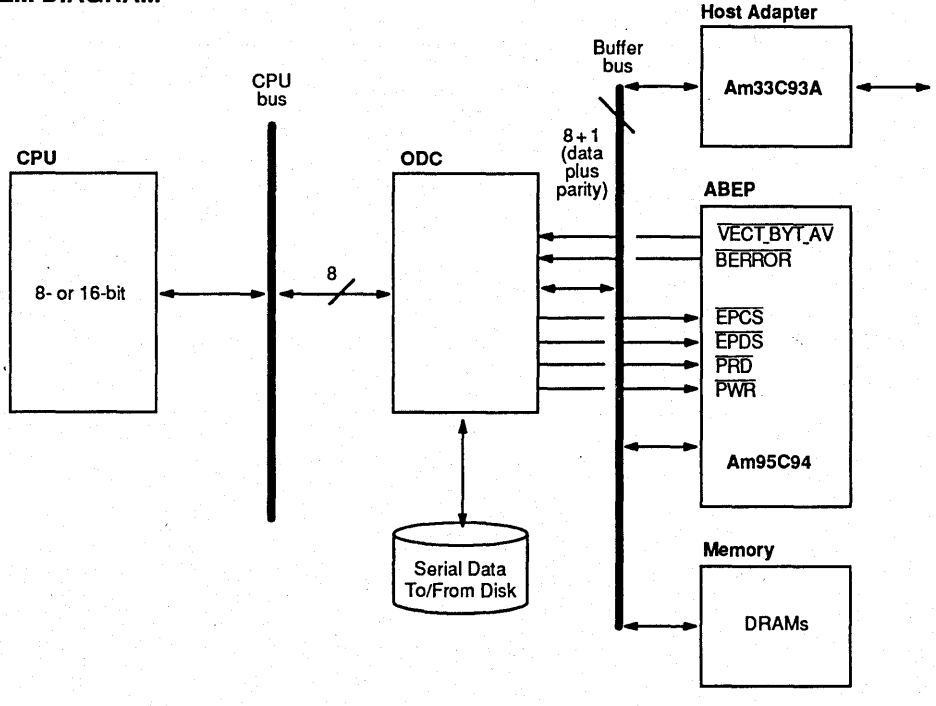
To order a full version of this data sheet (order #10934A), please contact your local sales office.

Publication #13079 Rev. A Amendment /0
Issue Date: October 1989

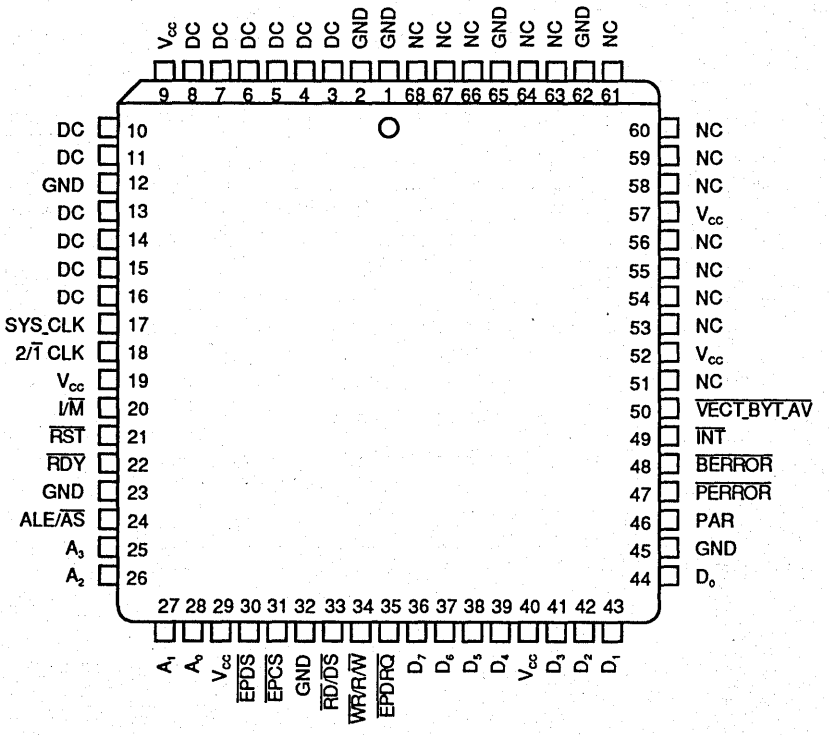
BLOCK DIAGRAM



SYSTEM DIAGRAM

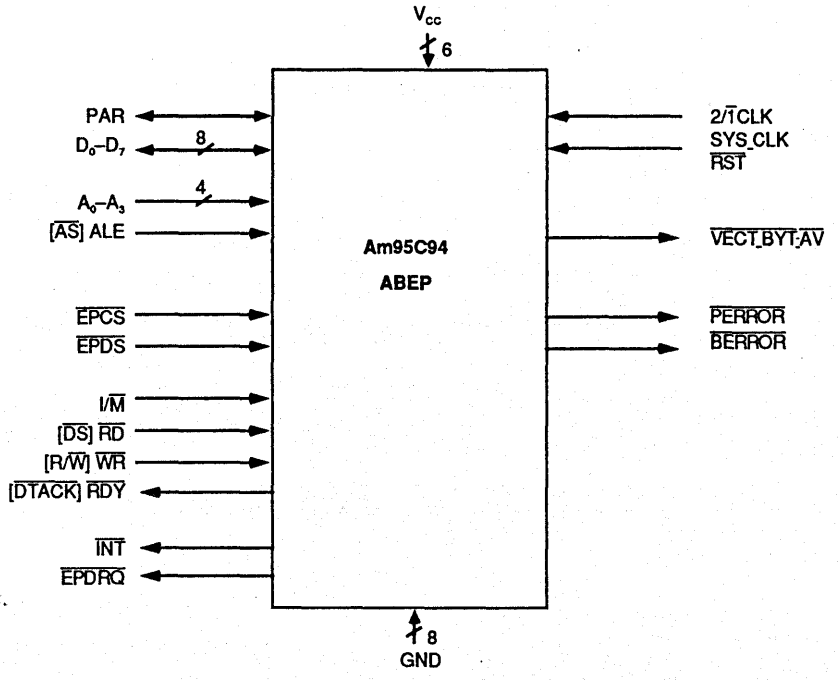


CONNECTION DIAGRAM



Note: NC means No Connection to Die,
DC means Do Not Connect These Pins.

LOGIC SYMBOL

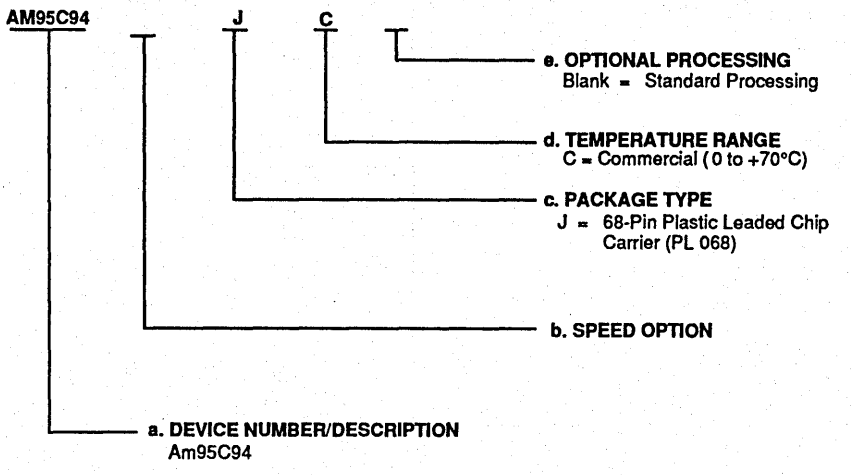


4

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C94	JC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Preliminary

Am95C95

Magnetic Disk Controller

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Handles ESMD/ESDI and ST-506 Magnetic Disk Data Format
- User-programmable disk formats
- On-chip Reed-Solomon Error Detection and Correction
- Minimal Rotational Latency Read/Write Operations
- 1:1 Sector Interleave Read/Write Operations
- 20-MHz System Clock
- Up to 32 MHz NRZ Serial Disk Data
- High-Level Command Set
- Supports Intelligent SCSI-type Interfaces
- On-chip Buffer Management
- Direct support of DRAM Buffer (up to 4 Mb)

GENERAL DESCRIPTION

The Magnetic Disk Data Controller (MDC) is a highly programmable magnetic disk data format controller designed for use in both standard and custom applications. The device has been optimized for applications utilizing an embedded SCSI disk data manager, but it easily supports other intelligent proprietary interfaces as well.

As an embedded controller, the MDC handles the timing of all disk read/write control signals, the transfer of NRZ data between the disk and buffer RAM, and the transfer of disk data between a parallel host interface and buffer RAM. The MDC performs all buffer management and arbitration associated with this transfer of data while maintaining data integrity through on-chip error detection and correction logic. In addition, the MDC can perform advanced data mapping operations during data transfer, simplifying the task of cache management. The MDC can handle up to 4 Mb of data buffer Dynamic RAM directly, including address multiplexing and refresh, with no external logic.

The MDC acts as an intelligent peripheral to a microprocessor or microcontroller. In an embedded SCSI disk controller application, the control CPU (referred to in this document as simply the *CPU*), interprets all SCSI commands and controls disk head positioning and head selection, while the MDC handles all data transfers. The CPU sets up and initiates MDC operation by modifying as few as six MDC registers. MDC registers are directly addressable, and can be read or written directly with either an iAPX multiplexed address/data bus, or a Motorola 68xxx-type non-multiplexed bus interface.

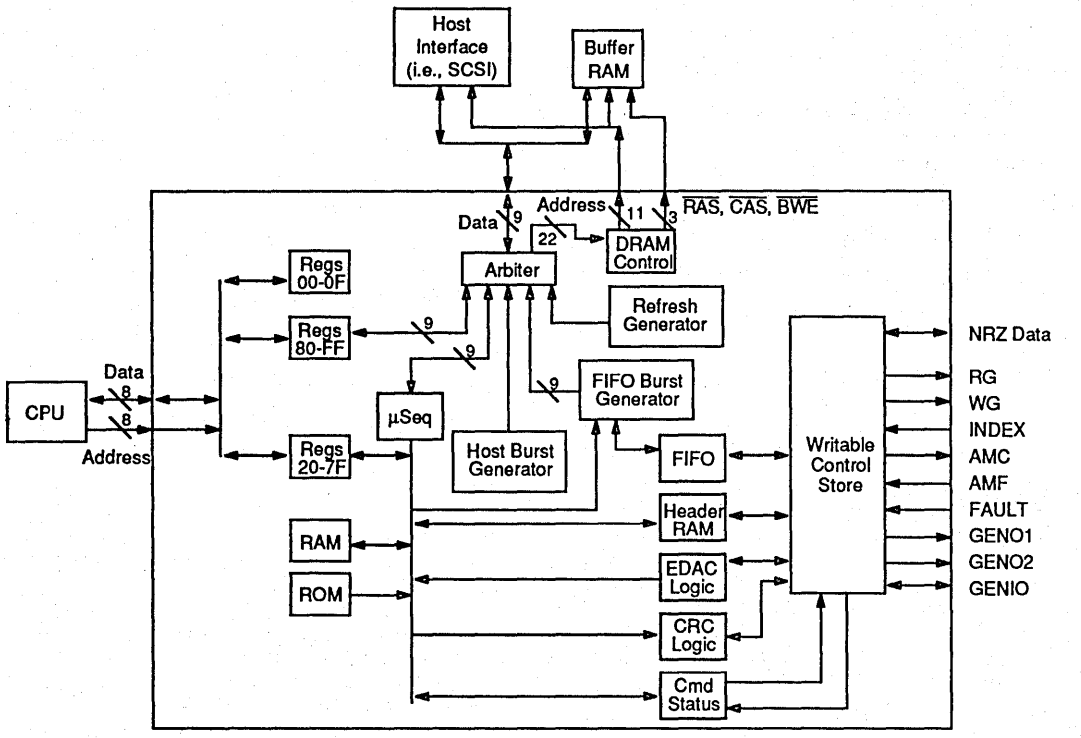
The on-chip Reed-Solomon Error Detection and Correction (EDAC) logic enables the MDC to perform on-the-fly error correction during disk read operations while maintaining a zero sector interleave on the disk. The MDC performs such error correction without CPU intervention. The Reed-Solomon logic supports two, three or five EDAC interleaves within a sector data field and either single- or double-burst correction capability.

4

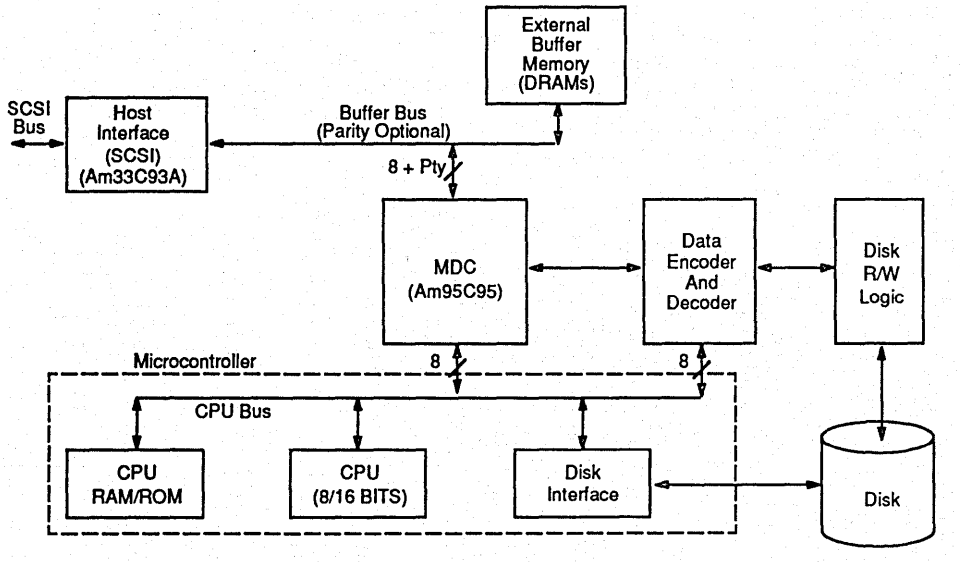
To order a full version of this data sheet (order #10935A), please contact your local sales office.

Publication # 13080 Rev. A Amendment /0
Issue Date: October 1989

INTERNAL BLOCK DIAGRAM



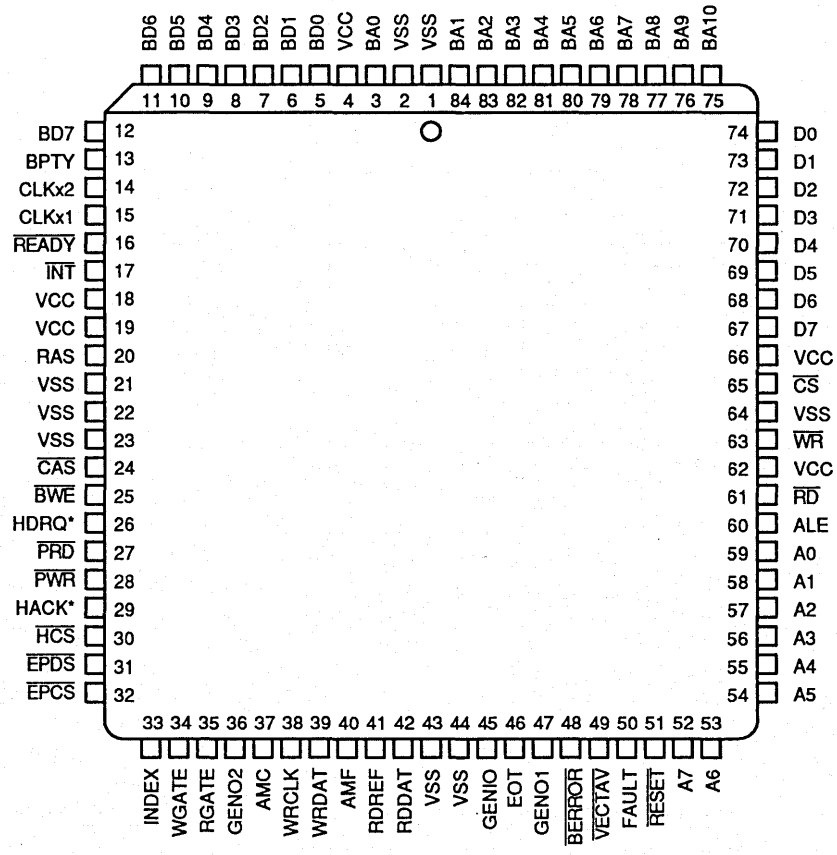
SYSTEM APPLICATION



RELATED AMD PRODUCTS

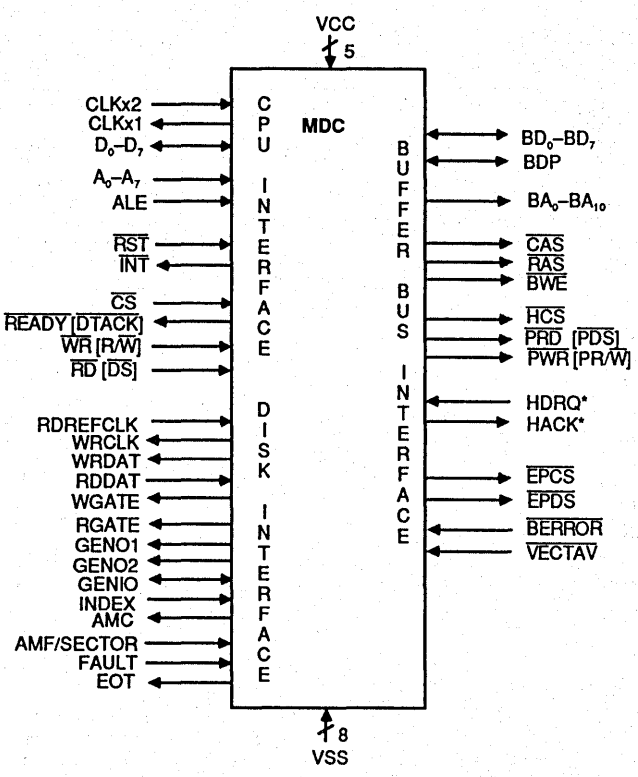
Part Number	Description	Part Number	Description
Am33C93A	SCSI Controller	Am80188	Microprocessor
Am53C80	SCSI Controller	Am95C94	Advanced Burst Error Processor
Am805x	Microcontroller		
Am80C521	CMOS Microcontroller	Am95C96	Optical Disk Controller

CONNECTION DIAGRAM



*These pins have programmable polarity.

LOGIC SYMBOL

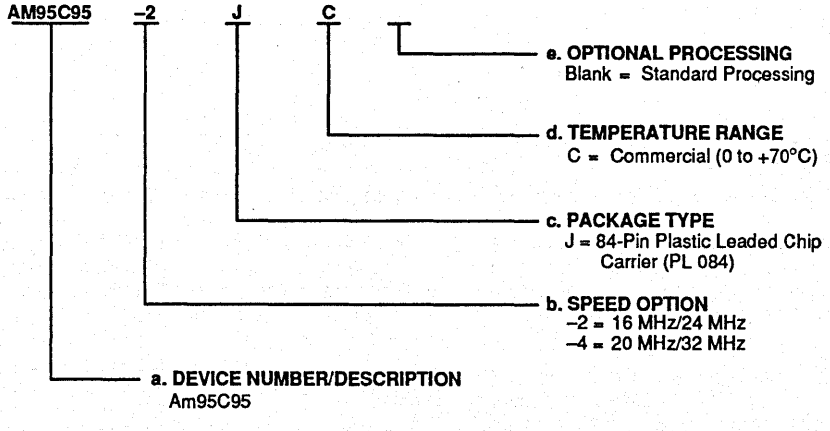


*These pins have programmable polarity.

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C95–2JC	
AM95C95–4JC	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Preliminary

Am95C96

Optical Disk Controller

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Supports the proposed X3B11 and ISO Continuous Composite Servo Standard
- User-programmable disk formats
- Minimal Rotational Latency Read/Write Operations
- 1:1 Sector Interleave Read/Write Operations
- 20-MHz System Clock
- Up to 32-MHz NRZ Serial Disk Data
- High-Level Command Set
- Supports Intelligent SCSI-type Interfaces
- On-chip Buffer Management
- Direct support of DRAM Buffer (up to 4 Mb)

GENERAL DESCRIPTION

The Optical Disk Data Controller (ODC) is a highly programmable optical disk data format controller designed for use in both standard and custom applications. The device has been optimized for applications utilizing an embedded SCSI disk data manager, but it easily supports other intelligent proprietary interfaces as well.

As an embedded controller, the ODC handles the timing of all disk read/write control signals, the transfer of NRZ data between the disk and buffer RAM, and the transfer of disk data between a parallel host interface and buffer RAM. The ODC performs all buffer management and arbitration associated with this transfer of data while maintaining data integrity by using a dedicated interface to an external, high-performance Advanced Burst Error Processor (ABEP). In addition, the ODC can perform advanced data mapping operations during data transfer, simplifying the task of cache management. The ODC can handle up to 4 Mb of data buffer Dynamic RAM directly, including address multiplexing and refresh, with no external logic.

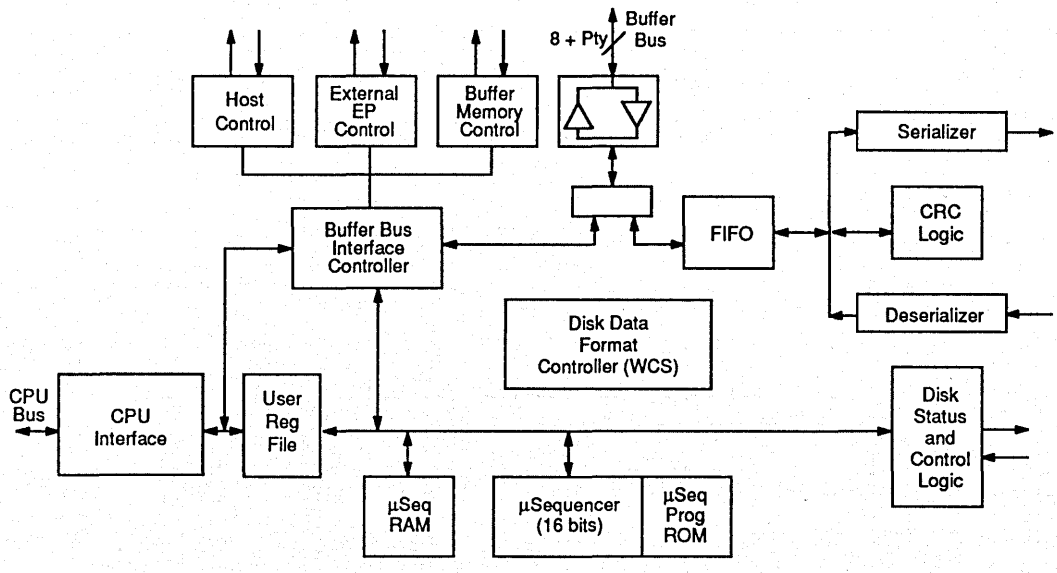
The ODC acts as an intelligent peripheral to a microprocessor or microcontroller. In an embedded SCSI disk controller application, the control CPU (referred to in this document as simply the *CPU*), interprets all SCSI commands and controls disk head positioning and head selection, while the ODC handles all data transfers. The CPU sets up and initiates ODC operation by modifying as few as six ODC registers. ODC registers are directly addressable, and can be read or written directly with either an iAPX-multiplexed address/data bus, or a Motorola 68xxx-type non-multiplexed bus interface.

A dedicated interface to an external error processor permits the ODC to cope with the high error rates associated with optical media. Through this interface, the ODC, in combination with AMD's Advanced Burst Error Processor (ABEP), can correct an error in as little as 10 µs. The ABEP supports 5- or 10-way interleaving with optional overlay CRC capability. The ABEP can correct up to 80 errors in a single sector in less than a millisecond. With this error-correction capability, the ODC and ABEP can perform on-the-fly error correction without missing a sector. This avoids the necessity of jumping back to continue a disk read operation following a read error when handling spiral formatted media.

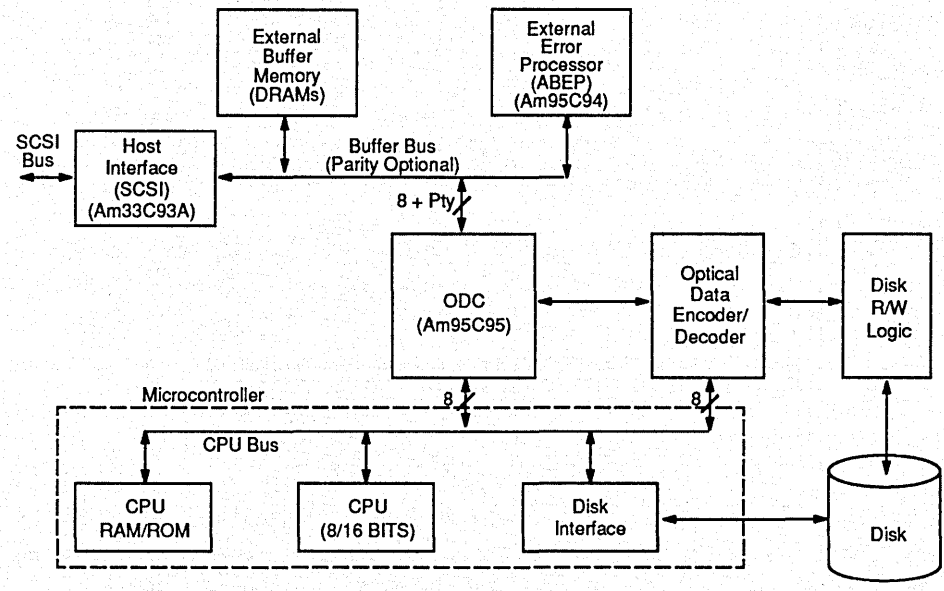
To order a full version of this data sheet (order #10936A), please contact your local sales office.

Publication #13081 Rev. A Amendment /0
Issue Date: October 1989

INTERNAL BLOCK DIAGRAM



SYSTEM APPLICATION

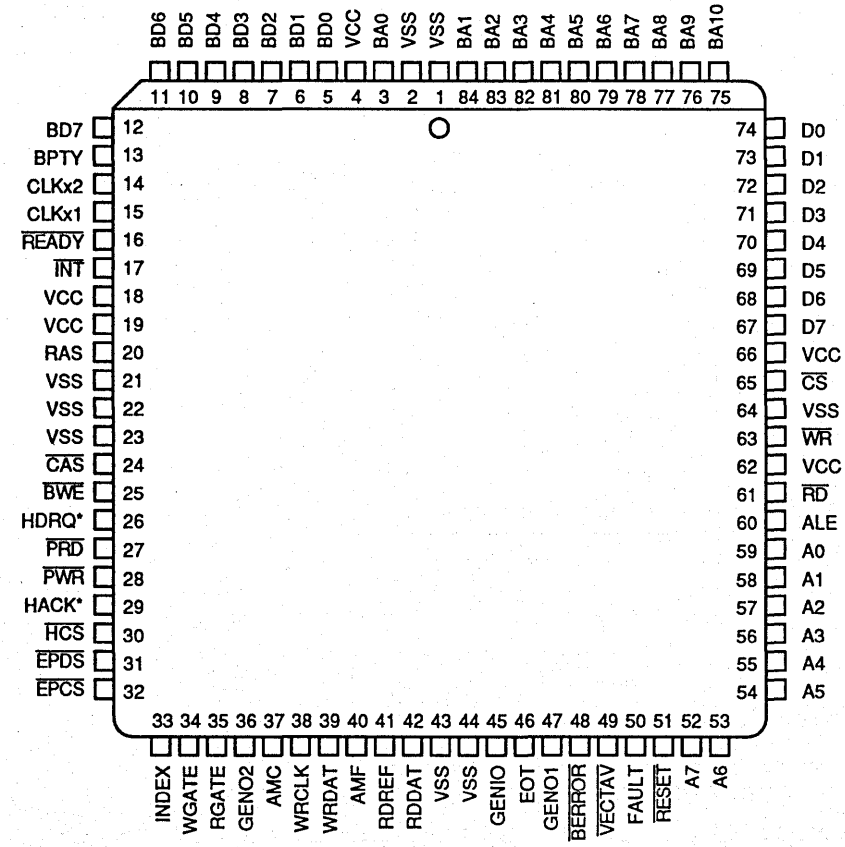


4

RELATED AMD PRODUCTS

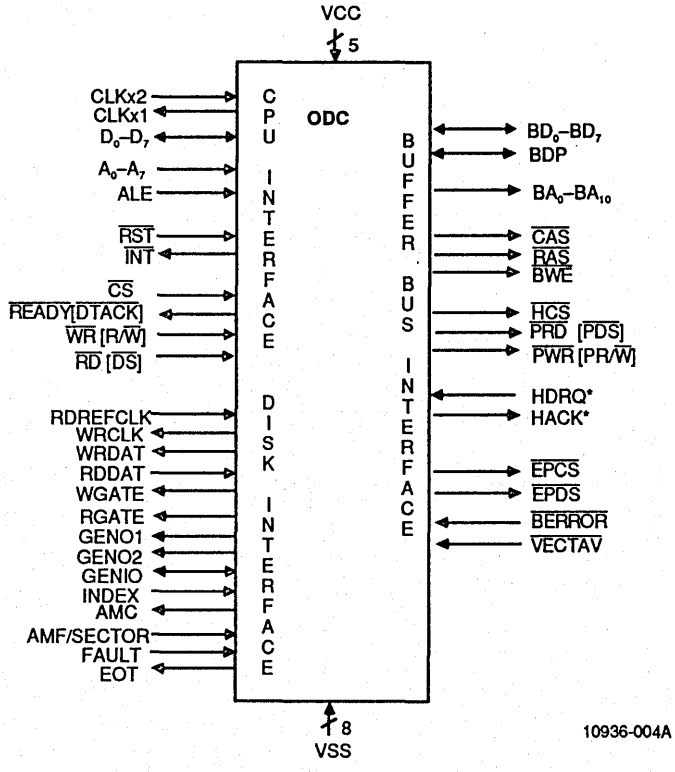
Part Number	Description	Part Number	Description
Am33C93A	SCSI Controller	Am80188	Microprocessor
Am53C80	SCSI Controller	Am95C94	Advanced Burst Error Processor
Am805x	Microcontroller		
Am80C521	CMOS Microcontroller	Am95C95	Magnetic Disk Controller

CONNECTION DIAGRAM



*These pins have programmable polarity.

LOGIC SYMBOL

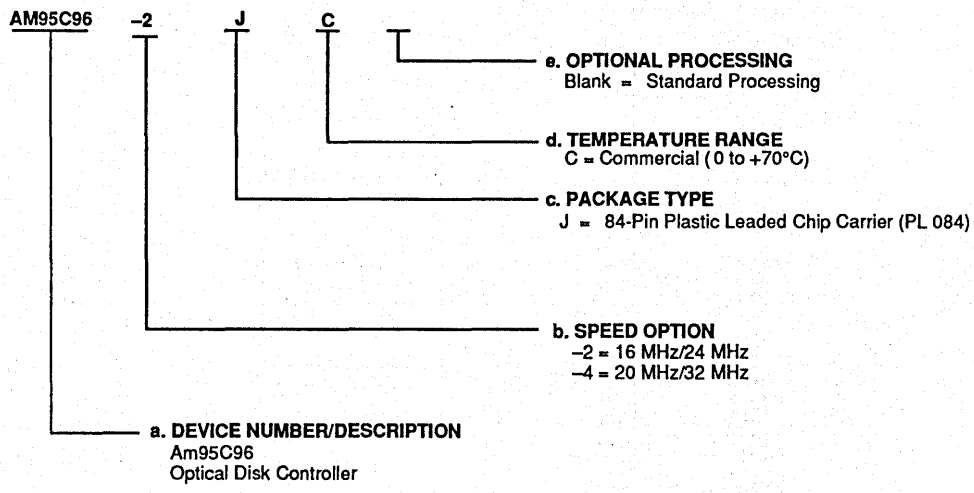


*These pins have programmable polarity.

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C96-2	JC
AM95C96-4	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Am9580A/Am9590

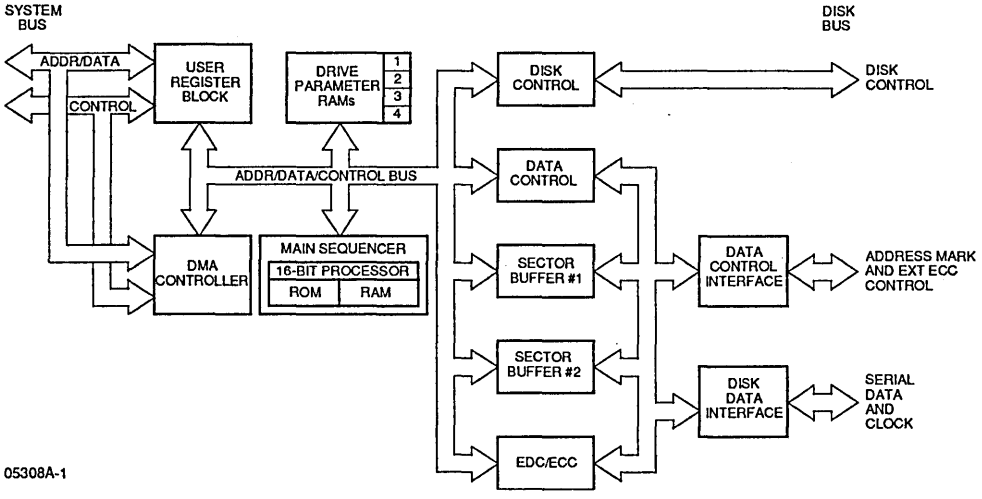
Hard Disk Controllers

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- **Am9580A supports ST506/412 and IBM double-density floppy formats**
- **Am9590 supports ESDI, ST506/412, and IBM double-density floppy formats**
- **Supports hard- and soft-sectored formats**
- **Controls up to four drives in any mix of hard and flexible formats**
- **Two on-chip 512-byte sector buffers support zero-sector interleaving**
- **Supports error checking algorithms including:**
  - CRC/CCITT
  - Single-Burst Reed-Solomon
  - Double-Burst Reed-Solomon
  - External ECC (user-definable Error Correcting Codes)
- **Linked-list command and data structures**
- **On-chip DMA controller supports 32-bit addressing and 8/16 bit data**
- **Am9590 supports data rates up to 15 Mbit/second**

BLOCK DIAGRAM



To receive a complete data sheet, contact your local sales office.

Publication #	Rev.	Amendment
09853	A	/0

Issue Date: March 1988

GENERAL DESCRIPTION

The Am9580A and the Am9590 are single-chip solutions to the problems encountered in designing data formatters and disk system controllers. A companion part is also offered, the Am9582 Disk Data Separator, which when combined with one of the above disk controllers provides all of the functions which until now have been found only on sophisticated board-level products.

Both of these highly integrated disk controllers are flexible enough to cope with the differing requirements of today's broad marketplace, while using the advanced technology and innovative features that tomorrow's market will demand.

These disk controllers support both rigid and flexible disk drives and their respective data formats. Four drives of any mix (hard and flexible) can be controlled with these devices, with individual drive characteristics easily user-programmable.

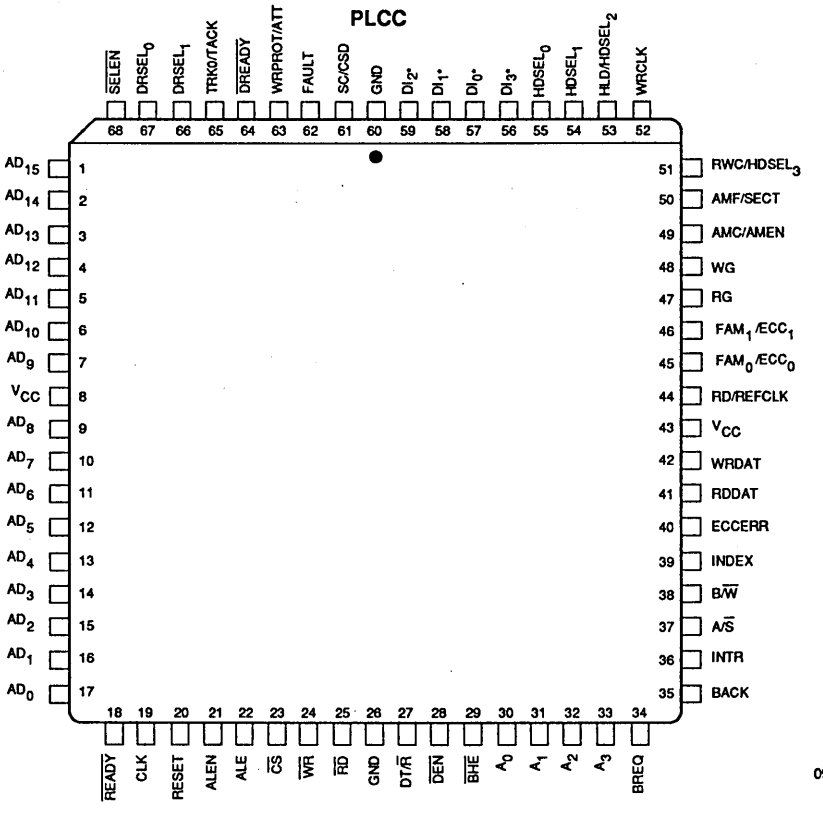
A sophisticated on-chip DMA controller fetches commands, writes status information, fetches data to be written on disk, and writes data that has been read from the disk. The DMA operation is programmable to adjust the bus occupancy, data bus width (8 or 16 bits), and wait state insertion. Two sector buffers allow zero-sector interleaving to access data on physically adjacent sectors, improving both file access time and system throughput. Sector sizes of 128, 256, and 512 bytes are programmable.

Both controllers ensure data integrity by selecting either an error-detecting code (CRC-CCITT), or one of two error-correcting codes (Single- or Double-Burst Reed-Solomon). Additionally, the Hard Disk Controller (HDC) provides handshake signals to control external Error-Correcting Codes (ECC) circuitry to implement any user-definable ECC algorithm.

The ESDI and ST506/412 interfaces are completely supported by the Am9590. Users interested only in the ST506/412 standard can use the Am9580A. Both of these controllers provide all of the required signals.

CONNECTION DIAGRAMS

Top View



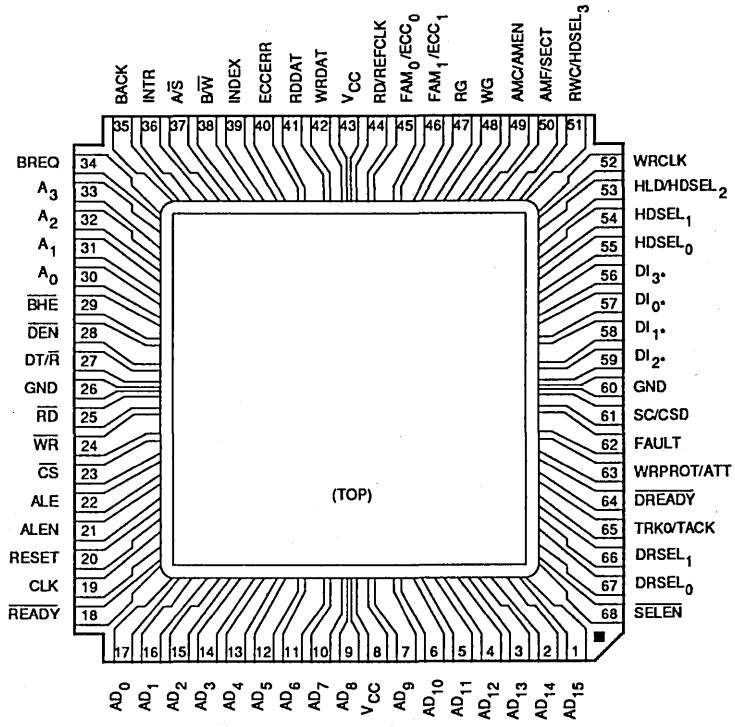
09853A-2A

*Refer to Pin Description section for options.

CONNECTION DIAGRAMS (Cont'd.)

Top View

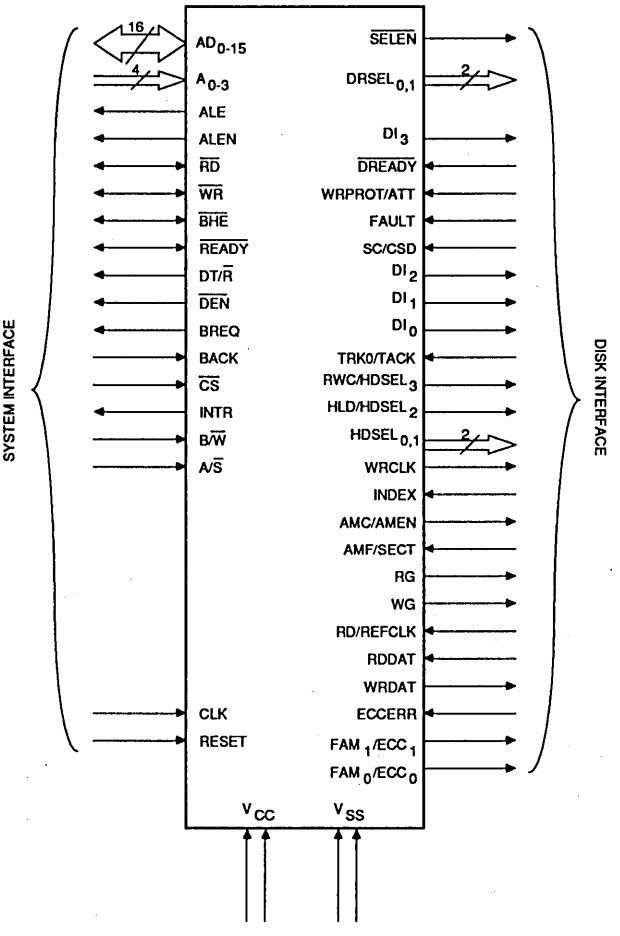
LCC



* Refer to Pin Description section for options.

09853A-1A

LOGIC SYMBOL



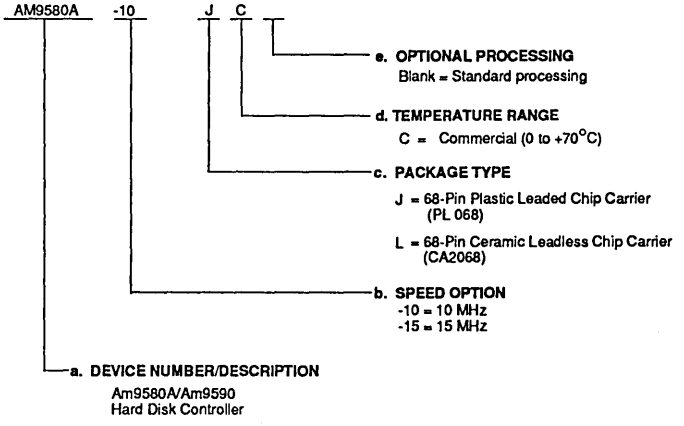
09853A-20

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing

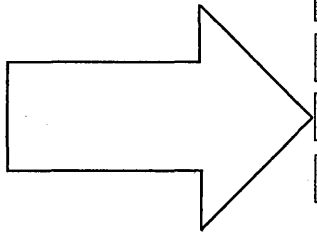


Valid Combinations	
AM9580A-10	JC, LC
Am9590-15	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Advanced
Micro
Devices



CHAPTER 5
DOCUMENT PROCESSING

Am7971A

Compression Expansion Processor
(CEP with image bit-boundary processing)
FINAL

DISTINCTIVE CHARACTERISTICS

- Image preserving compression and expansion of two-tone image using run-length (one-dimensional) coding and relative element address (two-dimensional) coding.
- Compatible with internationally accepted CCITT Group III and IV (Recommendations T.4 and T.6) image compression standards.
- Image bit-boundary operations.
- High performance of 1 to 12 MHz pixel rates with 3, 5, and 8MHz clock.
- CPU bus and optional local Document Store Bus with on-chip DMA. The CEP can address up to 16Mbytes on each bus.
- Handles four memory buffers: source and destination buffers for both the compressor and expander.
- Full duplex mode for simultaneous compressor and expander operations with each processor independently programmable.
- On-chip error detection to catch data corruptions and support for easy error recovery.
- 46 user programmable registers allow for very easy and highly flexible system implementation. Includes:
  - Programmable page width (up to 16K pels), frame width and top, left and right margins.
  - Optional Express mode during compression and Granularity mode during expansion for vertical resolution conversion.
  - Programmable K parameter.
  - Optional Wraparound mode.
  - Transparent mode.

GENERAL DESCRIPTION

The Am7971A Compression Expansion Processor (CEP) with Image Bit-Boundary Processing capacity is a high performance peripheral which compresses and expands two-tone bit mapped images or documents in accordance with internationally accepted CCITT standards. These fully image preserving algorithms reduce storage requirements and data transmission time for systems handling bit-mapped data.

The Am7971A is a functionally enhanced version of the Am7971 offering improved negative compression and error recovery performance. The Am7971A can replace the Am7971 in existing systems without board/system/timing alterations.

The Am7971A performs one-dimensional Modified Hoffman (MH) run-length coding as well as two-dimensional Modified READ (MR/MMR) relative coding as specified in CCITT Recommendations T.4 and T.6 for Group III and Group IV compatible equipments. The typical compression ratio for the eight CCITT test documents is 5:1 to 50:1.

The compressor and expander operate not only in full duplex mode but each processor can be independently programmed for one-dimensional encoding/decoding, two-dimensional encoding/decoding, or transparent data transfer.

Equipped with an on-chip error detection mechanism, the Am7971A detects data corruptions by checking for illegal codes, negative run-lengths and incorrect line lengths. Furthermore, its architecture allows for error recovery with minimal CPU intervention.

With 46 user programmable registers, standard Am8088-like microprocessor bus interface, dual bus architecture and on-chip DMA the Am7971A offers tremendous system flexibility and ease of implementation. After initialization the Am7971A will operate with minimal CPU overhead. Its status is available through polled registers and exception conditions may be signalled using an external interrupt.

Document page width is programmable up to 16K picture elements (pels). Programmable frame width enable windowing features and programmable top, left and right margins allow image boundaries to be left blank.

Optional express mode allows one line to be skipped after every 'nth' line to accelerate compression (n = 1 to 255). On the expansion side, the granularity option allows the processor to duplicate every mth line (m = 1 to 7).

In two-dimensional mode, the programmable K-parameter (k = 1 to 255 and infinity) defines the number of lines to be encoded in 2-D coding sequence before a 1-D line is inserted. For error free environments (Group 4) K = infinity allows for maximum compression.

The CEP can address up to 16 Mbytes of memory on each bus and two buffers (source and destination) on both the compressor and expander. Starting address, buffer length and current address for image and coded data are stored in internal registers independently for both the compressor and expander.

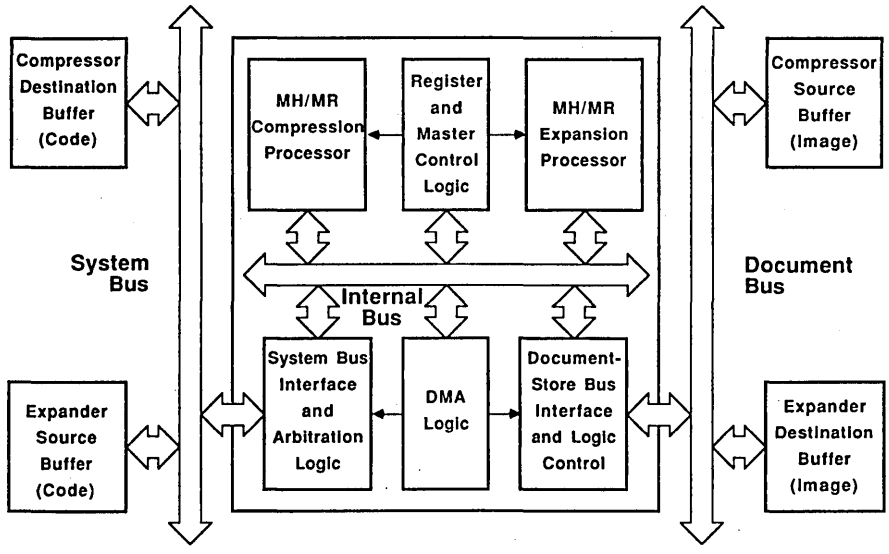


Figure 1. Am7971A Block Diagram

RELATED AMD PRODUCTS	
Part No.	**Description**
Am7971A EVAL	Am7971A Evaluation Board

CONNECTION DIAGRAMS
Top View

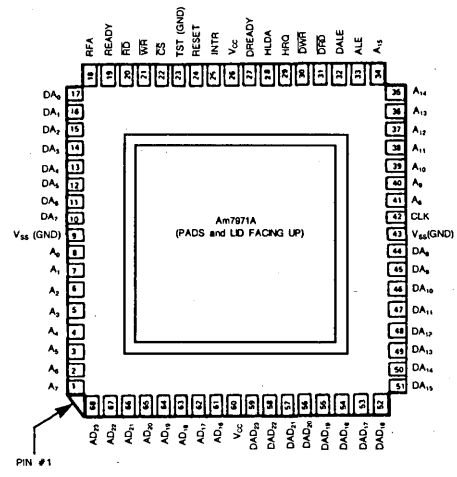


Figure 2. Am7971A Pinout for Leadless Chip Carrier (LCC)

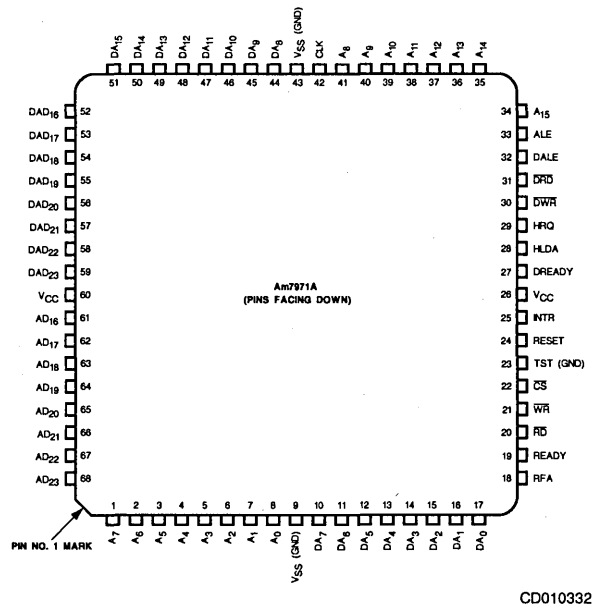


Figure 3. Am7971A Pinout for Plastic Leaded Chip Carrier (PLCC)

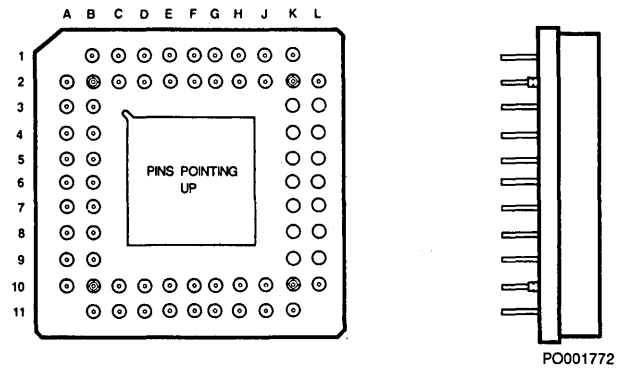


Figure 4. Am7971A (CEP) Pinout for a Pin Grid Array (PGA) Package

PIN DESIGNATIONS							
(SORTED BY PIN NAME)				(SORTED BY PIN NUMBER)			
PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NO.	PIN NAME	PIN NO.	PIN NAME
A_0	E-2	DA_7	F-2	A-2	AD_{23}	G-1	DA_6
A_1	E-1	DA_8	F-10	A-3	AD_{21}	G-2	DA_5
A_2	D-2	DA_9	E-11	A-4	AD_{19}	G-10	CLK
A_3	D-1	DA_{10}	E-10	A-5	AD_{17}	G-11	A_8
A_4	C-2	DA_{11}	D-11	A-6	V_{CC}	H-1	DA_4
A_5	C-1	DA_{12}	D-10	A-7	DAD_{22}	H-2	DA_3
A_6	B-2	DA_{13}	C-11	A-8	DAD_{20}	H-10	A_9
A_7	B-1	DA_{14}	C-10	A-9	DAD_{18}	H-11	A_{10}
A_8	G-11	DA_{15}	B-11	A-10	DAD_{16}	J-1	DA_2
A_9	H-10	DAD_{16}	A-10	B-1	A_7	J-2	DA_1
A_{10}	H-11	DAD_{17}	B-10	B-2	A_6	J-10	A_{11}
A_{11}	J-10	DAD_{18}	A-9	B-3	AD_{22}	J-11	A_{12}
A_{12}	J-11	DAD_{19}	B-9	B-4	AD_{20}	K-1	DA_0
A_{13}	K-10	DAD_{20}	A-8	B-5	AD_{18}	K-2	READY
A_{14}	K-11	DAD_{21}	B-8	B-6	AD_{16}	K-3	$\overline{WR}$
A_{15}	L-10	DAD_{22}	A-7	B-7	DAD_{23}	K-4	TST (GND)
AD_{16}	B-6	DAD_{23}	B-7	B-8	DAD_{21}	K-5	INTR
AD_{17}	A-5	$\overline{DALE}$	L-9	B-9	DAD_{19}	K-6	DREADY
AD_{18}	B-5	$\overline{DRD}$	K-8	B-10	DAD_{17}	K-7	HRQ
AD_{19}	A-4	DREADY	K-6	B-11	DA_{15}	K-8	$\overline{DRD}$
AD_{20}	B-4	$\overline{DWR}$	L-8	C-1	A_5	K-9	ALE
AD_{21}	A-3	HLDA	L-7	C-2	A_4	K-10	A_{13}
AD_{22}	B-3	HRQ	K-7	C-10	DA_{14}	K-11	A_{14}
AD_{23}	A-2	INTR	K-5	C-11	DA_{13}	L-2	RFA
ALE	K-9	$\overline{RD}$	L-3	D-1	A_3	L-3	$\overline{RD}$
CLK	G-10	READY	K-2	D-2	A_2	L-4	$\overline{CS}$
$\overline{CS}$	L-4	RESET	L-5	D-10	DA_{12}	L-5	RESET
DA_0	K-1	RFA	L-2	D-11	DA_{11}	L-6	V_{CC}
DA_1	J-2	TST (GND)	K-4	E-1	A_1	L-7	HLDA
DA_2	J-1	V_{CC}	A-6	E-2	A_0	L-8	$\overline{DWR}$
DA_3	H-2	V_{CC}	L-6	E-10	DA_{10}	L-9	DALE
DA_4	H-1	V_{SS} (GND)	F-1	E-11	DA_9	L-10	A_{15}
DA_5	G-2	V_{SS} (GND)	F-11	F-1	V_{SS}		
DA_6	G-1	$\overline{WR}$	K-3	F-2	DA_7		
				F-10	DA_8		
				F-11	V_{SS}		

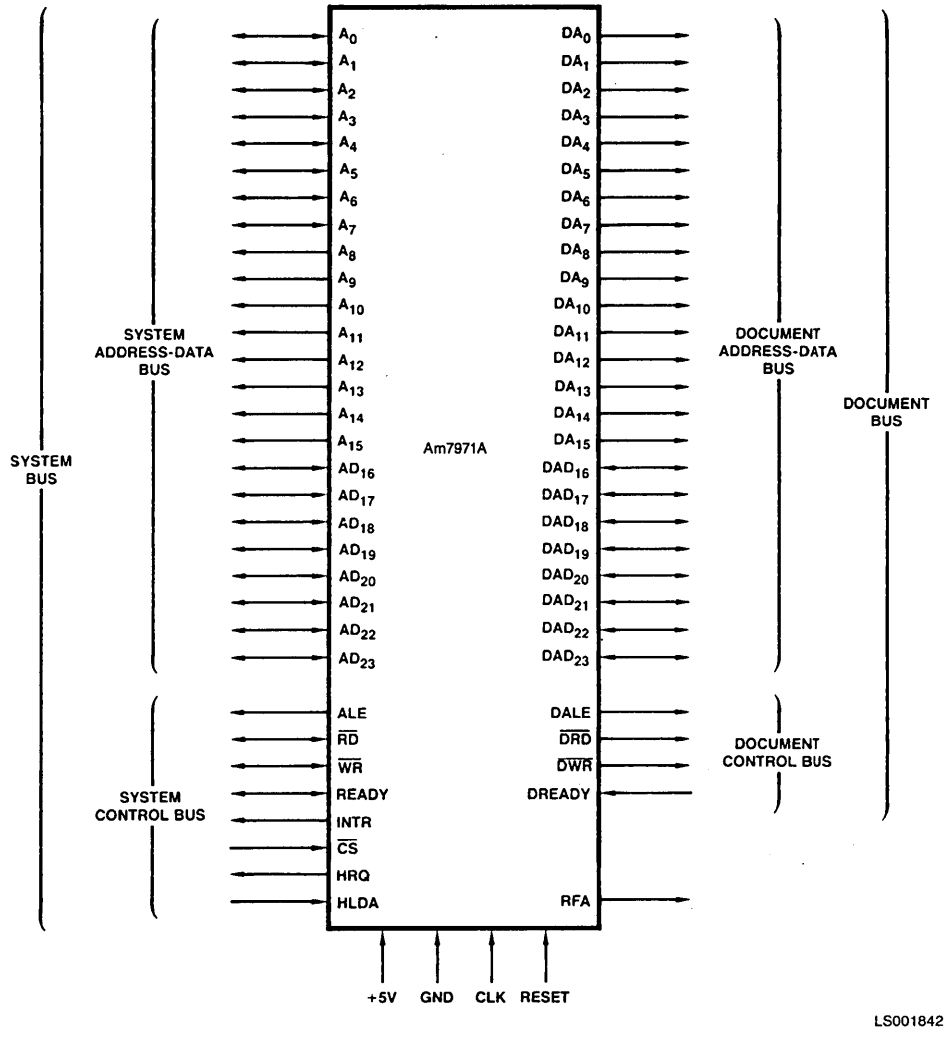
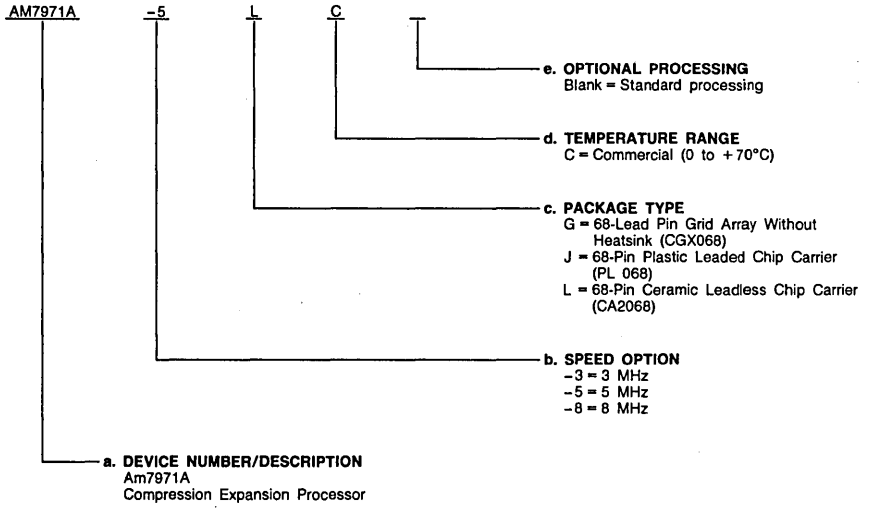


Figure 5. Am7971A Logic Symbol

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM7971A-3	JC
AM7971A-5	GC, JC, LC
AM7971A-8	GC, LC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

CLK Clock (Input)
The Clock signal controls most of the CEP's internal operations and determines the rates of its data transfers. The Clock input accepts a TTL voltage level. The input signals $\overline{CS}$, HLDA, $\overline{RD}$, and $\overline{WR}$, can make transitions independent of the CEP clock (asynchronous operation).

RESET (Input)
RESET is an asynchronous, active-High input which initializes the CEP to an idle state. This input must be driven High for at least four clock cycles. The hardware reset initializes the internal state machine, the DMA, and the interrupt controller. Then it starts a software reset on the expander and compressor.

V_{CC}
Power supply input. Connect to +5 V.

V_{SS}
Ground (GND) input. Connect to 0 V.

SYSTEM BUS CONTROL SIGNALS

$A_0 - A_7$ Lower Address Outputs/Internal Register Address Inputs (Bidirectional)

$A_8 - A_{15}$ Lower Address (Three-state Outputs)
When the CEP is not in control of the system bus (HRQ and HLDA Low = Slave Mode), and the $\overline{CS}$ input is Low, $A_1 - A_7$ are used as input address lines to access the CEP's internal registers. During this time, the address lines A_0, $A_8 - A_{15}$ are ignored by the CEP. The input addresses on $A_1 - A_7$ are latched by the rising edge of $\overline{RD}$ or $\overline{WR}$. In the Bus Master mode (HRQ and HLDA High) $A_0 - A_{15}$ are three-state non multiplexed address outputs used for the system side memory transactions. The presence of valid address on $A_0 - A_{15}$ is defined by the falling edge of ALE. These lines are enabled 2 clock cycles after HREQ and HLDA = High. After the High-to-Low transition of HRQ, the $A_0 - A_{15}$ lines will float.

$AD_{16} - AD_{23}$ Address-Data (Input/Output, Bus Three-state)
The Address-Data Bus is a time-multiplexed (in Master Mode only), bidirectional, active-High, 3-state bus used for all system bus I/O and memory transactions. The presence of a valid address during Bus Master operations is defined by the falling edge of ALE and valid data is defined by the $\overline{WR}$ and $\overline{RD}$ signals; otherwise these lines are floating. While the CEP $\overline{RD}$ output is Low, $AD_{16} - AD_{23}$ must contain valid input data from the system while the READY input is High. When the CEP $\overline{WR}$ output is asserted Low, $AD_{16} - AD_{23}$ has valid CEP output data. When the CEP is acting as a Bus Slave (HRQ and HLDA Low) and the $\overline{CS}$ input is driven Low, $AD_{16} - AD_{23}$ are used strictly as data lines $D_0 - D_7$. They behave as input data lines when $\overline{WR}$ is asserted Low and as output data lines when $\overline{RD}$ is asserted Low. At all other times they are floating.

ALE Address Latch Enable (Output)
This active-High signal is provided by the CEP to latch the address signals $AD_{16} - AD_{23}$ into an address latch. This pin is never floated. ALE is asserted during address time when the CEP is Bus Master; otherwise it is Low. Address is defined as valid prior to the High-to-Low (trailing) transition of ALE.

$\overline{CS}$ Chip Select (Input, Active Low)
$\overline{CS}$ is an asynchronous, active-Low input. A CPU or other external device uses $\overline{CS}$ to activate the CEP for reading from or writing to its internal registers. Once asserted, this input can remain Low until all register accesses have been completed (block transfer). Once $\overline{CS}$ goes High, it may not go Low again for at least 100 ns. $\overline{CS}$ is ignored when the CEP is in control of the system bus.

HLDA Hold Acknowledge (Input)
HLDA is an asynchronous, active-High input indicating that the CPU has relinquished the bus and that no higher priority device has assumed bus control. HLDA is internally synchronized by the CEP. The HLDA input to the CEP can be set Low prior to HRQ (Preemption). This forces the CEP to release the bus within a maximum time of 5 clock periods (assuming READY is High and no wait states).

HRQ Hold Request (Output)
Hold Request is an active-High signal used by the CEP to obtain control of the bus from the system CPU.If the HLDA input is High after the HRQ output goes High, HRQ will remain High until the CEP has completed one memory transaction. The HLDA input may go Low prior to HRQ going Low. The HRQ signal remains Low for a minimum of 2 clocks to allow the bus master to arbitrate for the bus. If HLDA is not asserted, HRQ can be forced Low only by a hardware reset.

INTR Interrupt Request (Output)
Interrupt Request is an active-High output used to interrupt the CPU. It is driven High whenever an exception or terminating condition exists in either the Compressor (if the Compressor Interrupt Enable bit is set) or Expander (if the Expander Interrupt Enable bit is set). The INTR line is reset to Low when the CPU reads the CEP Master Status Register or when the CEP is hardware reset.

$\overline{RD}$ Read (Input/Output, Active LOW, Three-state)
$\overline{RD}$ is a bidirectional, active-Low, 3-state signal. A Low indicates that the $AD_{16} - AD_{23}$ bus is being used for a Read Data Transfer. When the CEP is not in control of the system bus and the external system is transferring information from the CEP, $\overline{RD}$ is an asynchronous timing input used by the CEP to move data between registers and the $AD_{16} - AD_{23}$ bus. $\overline{RD}$ is an output when the CEP is Bus Master (HRQ and HLDA are both High). The CEP asserts $\overline{RD}$ when data from system memory is required.

READY (Input/Output, Three-state)
READY is a synchronous, active-High, 3-state, bidirectional signal. READY is used as an input signal when the CEP is Bus Master. In Master Mode, the CEP samples the READY line with the rising edge of T2 before $\overline{RD}$ or $\overline{WR}$ are asserted by the CEP. If READY is Low during this time, wait cycles are inserted until READY is returned High. This input must be synchronized to the CEP clock. READY is used as an output signal when the CEP is Bus Slave. After $\overline{CS}$ has been asserted by the CPU, READY is kept Low by the CEP until it is ready to respond.

$\overline{WR}$ Write (Input/Output, Active Low, Three-state)
$\overline{WR}$ is a bidirectional, active-Low, three-state signal. A Low indicates that the $AD_{16} - AD_{23}$ bus is being used for a Write Data Transfer. When the CEP is not in control of the system bus and the external system is transferring information to the CEP, $\overline{WR}$ is an asynchronous timing input used by the CEP to move data from the $AD_{16} - AD_{23}$ bus into its internal registers. $\overline{WR}$ is an output when the CEP is Bus Master (HRQ and HLDA are both High). The CEP asserts $\overline{WR}$ when data is to be written into Main Memory.

DOCUMENT BUS CONTROL SIGNALS

$DA_0 - DA_{15}$ Document Store Lower Address Bus (Output, Three-state)

The Document Store Lower Address Bus is a non-multiplexed, active-High, Three-state bus used in addressing all local document memory transactions. When the CEP is in control of the Document Store Bus, the presence of a valid address on $DA_0 - DA_{15}$ is defined by the falling edge of DALE. When the CEP does not use the document bus for a memory transaction, these lines are floating.

$DAD_{16} - DAD_{23}$ Document Store Upper Address-Data Bus (Input/Output, Three-state)

The Document Store Upper Address-Data Bus is a time-multiplexed, bidirectional, active-High, Three-state bus used for all local document memory transactions. The presence of a valid address during a document bus memory transaction is defined by the falling edge of DALE and the valid data is defined by the $\overline{DWR}$ and $\overline{DRD}$ signals; otherwise these lines are floating.

DALE Document Store ALE (Output, Three-state)

This active-High output signal is provided by the CEP to indicate valid address signals $DAD_{16} - DAD_{23}$. When the CEP does not need the document interface for a data transfer, this pin is floating.

$\overline{DRD}$ Document Store Read (Output, Active Low, Three-state)

$\overline{DRD}$ is an active-Low, Three-state output signal. A Low on this signal indicates that the $DAD_{16} - DAD_{23}$ bus is being used for a Read Data Transfer. When the CEP does not need this interface for a data transfer, this pin is floating.

DREADY Ready (Input)

DREADY is a synchronous, active-High input. DREADY is used as an input signal when the CEP is Bus Master. The CEP samples the DREADY line with the rising edge of T2 before $\overline{DRD}$ or $\overline{DWR}$ is asserted by the CEP. If DREADY is Low during this time, wait cycles are inserted until DREADY is returned High. This input must be synchronized to the CEP clock.

$\overline{DWR}$ Document Store Write (Output, Active Low, Three-state)

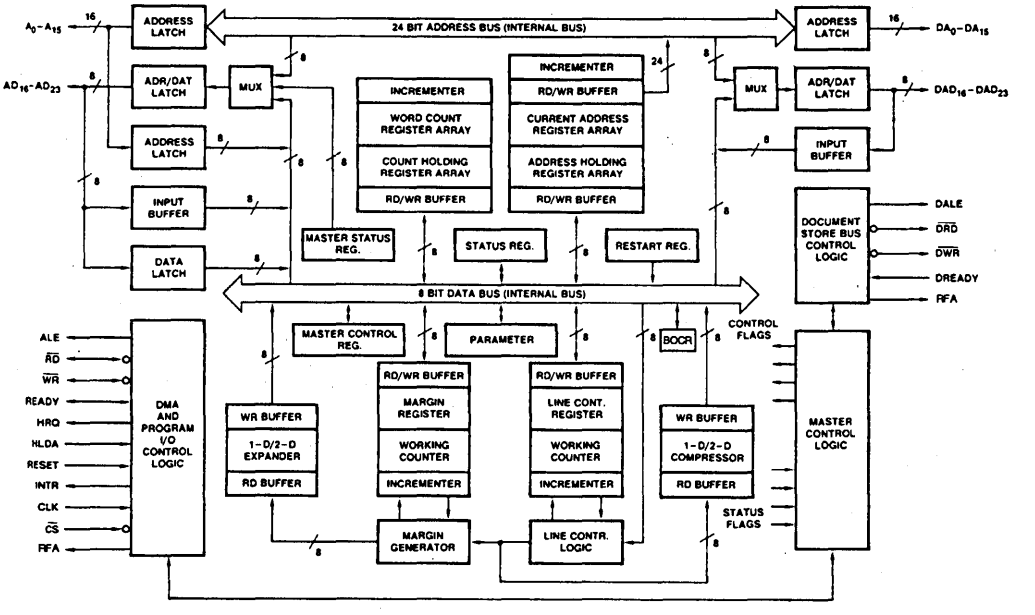
$\overline{DWR}$ is an active-Low, Three-state output signal. A Low on this pin indicates that the $DAD_{16} - DAD_{23}$ bus is being used for a Document Bus write data transfer. When the CEP does not need this interface for a data transfer, this pin is floating.

RFA Reference Line Access (Output, Active-High)

This line indicates a reference line access in conjunction with an active "$\overline{RD}$ or $\overline{DRD}$" signal. If the CEP performs a reference line access either through the system or the document interface, this output is asserted High. It stays Low during all other data transactions.

TST Test (Input)

This input is used for factory testing. It must be tied to Ground.



BD005331

Figure 6. Am7971A Block Diagram

FUNCTIONAL DESCRIPTION

I. OVERVIEW

Figure 6 shows the internal structure of the CEP. The Am7971A contains two separate buses, the System bus and the Document Store bus. One DMA Controller on the CEP chip serves both buses. DMA data transfers cannot take place on both buses at the same time, however, slave transfers can occur on the system bus while a DMA transfer is taking place on the Document Store bus. Data transfers between the Am7971A and Main Memory take place on the System bus. Data transfers between the Am7971A and the Document Store Memory take place on the Document Store Bus.

The Am7971A processes two types of data: uncompressed or image data and coded or compressed data. Image data is stored in that portion of memory called the Image Buffer. Compressed data is stored in a portion of memory called Code Buffer. In an Am7971A system, the Code and Image Buffers are external to the CEP and each can be located in either the Main Memory or the Document Store in any combination.

Consideration should be given to the assignment of the buffers to memory. All control information exchanges between the Am7971A and the host processor take place on the System bus. Because of the high data rate of image data, it is recommended that the Image Buffer be placed in the Document Store. This way, it can be accessed with maximum speed without slowing down the host system. For maximum performance, the Image buffer should be large enough to store one uncompressed document. The Code Buffer can be placed in the Main Memory so that the CPU can access it rapidly during transmission or reception of data. Since the compressed code is considerably smaller than the image data, it does not seriously slow down the system bus and thus does not impact the CPU.

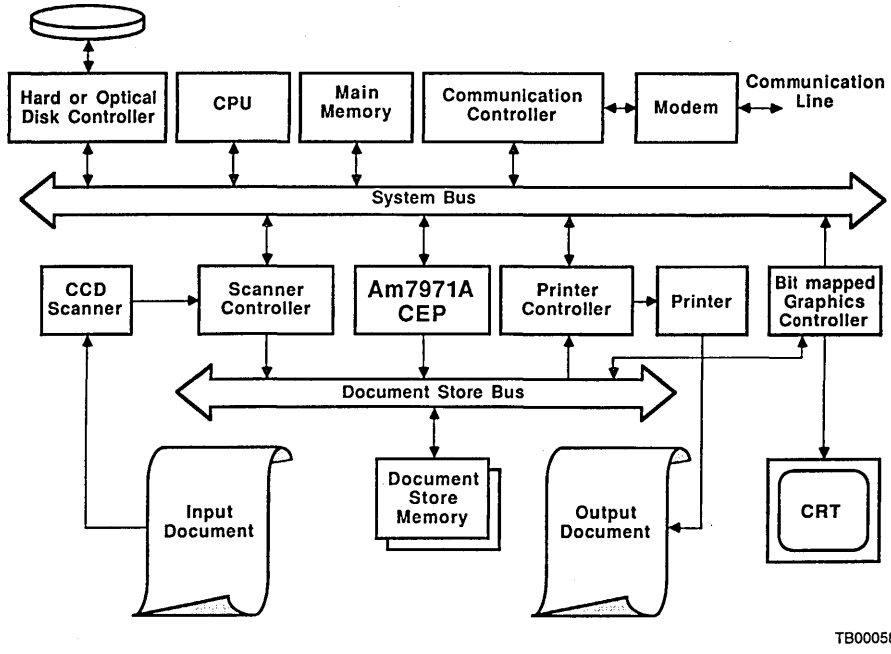


Figure 7. CEP Typical System Configuration

Figure 7 shows a typical configuration with the code buffers in the Main Memory and the image buffer in the dedicated Document Memory. All image handling devices like scanners or printers are located on this bus. It can also be utilized by a graphics processor. The system bus only carries coded data to a transmission line or a mass storage device. Thus, the system CPU is released from any significant task for image handling.

CEP operations consist of three phases: initialization, operation, and termination. In the first phase, the registers (for the compressor or expander) are initialized to specify and control the desired operation. In the second phase, the processing operation itself is started and performed. The final phase involves terminating the selected processor and performing any actions that are appropriate to that termination.

The Am7971A contains registers to specify the starting address and assigned length of both the Image Buffer and the Code Buffer. The Compressor takes image data from its Image Buffer and loads the resulting compressed data into its Code Buffer. The Am7971A Expander takes compressed data from its Code Buffer for processing and loads the resulting image data into its Image Buffer. In an Am7971A system, both the Compressor and the Expander are completely independent and can operate simultaneously.

In principle, the compression and expansion algorithm implemented in the CEP works with any image resolution (PELs per

inch). However, if the resolution is too low, the compression ratio becomes insignificant.

For certain images (such as half tone or low resolution), the compressed data representing a line may be longer than the original line of the image. This is called negative compression. The Am7971A checks for this condition after compressing a line and alerts the host processor via an interrupt and a status bit.

Each compressed line may be delimited by an End of Line (EOL) code according to the CCITT recommendation for Group 3 facsimile apparatus. However, this automatic EOL insertion can be suppressed by appropriate bit settings of the Am7971A.

The CCITT recommendation T.4 for Group 3 equipment requires each coded line to have a certain minimum length. Fill bits are added by the CEP to a short line when necessary to meet this requirement. The Am7971A contains a Time Fill Register to specify the minimum line lengths (including zero).

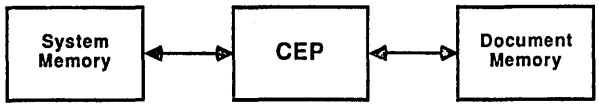
Data is vulnerable to modification by transmission errors. When erroneous data is expanded, the resulting image is very different from the original. The Am7971A checks the expanded line for the number of picture elements required by the specified paper width. If there is a discrepancy, the CPU is alerted via an interrupt.

II. INTERFACE DESCRIPTION

The two interfaces of the CEP both consist of a 24 bit address bus and a 8 bit data bus. The 8 bit data bus is multiplexed with the upper 8 address lines. ALE/DALE is used to latch the upper part of the address; $\overline{RD}/\overline{DRD}$ and $\overline{WR}/\overline{DWR}$ are used to indicate the read or write access of the CEP. Any number of wait states can be inserted in a CEP memory access by keeping READY/DREADY Low. A memory access without wait-states takes 3 clock cycles.

The system interface is designed to perform an iAPX 8086 like bus arbitration for the system bus using the signals HRQ and HLDA. HRQ is asserted when the CEP wants to perform a Master DMA access on the system bus. This request is granted by an active HLDA.

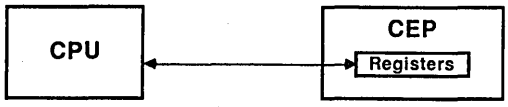
The above sequence is called Master Mode because the CEP performs an independent Master DMA cycle on the system bus (see figure 8). No external DMA device is needed to supply the expander or compressor with data. The Am7971A usually performs one memory access cycle for each bus arbitration cycle. If the begin or end of an image line is not on a byte boundary (see bit boundary image processing) the Am7971A expander performs a Read Modify Write operation in order to mask expanded image bits into a frame picture which results in a two byte burst transfer. It only requests the bus once for this operation. The Read and Write accesses occur back to back which results in a two byte burst transfer.



- **CEP accesses memory for image data and coded data**

TB000570

Figure 8. CEP Master Mode.



- **CPU accesses/initializes registers**
- **CPU polls status**
- **Block Transfer mode (optional)**

TB000560

Figure 9. CEP Slave Mode.

The Am7971A has 46 registers for address pointers, parameters and status information. These registers provide a maximum of flexibility in memory management, format control and operational control over the CEP opertions. They are initialized before a compression or expansion sequence is started. During an expansion/compression operation, three status registers supply information about the current operation of the expander or compressor.

While the CEP is busy compressing or expanding a picture, all internal registers can be accessed whenever the CEP is not using the system bus. This is called Slave Mode (see figure 9).

Since the system interface is used for Master DMA accesses as well as for Slave accesses to the registers, several control signals are bidirectional ($\overline{RD}$, $\overline{WR}$, READY, $A_1 - A_7$). These input/output signals are 3-stated by the CEP when it is not in Master Mode. The CEP recognizes a register access request when the $\overline{CS}$ signal is asserted. All registers are directly addressed through the address lines A_1-A_7 (Only even addresses are used). The data transfer to and from the registers is channeled through $AD_{16} - AD_{23}$. $\overline{CS}$ can be kept Low for consecutive slave accesses (Block I/O Transaction).

With one exception, the registers are not directly connected to the system interface. On a Slave access, an internal microprogram takes care of the data transfers to and from the registers. After data is available from or successfully written into the registers, the CEP responds by asserting READY High. The response time of the CEP to these register access requests varies from register to register. READY might be suppressed between 4 to 20 clock cycles if the CEP is idle and up to 50 clock cycles if the CEP is busy. If the host system aborts the slave access by driving $\overline{RD}$ or $\overline{WR}$ High before the CEP responded to the access by asserting READY High, the slave access is disregarded and no register contents are changed (CPU Program Read/Write Access Abort).

The Master Status register (MSR) supplies the host system with an overview about the current status of the compressor and the expander. Only this register is accessible without delay (4 clock cycles) providing fast access to the status information in polling mode.

III. MEMORY BUFFER MANAGEMENT

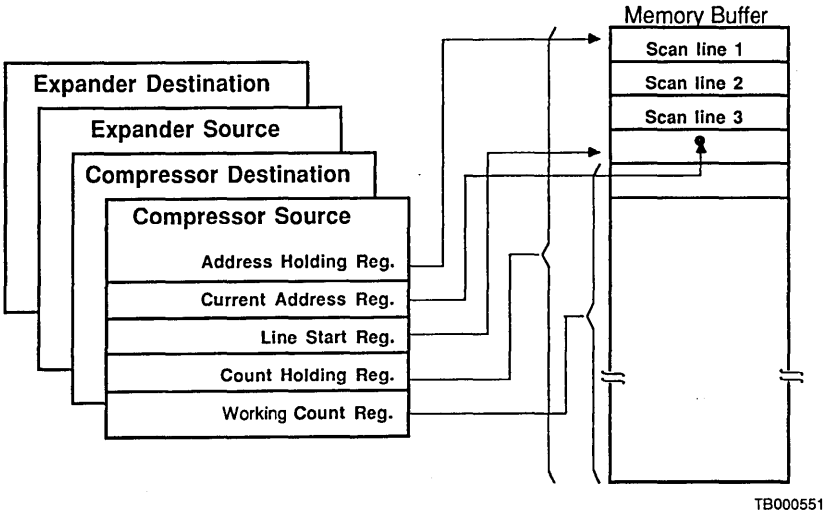


Figure 10. CEP Memory Buffer Management.

Each section of the CEP, the compressor and the expander, needs one set of source and destination buffers. Since full duplex operation is possible, the expander can expand an image from the expander source buffer into the expander destination buffer, while the compressor is compressing an image from the compressor source buffer into the compressor destination buffer. For each of these four possible buffers, there is a separate set of address pointers and byte counters as shown in the Figure 10.

A Buffer is defined by specifying which memory it is in (Main Memory or Document Store), the starting address and the capacity of the Buffer. The source and destination control bits in the expander/compressor master control register determine whether the source or the destination buffer is located on the system bus or the document bus.

The CEP always uses the Current Address Register (CAR) content to access memory for its current operations. It always points to the byte which will be accessed next. The CEP automatically increments this register after each memory access. When the CEP starts processing a new image line it always copies the content of the current address register into the Line Start Address Register (LSR). Thus, when an exception occurs, it is always possible to recover the beginning of the last processed image line and its corresponding code.

For each access to a buffer the CEP increments the corresponding Working Count Register (WCR). Before the CEP starts processing a buffer this register contains the total number (in 2's complement form) of bytes available in a buffer. Hence it always shows the number of bytes remaining unprocessed in a buffer. If the content of WCR reaches zero the CEP terminates its operation and flags a buffer overflow condition. The software has to decide if the whole page is finished or if the operation is to be resumed to complete a page (see exception processing).

For its operation, the CEP only needs and modifies the current registers (current address, working count) and the Line Start address register. In addition to these registers an Address Hold Register (AHR) and a Count Hold Register (CHR) are provided for each buffer to maintain a copy of the initial values. The AHR stores the starting address of a buffer and the CHR stores the 2's complement of the buffer size as a byte count. These Registers are never changed by the CEP but they are used to update the CAR or WCR when the same buffer is used over and over again. This feature reduces the software overhead for updating the address pointers and counter values to an absolute minimum (Refer to the section 'Buffer Overflow' for more details).

IV. PAPER FORMAT CONTROL

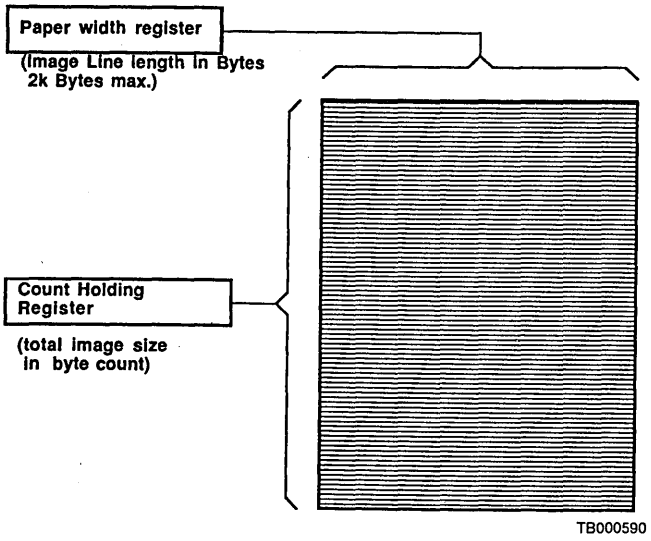


Figure 11. CEP Paper Size Control

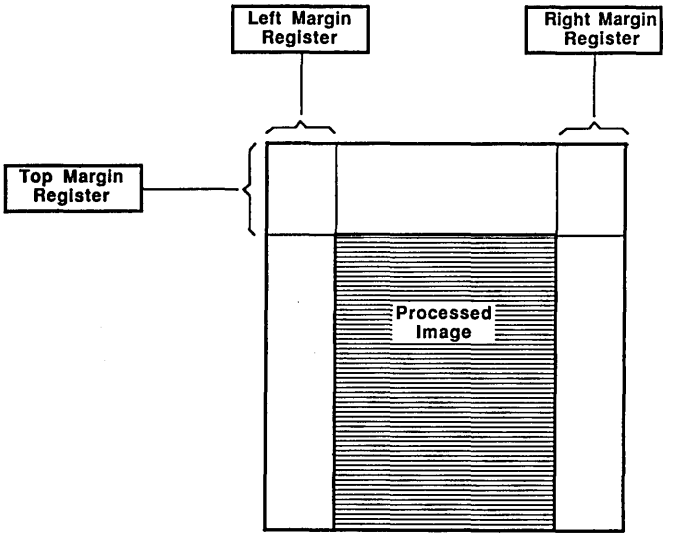


Figure 12. CEP Margin Control

Figure 11 shows, that the width of a document to be processed is defined by the Page Width Register. The length of a document during compression is determined by the user software. The CEP stops compressing whenever the source image buffer is empty. The Count Holding Register contains the number of bytes of image in the image buffer. If the source image buffer specified is smaller than the document, more processing control is required.

The simplest case is when the source image buffer length is the same length as the document (See Figure 12). In this case, the CEP is programmed to append an "end of document" code to the coded image at the end of compression. During expansion, processing continues until the end of document code (RTC for Group 3 or EOP for Group 4) is detected assuming that the destination image buffer is also large enough to hold the expanded document.

If the source image buffer for the compressor is smaller than the document, the document is processed in segments (Refer to the discussion of page fraction processing for details).

During expansion of a document larger than the image buffer, the CEP stops whenever there is a destination image buffer overflow. The CPU must then empty the image buffer and resume the expansion operation. For more details, refer to the page fraction processing discussion.

CCITT recommendation T.4 covers compression and expansion of scan lines up to 2560 bits. The Am7971A accommodates much wider pages by the use of multiple make-up codes. The CEP allows specification of scan line lengths up to 16K bits in both, 1-D and 2D mode operation.

Figure 12 shows how a white margin around the image can be specified for compressor operation. This feature is useful to suppress tolerances in scanner adjustments. Specifying a margin provides a clean surrounding for the scanned image.

The three margin control registers specify white right, left or top margins. If any of these registers are non zero, the compressor reads the image data within these margins but disregards the content and encodes it as white image. If the image following these margins begins with white pixels, the compressor combines them together with the margin into one white runlength code.

Since, by definition, the Top Margin white space is to occur only once per document, the compressor logic decrements the Top Margin Register by one after processing each scan line until it reaches "0", at which time normal compression proceeds. Because of this, the Top Margin Register must be initialized each time a new page is started.

Margins can only be defined in byte boundaries. If the margin continues into white data, a single white code is generated for the margin and the following white pels in the image line.

When the margin specifications are not consistent with the page width, the CEP will terminate operation after setting the Illegal Command bit in the appropriate Status register.

The expander does not have a margin control feature.

V. WINDOW PROCESSING

The CEP provides the possibility to define an image window of any size at any location in memory. The idea is to define a larger image as a frame, and a window as a page within this frame. Figure 13 shows how the frame width register specifies the scan line width of the frame picture in memory.

This ability provides window processing without any additional processing of the image. A partial image area of any size at any location can be compressed from a larger image and expanded into a different location of a target image.

The window is specified like a small page. If the frame width and the page width are the same, the full frame is processed and thus the window processing feature is disabled.

If the frame width is larger than the page width, window processing is performed. The position of the window is specified by the address hold register. The width of the window is specified by the page width register. The length of the window is specified by the number of bytes processed (CHR and WCR).

Whenever the CEP reaches the end of a scan line, it adds the frame width to the image buffer LSR which still contains the start address of the last line. Thus it then points to the first byte of the next line. After this, the same address is automatically copied into the image buffer's current address register (CAR). Then it continues operation from this new position. This scheme is used for the expander destination buffer and for the compressor source buffer (see also the description of the Line Start Register under buffer overflow processing and Figure 19).

Any memory location outside this defined window is untouched. If the expansion results in a longer runlength than specified in the page width register, which might happen due to a data error, the CEP will skip outputting any bits beyond the specified line length.

Margin control is effective during window processing. Therefore, it is possible to have a window with white margins.

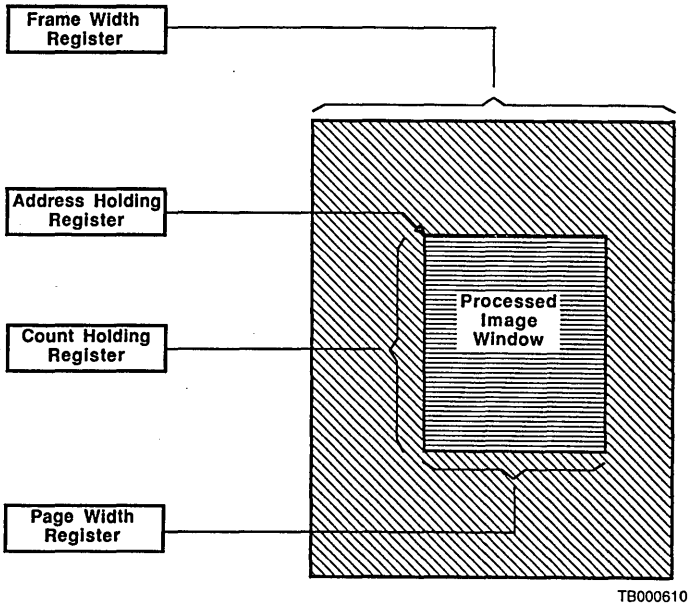


Figure 13. CEP Window Processing

VI. BIT OFFSET CONTROL

The Am7971A allows specification of an image scan line on any possible bit boundary. This is useful in preserving the information at the edges of documents which are scanned on a bit boundary, especially important for window processing. The Am7971A can compress from a window at any bit position in a picture and expand it into any position in a destination image.

To specify the left and right bit offset, the Am7971A provides a register for the compressor and the expander, called the expander or compressor bit offset control register (EBOCR/CBOCR). Figure 14 shows how the contents of these registers correlate to the definition of a bit offset at the right or left edge of a page (or window). Any Bit Offset between 0-7 bits can be selected. The Bit Offset specifies the number of bits in the outermost byte of a line that are not used. During expansion, the Am7971A performs a read-modify write operation at the line extremes to patch the bit offset into an existing image.

Note: If the bit offset value is "0", the Am7971A behaves exactly as the Am7970A with line termination parameter = "0".

VII. CODE FORMAT CONTROL

The CEP generates a code format as recommended by CCITT for Group III and Group IV apparatus. This standard coding procedure was originally intended for facsimile transmission but is also widely accepted as standard for graphics workstations and electronic archiving purposes. Since not all image processing environments necessarily need to follow the CCITT recommendation precisely, the CEP offers a variety of coding options which can be used to gain a higher compression ratio. The usefulness of these options depends on the specific application and must be evaluated from case to case.

Since the CEP maintains great flexibility in the code format to be used, much care must be taken in the correct initialization of the registers which affect the format of the compressed image. Only a specific combination of parameters will produce a Group III format, while a different combination will produce Group IV code. The internal microcontroller checks all parameters before starting compression or expansion, and determines if it is a Group III or Group IV data format.

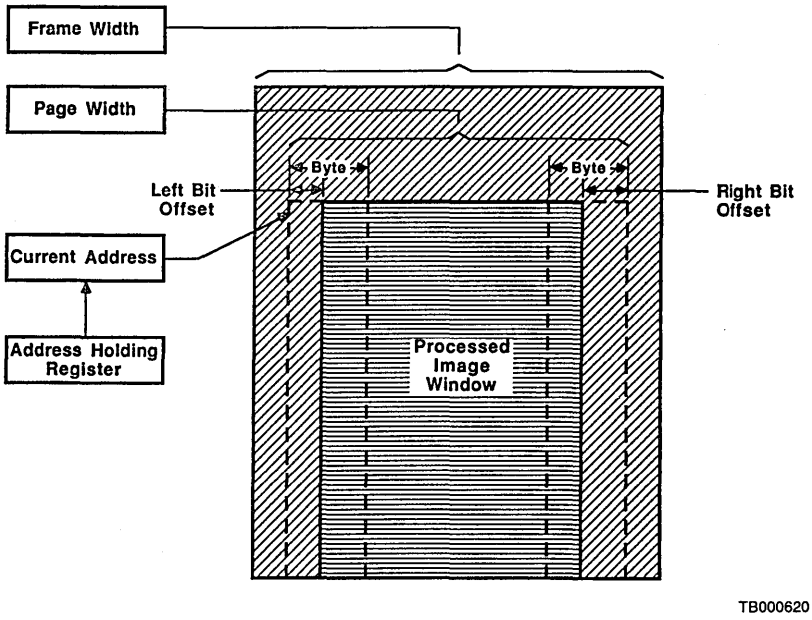


Figure 14. CEP Bit-Boundary Image Processing With Window

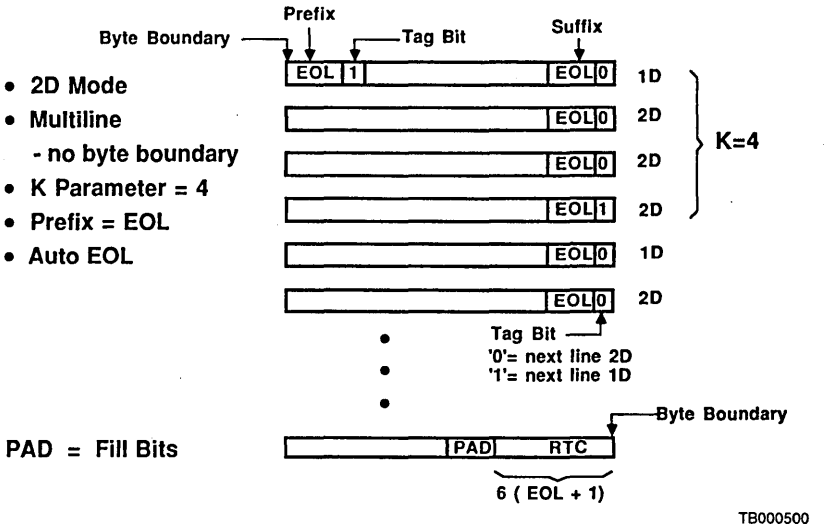


Figure 15. CCITT Group III Format Control

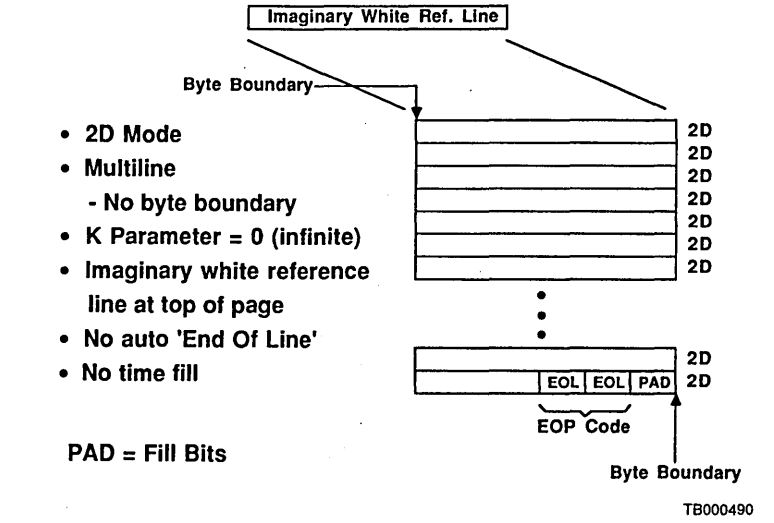


Figure 16. CCITT Group IV Format Control

Figures 15 and 16 are examples of Group III and Group IV code formats and show the differences of the selected parameters.

Auto EOL Control:

The EOL code terminating a coded image line plays an important role for resynchronization of the expansion process after the occurrence of corrupted data from transmission errors in Group III coded data. The compressor terminates each coded line with an EOL code if "Auto EOL" mode is selected in the Compressor Parameter Register (CPR). The expander recognizes EOL as the end of a coded line if "Auto EOL" is selected in the EOL control bit of the expander parameter register (EPR).

Byte Boundary Control:

The CEP is able to adjust the end of a coded line to either bit or byte boundary. After an exception condition, such as negative compression or data error, it is easier to resume from a byte-boundary adjusted line. However bit-boundary lines result in higher compression and throughput. The CCITT Group III standard leaves this option up to the user.

If the "byte boundary" option in the compressor parameter register (CPR) is selected, the CEP compressor adjusts each coded line to a byte boundary. It does that by adding fill bits (= "0") between the compressed image line and the EOL code. If a byte boundary adjusted compressed image is to be expanded, the expander knows from the expander restart control register (ERCR) whether to disregard these fill bits or not. If the expander input is byte boundary adjusted but the expander set to no byte boundary operation or vice versa, a data error will occur and the operation is terminated. However, if Auto EOL is programmed, the expander ignores the byte boundary control bit (fill bits are automatically disregarded if there are any).

Coding Mode Control:

The "mode control" bits in the Master Control Registers (EMCR/CMCR) determine the coding scheme used to compress or expand an image. The choices are 1-D, 2-D and transparent mode.

One Dimensional Mode (1-D):

In one dimensional mode, the CEP processes an image by the Modified Huffman coding scheme only. Image lines are scanned for groups of pixels with the same color (black or white) from left to right. The number of pixels between two color changes is the color runlength. This runlength is passed through a coding table. Two different code tables are used for black and white runlengths. The lengths of the codes are optimized according to the statistical probability of occurrence of a runlength.

For each new line a new code is started. Each line is assumed to start with a white pixel. If it actually starts with a black pixel, a white zero length code is inserted. The codes for each line are independent from previous or following lines. Thus the Huffman coding scheme is a one dimensional (1-D) or horizontal coding method (Refer to the CEP technical manual for detailed information about the coding methods and the CCITT specification).

When the CEP is in 1-D mode and auto EOL operation is selected, it will not append a Tag bit to any EOL since all lines are coded with the same method. A Tag Bit is used to distinguish the coding mode of the next line.

Two Dimensional Mode (2-D):

In two dimensional mode, the CEP processes data by the modified READ (Relative Element Address) coding scheme. This method detects similarities between two adjacent lines. Only the differences between lines are coded into very short code words of varying length. Since normal text images have a lot of equal or similar adjacent lines, this coding scheme is very efficient for such documents. To generate the code for a scan line, the compressor or expander must access the previous scan line for reference. Thus the READ coding scheme is called a two dimensional or vertical method.

The two dimensional scheme only deals with relative positions of color changes between two lines. If the first line of a document is to be coded in 2-D mode, it is necessary to assume an imaginary white reference line on top of a page. Starting from this assumption, it is possible to recover the image by relative information only. The vertical coding scheme is only efficient for small differences between lines. The READ scheme falls back to the Huffman scheme when the differences between two lines are too big (Refer to the CEP technical manual for detailed information about the coding methods and the CCITT specification).

The CCITT specification uses the 2-D mode in two different ways. In Group IV environments, READ code is used exactly as shown above (see also figure 16). All lines are coded in the 2-D scheme only. In Group III environment, the 2-D coding scheme is used in conjunction with the 1-D scheme. After N lines are encoded in 2- D, one line is encoded in 1-D. The K-parameter determines the number of 2-D lines for each 1-D coded line (K = N + 1). In this mode each coded line must be terminated by an EOL code including a Tag Bit which indicates the coding scheme of the next line (Also see figure 15).

The Tag Bit:

The compressor appends a Tag Bit to each EOL code when 2-D coding mode is selected. The expander determines from the Tag Bit the coding scheme of the following line. In 1-D mode the Tag Bit is unnecessary and therefore omitted.

EOL code for 1-D Mode: 0000 0000 0001
2-D Mode: 0000 0000 0001T

T = Tag Bit
= 0 if next line is 2-D
= 1 if next line is 1-D

In Group III (2-D) mode, when the SA bit is set (e.g. at the beginning of a page), the expander does not simply assume a 1-D coded line but expands the line according to the Tag Bit appended to the prefix EOL. A Tag Bit is also appended to each EOL code of the RTC suffix in Group III coding mode. For this purpose the Tag bits are always set "1". If the EOL codes are omitted (auto EOL mode off), the expander decodes data according to the K-parameter.

Transparent Mode:

In addition to the one dimensional and two dimensional scheme, the CEP provides an optional transparent mode. This mode disables the coding/encoding algorithm but keeps all format control parameters in effect. If data is not to be changed when being transferred by the compressor or expander in transparent mode, all format control parameters must be disabled (no EOL, no byte boundary, no SA, granularity = 0, express mode = 0, no time fill, no margins). Alternately, the expander in transparent mode can be used to strip off the prefix and suffixes by leaving Auto EOL mode, SA bit and RTC or EOP format on.

The transparent mode provides a plain DMA channel for data moves between the system interface and the document interface and for general DMA actions on either one of the two interfaces. While the CEP is busy expanding or compressing in half duplex mode, the unused counterpart could also be used for DMA transactions of the next picture to be processed.

K-Parameter:

If the image was compressed in 2-D only, it is impossible to recover a useful image after a data error. The CCITT. therefore, specifies for Group III type data a factor called the K-Parameter. Using this K-Parameter 1-D coded lines are interleaved with 2-D coded lines. There are K-1 lines coded in 2-D mode for each 1-D coded line. Unlike the 2-D mode, the expander in 1-D mode does not need to access an already expanded image line as reference for the next line. Therefore, a data error can be stopped from propagating through the whole remaining image by these 1D coded lines.

Since a maximum of K lines can be corrupted by a data error, the K-parameter represents the maximum tolerable distortion of an image. Thus, the K-parameter depends on the vertical resolution of an image and the probability of data errors during transmission.

The K-Parameter Register (CKPR) is always needed for 2-D mode compression. The expander disregards its K-Parameter register (EKPR) in CCITT Group III mode because the Tag Bit appended to the EOL suffixes indicates the coding scheme of the next line to be expanded. The expander K-Parameter is only used when suffix EOL codes are omitted (Non CCITT compatible mode, auto EOL mode off).

Examples of parameters for different formats:

The mode control parameter (in the MCR), the EOL field in the EPR/CPR and the K-parameter provides among others the following methods of decoding/encoding the image:

- 1D mode and Auto EOL
  All lines are coded in 1D mode only and terminated by a EOL code without a Tag bit.
- 2D mode with K = 1 and Auto EOL
  All lines are coded in 1D mode only and terminated by an EOL code with a Tag bit (= "1").
- 2D mode with 256 > K > 1 and Auto EOL.
  Lines are coded in 2D mode with interleaving 1D coded lines. Every line is terminated by an EOL code with a Tag bit indicating the coding scheme for the next line.
- 2D mode with K = 0 (infinite)
  All lines are coded in 2D mode. Each line is terminated with EOL code plus Tag bit (= "0") if Auto EOL is selected. Otherwise no EOL is appended.

Source Attribution:

The CCITT Group III recommendation defines a prefix indicating the beginning of a compressed image. For this prefix a single EOL code including a Tag bit is used. Group III apparatus starts transmitting signals as soon as the connection is established. The first EOL among these signals indicates the beginning of a compressed image page.

The system program must set the Source Attribute bit, SA, in the Parameter Registers (CPR/EPR) if the CEP is to recognize this prefix. If the SA bit is set, the CEP does not start expansion of the input data until it has received an EOL code. This feature is also very useful in searching for the next EOL in the code after a data error has occurred (see exception processing).

The Source Attribution bit must also be set before Group IV data is processed. Since all lines are coded in 2D mode, there is no reference line available to expand the first line of a page. For Group IV processing, the CEP must assume an imaginary all white line as reference for the first image line. The CEP does this if the following parameter setup is specified: No auto-EOL and K = 0 and 2-D mode and SA = 1. If the SA bit is not set, the CEP will not assume an imaginary white line (Refer to the technical manual for details on this non CCITT format).

The SA bit is automatically reset by the CEP after processing of the first line. It usually does not need to be modified by the system until a new page is started or expander data error recovery is being attempted.

Format Control:

The two coding schemes also differ in the suffix used to indicate the end of a compressed image. Group III coding uses a sequence of six EOL's to indicate the end of the coded page. If 2-D mode is selected, each EOL is appended by a Tag bit set to '1'. The 1-D scheme uses EOL codes without Tag Bit. The compressor generates this type of suffix on a source buffer overflow if suffix RTC code format is selected in the parameter register (CPR). For non-Group IV processing, the expander needs three EOLs to recognize the end of a compressed image.

Group IV coding uses an EOP code to indicate the end of a compressed image. The EOP consists of two plain EOL codes without any tag bits. The compressor generates these two EOLs on a source buffer overflow if Suffix EOP Code is selected in the Compressor Parameter Register. When Group IV processing is selected the expander needs a sequence of two EOL codes to recognize the end of a compressed image.

If the source buffer is smaller than the entire document, a source buffer overflow may not necessarily indicate the end of a page. If it is not an end of page, RTC or EOP suffix should not be appended. Therefore, the first part of the image must be compressed in byte-boundaries or non byte-boundaries mode without selecting a suffix. When the last portion of the document is being processed, the data format control is switched to Suffix RTC or EOP code mode. Alternatively, if the size of the image buffer is large enough to permit processing of an entire image without stopping, the suffix options can be selected right at the beginning.

Both suffix RTC and EOP code formats imply non Byte-boundary formatted code. If the image has to be compressed in byte boundary mode, all scan lines except the last one have to be compressed with the byte boundary mode selected. Then for the last line the compression must be resumed with either Suffix RTC or EOP code mode and single line operation mode selected.

Changing modes between lines must be done very carefully because it might corrupt overhanging code from the previous line which is still in the output pipeline (Refer to the technical manual for more details).

In single line operation each compressed line is suffixed with the code selected in the Parameter Register.

When the expander recognizes either an RTC or EOP code, it terminates the operation and sets the EOP Code Detected flag in the Master Status Register (MSR).

Fill Bits:

If a coded line and the RTC code does not end on a byte boundary, the compressor adds fill bits between the compressed data and the RTC code to end the line on a byte boundary. Alternatively, if the code is terminated by an EOP code, the CEP adds fill bits after the EOP code to end the code on a byte boundary. Fill bits are always "0".

The compressor also inserts enough fill bits between a coded line and the EOL code to end a code line (including the EOL) on a byte boundary.

Time Fill:

The CCITT recommendation for Group III data requires each coded line to have certain minimum length. This is necessary for compatibility with older and slower equipment which cannot print fast enough.

The Time Fill Register (TFLR) specifies the time fill parameter which defines the minimum number of bytes in a coded scan line. If the generated code line is shorter than this, enough fill bytes (='0') are added between the code and the EOL to meet the minimum requirement.

The Time Fill Parameter can be calculated by the following formula:

TFLR = MSP * MTT / 8

MSP : Modem Speed
MTT : Minimum Transmission Time

When the Auto-EOL feature is supressed, the Am7971A ignores the time fill requirement; no time fill is inserted.

Bit Ordering in code and image bytes in memory:

In memory, the bits representing the pixels are stored as bytes. The first pixel at the top left corner of the image must be stored as the least significant bit of the first byte in the memory buffer. This is also the first bit to be transmitted. The compressed image follows the same rule.

This is contrary to normal computer convention where the LSB is the rightmost bit in the byte. Hand-coded data from the coding table (following traditional computer rules) result in the wrong bit order in each byte, and, therefore, requires the bit order in every byte to be reversed (bit 0 → bit 7, bit 1 → bit 6 etc).

Theoretically the bit ordering of image data does not matter, but the wrong bit order will almost always result in additional, unnecessary color changes, and therefore upset the compression ratio.

Note also, that scanners are not standardized. Some put the leftmost bit into the LSB (as required by the CEP), some do the opposite and, therefore, require reversal of bit ordering either by software or by hardware.

VIII. NON-CCITT OPTIONS

Wraparound Mode:

The wraparound mode groups a specified number of scan lines together into one effective line during encoding. This mode cannot be used simultaneously with 2-D compression or Express Mode. In 1-D compression, it gains approximately 15% code efficiency because fewer EOL codes are used and longer runlength can be encoded using multiple make-up codes. This coding scheme is not compatible with the CCITT recommendations.

The number of lines grouped together is specified by the content of the wraparound register plus 1. If the value is "0", then the effective line is identical to a scan line.

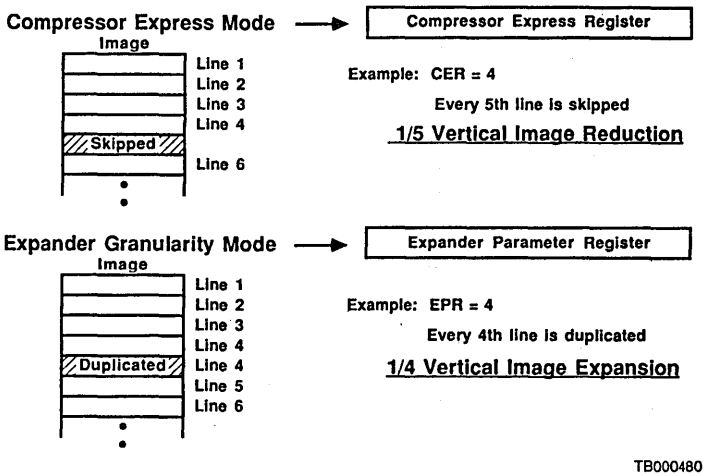


Figure 17. CEP Express and Granularity Modes

Vertical Image Resolution Conversion:

Figure 17 shows an example of the CEP's capability to reduce or expand the image vertically during compression or expansion. This feature is not compatible with the CCITT recommendation.

Express Mode:

The compressor express register specifies how many scan lines to compress before skipping one line. For example Figure 17 shows how every fifth line will be skipped resulting in a vertical image reduction of 20%. A '0' value in this register disables this feature.

If the Express Mode is defined with 2-D Compression, each line that is compressed is also the reference line for the next line that is compressed. Skipped lines are not used as reference lines.

Granularity Control:

The G-Parameter bits in the Expander Parameter Register contain the granularity factor (G-Parameter). It specifies how many lines to expand before duplicating the last line expanded. For instance, Figure 17 shows an example with G = 4. Here, every fourth scan line is duplicated resulting in a 25% vertical expansion.

IX. PROCESS CONTROL

Compression/Expansion Sequence:

Conceptually, the CEP operates as a coprocessor. The expansion and compression process is performed with minimum host CPU's assistance. Since the CEP is equipped with full Master DMA capability once a compression or expansion process is started, a whole page could be processed without any CPU intervention.

A compression or expansion sequence has the following steps:

- The host system provides the data in a memory location accessible by the CEP.
- The host system issues a software reset to the CEP
- The host system initializes the CEP's registers.
- The host system starts the CEP.
- The CEP accesses the data in the specified memory buffers and processes an image until a full page is completed or an exception condition occurs.
- The CEP alerts the system about the termination of its operation by asserting an interrupt.
- The system examines the CEP status registers to determine the cause of the interrupt. According to this information, it updates the memory buffers and initializes the CEP. The CEP resumes its operation or starts a new sequence for the next page.

The CEP executes three different operations, the Software Reset, Single Line Operation and Multi Line Operation. They are selected by the Operation Control Bits in the Master Control Register (CMCR, EMCR) and started by setting the 'GO' bit. The 'GO' bit is automatically reset after completion of the selected operation.

- The Software Reset clears the internal working registers, process control flags, the status and interrupt registers, sets "busy" to zero, flushes the input queue, and sets up the check for configurational errors and flags them. It is generally used to bring the CEP into an idle condition from where it can start on a new page. The Software Reset does not alter any user programmable registers.

- The Single Line Operation terminates the expansion or compression procedure after each processed effective image line. An effective image line can be longer than a single scan line if wraparound mode is selected (see options). The single line operation is useful for debugging, in systems with single line buffers. It is also used in some special situations (processing the last line in byte boundary mode, processing of negative compression or data errors).

- The Multi Line Operation terminates the process when an exception condition occurs. All pointers are automatically updated internally for each new image line. In this mode, the number of lines to be compressed or expanded is determined by the specified buffer sizes. Alternately the expansion can proceed until the end of a page is detected from a terminating code (RTC or EOP).

An entire image may be compressed or expanded in one operation if the code buffer and the image buffer are both large enough to contain the entire image. In this case the system program must specify a software reset operation before starting the compression or expansion of each new document.

The code buffer (compressor destination, expander source) can be smaller than an entire document. The CEP stops at the end of the buffer indicating a buffer overflow or underflow (This document makes no distinction between overflow and underflow of any buffer). Processing can be continued without issuing a reset after the buffers are updated. Similarily, the CEP is able to operate on image buffers that are smaller than the entire document. (Note: image buffer size should be an internal multiple of page width-PWR.)

Full duplex operation:

The Am7971A has two different operating configurations. In the full-duplex mode, the Expander and the Compressor are operated simultaneously. In the half-duplex mode, either the Expander or the Compressor are operating. Setting a '1' in the GO bit of the CMCR starts compression. A "1" in the GO bit of the EMCR starts expansion. For full-duplex operation, load a "1" into the GO bit of each register.

The expander and compressor sections work independently during full duplex operation. Either one can terminate its operation separately after an exception condition and assert an interrupt. The cause of the interrupt can be observed from the status registers and the terminated section can resume operation while the other section is still busy.

Initializing the CEP:

The Am7971A has the following initialization requirements:

- Source Buffer definition
- Destination Buffer definition
- Attributes
- Control Parameters
- Restart Condition
- Operating Mode
- Processing Mode
- Data Format
- Paper Format
- Minimum transmission time
- Options

These requirements are met by writing appropriate information into the 46 registers in the CEP. The system program should specify certain initial conditions before starting the operation of the Am7971A. After the CEP is started by setting the GO bit in the Master Control register, the CEP checks the parameters for consistency. If it finds an illogical condition, it terminates.

X. STATUS CONTROL

The Status Registers:

The CEP has three status registers: the Master Status Register (MSR), the Compressor Status Register (CSR), and the Expander Status Register (ESR). The MSR provides information about the general status of both the compressor and the expander. ESR and CSR inform specifically about the expander or compressor.

Most important are the EBY and CBY Bits in the MSR. The CBY Bit is set High when the compressor is busy and Low when the process is terminated. The EBY Bit provides the same information for the expander side. The CBY in the CSR and the EBY in the ESR are directly hardwired to those in the MSR (The meaning of all other status bits is explained under exception processing).

Interrupt Handling:

When interrupt mode is selected, the interrupt (INTR) signal is asserted upon termination of a process. This is when the busy bit returns to "0". The interrupt mode is selected by setting the interrupt enable bits (CIE,EIE) either in the expander or compressor master control registers (EMCR,CMCR). The interrupt can be enabled separately for the expander and the compressor.

The INTR line will remain High until the MSR has been accessed by the system. The system program must test the MSR register to distinguish Compressor interrupts from Expander interrupts by reading the busy bits. The system program then isolates the cause of the interrupt by reading the appropriate status register (CSR or ESR). Reading the Master Status Register clears the interrupt. The system program may then execute its interrupt service routine to respond to the interrupt.

Polling the Status:

If interrupts have not been enabled, the system program should periodically poll EBY and CBY by reading the MSR register. Polling the MSR is much faster than polling ESR or CSR because this register is directly accessible by the system without significant delay. All other registers are accessible only through the internal microprogram which is interrupted by a register access attempt of the system. Since this procedure is rather slow it is not recommended to access any other registers beside the MSR while the CEP is busy. System and CEP performance would be degraded.

Interrupt mode is better than polling operation because it frees the system from monitoring the busy bits. It is by far the most efficient way to control the CEP.

XI. EXCEPTION PROCESSING

The CEP will terminate its operation when there is a deviation from the normal compression or expansion sequence. These deviations are called exceptions and always require system interaction. An exception from normal processing occurs at the end of a page (EOP detected), on a buffer overflow, if negative compression is detected, when a data error or an illegal extension code is detected during expansion, if an illegal command was issued, or if the system aborts the procedure. Handling these exceptions properly is key to a successful expansion or compression of an image page.

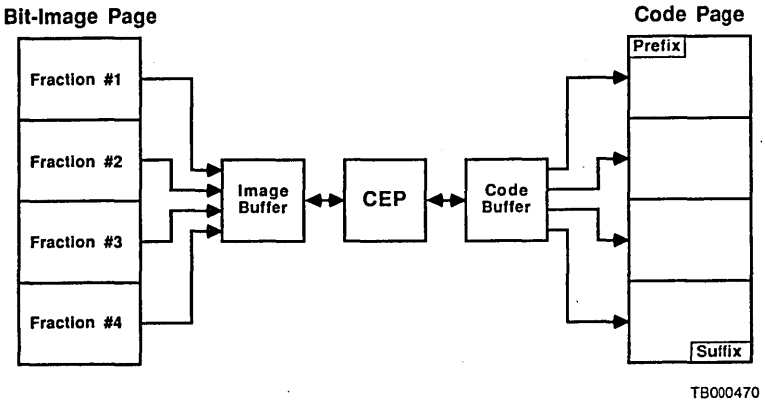


Figure 18. Fractional Page Processing (Memory Buffer Smaller Document Size)

Buffer Overflow (Operating on fractional code or image buffer):

The CEP is designed to work with any amount of memory available in a system. Thus the size of the source and destination buffers for expansion and compression may vary from 1 Byte to 16MByte. The sizes are defined in the Count Hold Registers. If a buffer is not large enough to accommodate the code of an entire document or the entire image data of a document, the CEP will run into an overflow condition.

When the CEP finishes processing the last byte of one of the buffers before the expansion or compression process is completed, it terminates the operation, asserts an interrupt and indicates this event by resetting the Busy Bit (Master Status Register) and setting the buffer overflow bits in the Expander or Compressor Status Registers.

After a buffer overflow, the CEP provides all the information needed to proceed with the compression or expansion from any boundary condition without producing code or image inconsistencies. There is only one restriction: since the CEP is designed for operation on image scan line image buffer must always be specified as an integral multiple of a scan line as defined in the page width registers (EPWR/CPWR) or an integral multiple of wrapped line length (if EWR/CWR is non-zero).

The buffer overflow status:

The host system exercises the CEP's status registers to determine which buffer is exhausted. There can be an overflow of the source or destination buffer or a simultaneous overflow on both sides. During full duplex operation four overflows can happen at the same time (expander/compressor source/destination). The overflow bits in the ESR and CSR indicate the source of an overflow.

Updating memory buffers after overflow:

The system must now decide if a destination buffer has to be saved or a source buffer has to be loaded with new data. If the two dimensional coding scheme is used, care must be taken that the last complete image line in the compressor source buffer is preserved, since it is needed as reference line for the following line except for the following conditions: when the two dimensional code is intermixed with one dimensionally coded lines (CCITT Group III) the reference line may be overwritten if the user specifies the buffer size such that the CEP compresses the first scan line at the beginning of a new buffer in the one dimensional coding scheme.

The system's response to an exception condition is crucial for the overall performance of the CEP. Therefore, the procedure for recovery from a buffer overflow condition needs to be highly optimized. The address hold registers and the restart

control registers were implemented to minimize the number of register accesses necessary to resume from a buffer overflow condition.

Optimizing the resume operation:

The address hold registers (EDAHR, ESAHR, CDAHR, CSAHR) keep a backup of the initial start positions of all memory buffers. The Count Hold Registers (EDCHR, ESCHR, CDCHR, CSCHR) keep a backup of the initial sizes of all buffers. The Restart Control Register (CRCR, ERCR) specifies whether or not an Address Hold Register is used to update the Current Address Register or the Count Hold Register is used to update the Working Count Registers.

Each bit in the Restart Control Registers specifies for a specific register whether its content is untouched (continue) or if it is loaded from the corresponding Hold Register (Restart) after the 'GO' bit is set for resumption of CEP operation. Accordingly, two different resume procedures are possible. The system can disregard the Hold Registers, initialize and update all Current and Working Registers and always keep the Restart Control Bits on Continue Mode. Under these circumstances, the Hold Registers are not used by the CEP at all. Alternately, the system can initialize the Hold Registers once at the beginning. Then the Restart Control Bits are used to specify which working registers are to be automatically loaded from the hold registers on start or resumption.

A typical system interaction:

The second choice is much more efficient because no address pointers have to be transferred. A typical system interaction on a buffer overflow exception would only consist of four CEP register accesses :

- Read Master status register. The MSR informs about the reason for terminating.
- Read the status of the CEP section in use (compressor or expander). The status register provides all the information needed to evaluate a situation and to decide what action to take and determine how the CEP can be initialized for a resume operation.
- Write the correct resume pattern (restart or continue) into the Restart Control Register. The use of the Address Hold Registers in the CEP by a Restart condition avoids any additional update of address pointers.
- Set the "GO" bit in the MCR.

If, for example, a source buffer overflow occurred, all Restart Control Bits are set to continue except the Source Address Control bit and the Source Count Control bit. Then, after the CEP is started, it will continue processing the destination buffer from the location where it stopped. In contrast, the CEP starts processing the source buffer from the beginning as defined in the Hold Register for a number of bytes as defined in the Count Hold Register.

Note, that a software reset is fatal for a proper resume operation since it clears the whole internal status and flushes the internal pipelines. If a software reset occurs after a buffer overflow termination, the CEP cannot encode or decode a consistent code string after resumption.

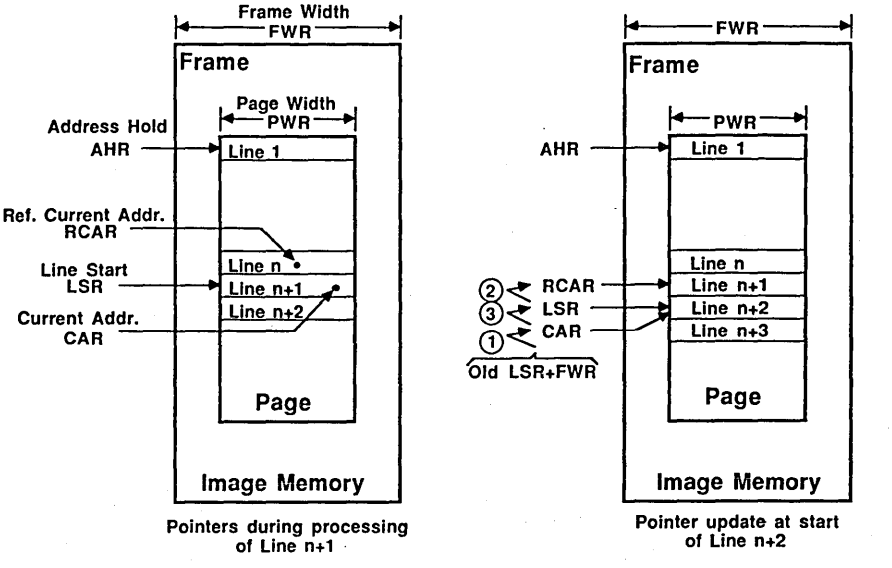


Figure 19. Address Pointer Management (During Multi-Line Operation)

The Line Start Register:

The Line Start Address Registers (EDLSR, ESLSR, CDLSR, CSLSR) have an important role for the management of the memory buffers and for proper recovery from data errors or negative compression. During multi-line operation the LSR is used to calculate the begin of consecutive image lines.

Figure 19 shows three important pointers used for processing an image line (expanding or compression). While the RCAR

and the CAR move through the reference and the current line, the LSR points to the first byte of the line currently being processed. After a line is completed, a new CAR is calculated by adding the frame width to the LSR. Then the previous line start address is loaded into the reference current address pointer. Finally the new CAR is loaded into the LSR and processing of the next line can begin. The Address Hold Register should normally be pointing to the beginning of the memory buffer.

Beside the Line Start Register for the image buffer (EDLSR, CSLSR) the CEP also provides LSRs for the code buffers (ESLSR, CDLSR). These registers always point to the EOL code (suffix) of the previous line or to the byte preceding it. They are updated after an EOL code is generated by the compressor or detected by the expander.

The Line Start Registers are very useful when an exception occurs because the information about the start position of the image line and its corresponding code is still available to the system. The system can also modify the LSRs to adjust the CEP's memory management to changed memory definitions. This is necessary for example during recovery from a data error (see data error) or for resumption after certain types of buffer overflow as described below.

Specifying the Line Start Registers:

At the beginning of a page the CEP is usually started in Restart Mode. Thus all working registers are loaded from the hold registers. If this is the case (Line Start Address Control Bit = '0'), the LSR is loaded from the address hold register. From there on 'Continue' Mode is selected, so that the LSR is only modified by the CEP for each new line it starts to process.

The Reference Line Register:

As shown in figure 19 in 2-D Compression Mode the content of the LSR is loaded into the Reference Current Address Register (ERCAR,CRCAR) before the CEP starts processing the next line. They serve as internal address registers pointing to the next address in the image buffer which the CEP will access for reference. Like the current address registers, they are always incremented by one. Altering the content of the Reference Line Registers will most likely corrupt the CEP's operation. However, there is no problem in reading them for debugging purposes. Also, it is necessary to write into these registers under certain conditions during expander error recovery from a transmission error.

If the position of the reference line must be changed, it can be specified through the LSR if 'Continue' operation is chosen and a new line is started. This is important in 2-D mode when the source buffer for the compressor is exhausted. When the system updates this buffer with new image data, the reference line is lost. The compressor starts a new line at the beginning of the new source buffer but expects the reference line still to exist at the end of this buffer. It thus generates incorrect code. To avoid corruption in this situation, the last line of image which was just processed, must be copied to a memory location outside this memory buffer, before the image lines are loaded into the source buffer. Before the compressor resumes, the position of the copied line is loaded into the CSLSR and the Compressor Source Line Start Control Bit set to continue. The CSCAR must be loaded with the start address of the source buffer (eg. from the Hold Register). The back-up line is then used correctly as reference line for the first line of the updated buffer.

For Group III processing this procedure can be avoided by specifying the compressor source buffer size as an integral multiple of the line length multiplied with the K-parameter. The source buffer will then always begin with a line which is coded in 1-D mode and no reference line access is necessary. On the expander side the the image buffer is not overwritten by the system, so it is still available at its old position after resumption. Therefore, no special expander destination overflow handling of the EDLSR is necessary in 2-D mode.

The Line Incomplete Bit:

The Line Incomplete Bit (LPI) in the Status Register is set when exception situation occurs while the CEP is in the middle of processing a scan line. The LPI of the Compressor is also set if there is partial code remaining in the CEP which has not yet been written out to the memory. This happens during non byte boundary processing because part of the code string has to be concatenated to the next line. LPI is always set on a data error during expansion (see section on data error).

Data Error:

The Group III CCITT-specification assumes the possibility of corrupted data because of transmission errors. If an image is coded in 2-D mode only, the remaining image after a data error would be lost. For this reason 2-D coded lines are interlaced with 1-D coded lines which are used by the expander to resynchronize the decoding procedure after a data error.

The code must be carefully screened for inconsistencies to keep distortion in the expanded image as small as possible. All logically possible methods to check the consistency of the coded input are implemented in the CEP, and it will cause a termination of the process with the Data Error Recognized Bit (DER) set in the Expander Status Register (ESR) if an inconsistent code is detected.

However, because of ambiguities of the coding scheme, not all data corruptions can be detected. There are situations in the 2D coding scheme where the code matches perfectly with the reference line and results in a correct image line length but still does not produce the correct image. An unpredictable number of lines are expanded after such a data error, until the code expander realizes a data corruption. A similar problem arises if the Tag-bit in the EOL code is flipped. The expander encodes a 2D line as a 1D line or vice versa and might not be able to detect the EOL code for an unpredictable number of lines. This is a problem with the coding scheme, not a lack of intelligence in the CEP.

Illegal Extension Codes:

A special case of data error is an illegal extension code which also terminates the process and sets the Extension Code Detected bit (ECD) in the MSR. In case of CCITT-compatible code this is always a data error and should be handled in the same manner. In non-CCITT compatible environments, this feature can be used to implement a variation from the specified CCITT coding table by using other extension codes than specified. This would have to be done by additional software. The expander detects these extension codes by flagging an illegal extension code. The software can than take over and resume processing the following code string.

The Error status:

The DER bit is set under the following circumstances:

- An illegal code is detected. The CEP stops immediately after detecting the illegal code, but the Current Address Register might point a couple of addresses ahead (The CEP prefetches up to 3 bytes in advance).
- A code expands into a negative runlength. This is possible in 2D code since the reference points in the reference line might be corrupted. The expander terminates as described above.
- After an EOL code is detected, the expanded image line does not match the specified page width (EPWR). The expander terminates on the end of this line.

The ECD bit is set under the following circumstances:

- An extension code is detected for which the least significant three bits are not all "1"s. The corresponding extension code can be read from the Extension Bits field in the MSR in reversed order.

A corrupted code might be expanded into a runlength larger than the specified page width before it is actually recognized as a data error. However, under all circumstances, the CEP clips an expanded image line to the specified line length (EPWR).

The LPI bit is always set in conjunction with the DER or ECD bit set.

Recovering from data errors:

After a data error is detected, an error recovery procedure iterates the CEP through the next lines until a resynchronization on a 1-D coded line is possible. The CEP proceeds from there on without further assistance from the system. Though the CEP cannot recover from a data error situation without support from the host system, it provides many features which reduce the systems burden to a minimum.

After beginning the expansion of a new image line, the CEP always stores the line start of the image and the line start of the code in the line start registers (ESLSR, EDLSR). After a data error, this information is very important. With this information the system can determine how much of the image was expanded correctly and the position of the last correct line. The corrupted image line can be overwritten either by a line of zeros (white) or, better, by duplicating the last correct line. The CEP's transparent mode operation can be used to perform this task.

Then the next EOL beyond the data error must be found as a point from which the CEP can resume its operation. This can be done by performing a software reset for the expander, then setting the source attribute bit (SA) in the expander parameter register and specifying a starting address between the data error and the next EOL and then starting the CEP. Under this condition, the CEP thinks it is starting at the beginning of a coded page and starts searching for an EOL because CCITT defines a prefix EOL code for Group III data. After it recognizes the next EOL, it automatically proceeds to expand the code following the EOL.

After resuming from a data error in a 2D coded line, the expander will most likely, but not necessarily, decode another 2D line and get another data error because the reference line is not correct. The above described procedure must be repeated until a 1D coded line is reached. From there on, the CEP can proceed without further corruptions in the expanded image.

Since the expander prefetches up to three bytes, the next EOL might be part of the prefetched code. It might also be corrupted itself. Therefore, great care must be taken to calculate the correct address to resume from (The CEP Technical Manual gives detailed information on how to determine the correct address).

Negative Compression:

For some images (such as half-tones or low resolution raster-pictures), the compressed data representing a line may be longer than the original line of the image. The Am7971A checks for this condition after compressing a line providing bit 0 of the Compressor Parameter Register (CPR) has been cleared to "0" by the user. The Am7971 automatically checks for negative compression regardless of the status of this bit. When it occurs, the host processor is alerted by an interrupt. The CEP then terminates the compression and sets the negative compression bit in its status register. If bit 0 in the CPR is set to "1" by the user, the Am7971A does not check for negative compression and continues to process the document.

The host system can react in one of the following three possible ways:

1. Set all restart control bits in the restart control register (CRCR) to continue and set compressors to "GO". This action tolerates the negative compression and just continues operation.
2. Read the source and destination line start addresses (CSLSR, CDLSR) from the compressor registers and re-encode the same line in uncompressed mode by software. Then continue compression by reinitializing the destination current address register and following the same steps as above.
3. Tolerate negative compression as described under 1. If the whole coded page is negatively compressed, transmit or store the image uncompressed.

Illegal Command Error:

The CEP immediately terminates its operation and indicates an illegal command condition under the following circumstances:

Compressor:

- 2D operation is selected and wraparound register is non-zero.
- Wraparound and express mode are selected together
- Left and right margins are overlapping or larger than paper width
- Reserved Modes are selected in the CMCR
- Compressor Page Width has been selected as '0'

Expander:

- 2D expansion mode and wraparound mode have both been specified
- A non-zero granularity parameter and a non-zero wraparound have both been specified
- Reserved Modes are selected in the EMCR
- The expander page width has been selected as "0"

Only conditions which are vital for a consistent operation of the CEP's internal logic are checked. Other parameters (like frame width, etc) are not checked for consistency. The check is only performed on a 'GO' after a reset was performed. For reasons of efficiency, the check is suppressed on all subsequent resume operations.

Abort Condition:

Any attempt to write into the EMCR or CMCR while the CEP is busy will cause the current expander or compressor operation to be aborted. The expander/compressor busy and new operation attempted bit is set and an interrupt asserted. The CEP cannot resume from such a condition. However, an abort is useful to stop the CEP in a system emergency situation. The CEP ignores any attempt to write into expander registers other than EMCR while the expander is busy and compressor registers other than EMCR while the compressor is busy.

XII. CEP REGISTERS

TABLE 1. COMPRESSOR REGISTER ADDRESS ASSIGNMENTS

Abbr.	Name	Size (Bits)	Number of Bytes	Port Address(es)
MSR*	Master Status Register	8	1	FE
CMCR	Compressor Master Control Register	8	1	76
CPR	Compressor Parameter Register	8	1	74
CSR	Compressor Status Register	8	1	78
CER	Compressor Express Register	8	1	68
CRCR	Compressor Restart Control Register	8	1	48
CKPR	Compressor K Parameter Register	8	1	66
TFLR	Time Fill Register	8	1	44
CBOCR	Compressor Bit Offset Control Register	8	1	7A
CWR	Compressor Wraparound Register	16	2	50 (LSB)/52 (MSB)
LMGR	Left Margin Register	16	2	40 (LSB)/42 (MSB)
RMGR	Right Margin Register	16	2	60 (LSB)/62 (MSB)
TMGR	Top Margin Register	16	2	30 (LSB)/32 (MSB)
CFWR	Compressor Frame Width Register	16	2	54 (LSB)/56 (MSB)
CPWR	Compressor Page Width Register	16	2	70 (LSB)/72 (MSB)
CSAHR	Compressor Source Address Holding Register	24	3	3A (LSB)/3C/3E (MSB)
CSCAR	Compressor Source Current Address Register	24	3	0A (LSB)/0C/0E (MSB)
CDAHR	Compressor Destination Address Holding Register	24	3	4A (LSB)/4C/4E (MSB)
CDCAR	Compressor Destination Current Address Register	24	3	2A (LSB)/2C/2E (MSB)
CSCHR	Compressor Source Count Holding Register	24	3	14 (LSB)/16/18 (MSB)
CSWCR	Compressor Source Working Count Register	24	3	04 (LSB)/06/08 (MSB)
CDCHR	Compressor Destination Count Holding Register	24	3	34 (LSB)/36/38 (MSB)
CDWCR	Compressor Destination Working Count Register	24	3	24 (LSB)/26/28 (MSB)
CSLSR	Compressor Source Line Start Address Register	24	3	5A (LSB)/5C/5E (MSB)
CDLSR	Compressor Destination Line Start Address Register	24	3	6A (LSB)/6C/6E (MSB)
CRCAR	Compressor Reference Current Address Register	24	3	1A (LSB)/1C/1E (MSB)

*This register is common to both the compressor and the expander.

TABLE 2. EXPANDER REGISTER ADDRESS ASSIGNMENTS

Abbr.	Name	Size (Bits)	Number of Bytes	Port Address(es)
MSR*	Master Status Register	8	1	FE
EBOCR	Expander Bit Offset Control Register	8	1	FA
EMCR	Expander Master Control Register	8	1	F6
EPR	Expander Parameter Register	8	1	F4
ESR	Expander Status Register	8	1	F8
ERCR	Expander Restart Control Register	8	1	C8
EKPR	Expander K Parameter Register	8	1	E6
EWR	Expander Wraparound Register	16	2	D0 (LSB)/D2 (MSB)
EPWR	Expander Page Width Register	16	2	F0 (LSB)/F2 (MSB)
EFWR	Expander Frame Width Register	16	2	D4 (LSB)/D6 (MSB)
ESAHR	Expander Source Address Holding Register	24	3	BA (LSB)/BC/BE (MSB)
ESCAR	Expander Source Current Address Register	24	3	8A (LSB)/8C/8E (MSB)
EDAHR	Expander Destination Address Holding Register	24	3	CA (LSB)/CC/CE (MSB)
EDCAR	Expander Destination Current Address Register	24	3	AA (LSB)/AC/AE (MSB)
ESCHR	Expander Source Count Holding Register	24	3	94 (LSB)/96/98 (MSB)
ESWCR	Expander Source Working Count Register	24	3	84 (LSB)/66/88 (MSB)
EDCHR	Expander Destination Count Holding Register	24	3	B4 (LSB)/B6/B8 (MSB)
EDWCR	Expander Destination Working Count Register	24	3	A4 (LSB)/A6/A8 (MSB)
ESLSR	Expander Source Line Start Address Register	24	3	DA (LSB)/DC/DE (MSB)
EDLSR	Expander Destination Line Start Address Register	24	3	EA (LSB)/EC/EE (MSB)
ERCAR	Expander Reference Current Address Register	24	3	9A (LSB)/9C/9E(MSB)

Note: All register addresses are even, the bytes in a register are, therefore, not addressed with contiguous addresses.

*This register is common to both the compressor and expander.

TABLE 3. CEP REGISTERS BY ADDRESS (LEAST SIGNIFICANT DIGIT)

Most Significant Digit	0	2	4	6	8	A	C	E
0			CSWCR (L)	CSWCR (M)	CSWCR (H)	CSCAR (L)	CSCAR (M)	CSCAR (H)
1			CSCHR (L)	CSCHR (M)	CSCHR (H)	CRCAR (L)	CRCHR (M)	CRCAR (H)
2			CDWCR (L)	CDWCR (M)	CDWCR (H)	CDCAR (L)	CDCAR (M)	CSCAR (H)
3	TMGR (L)	TMGR (H)	CDCHR (L)	CDCHR (M)	CDCHR (H)	CSAHR (L)	CSAHR (M)	CSAHR (H)
4	LMGR (L)	LMGR (H)	TFLR		CRCR	CDAHR (L)	CDAHR (M)	CDAHR (H)
5	CWR (L)	CWR (H)	CFWR (L)	CFWR (H)		CSLSR (L)	CSLSR (M)	CSLSR (H)
6	RMGR (L)	RMGR (H)		CKPR	CER	CDLSR (L)	CDLSR (M)	CDLSR (H)
7	CPWR (L)	CPWR (H)	CPR	CMCR	CSR	CBOCR		
8			ESWCR (L)	ESWCR (M)	ESWCR (H)	ESCAR (L)	ESCAR (M)	ESCAR (H)
9			ESCHR (L)	ESCHR (M)	ESCHR (H)	ERCAR (L)	ERCAR (M)	ERCAR (H)
A			EDWCR (L)	EDWCR (M)	EDWCR (H)	EDCAR (L)	EDCAR (M)	EDCAR (H)
B			EDCHR (L)	EDCHR (M)	EDCHR (H)	ESAHR (L)	ESAHR (M)	ESAHR (H)
C					ERCR	EDAHR (L)	EDAHR (M)	EDAHR (H)
D	EWR (L)	EWR (H)	EFWR (L)	EFWR (H)		ESLSR (L)	ESLSR (M)	ESLSR (H)
E				EKPR		EDLSR (L)	EDLSR (M)	EDLSR (H)
F	EPWR (L)	EPWR (H)	EPR	EMCR	ESR	EBOCR		MSR

(L): Low Byte
(M): Middle Byte
(H): High Byte

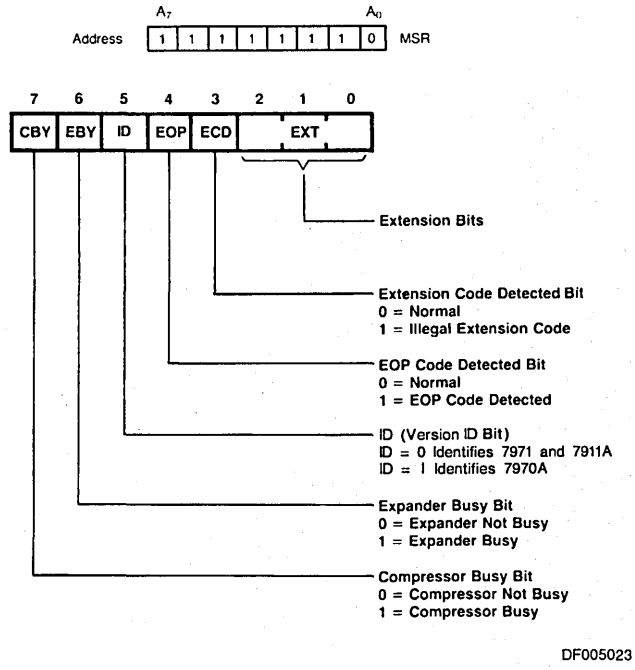


Figure 20. Master Status Register (MSR)

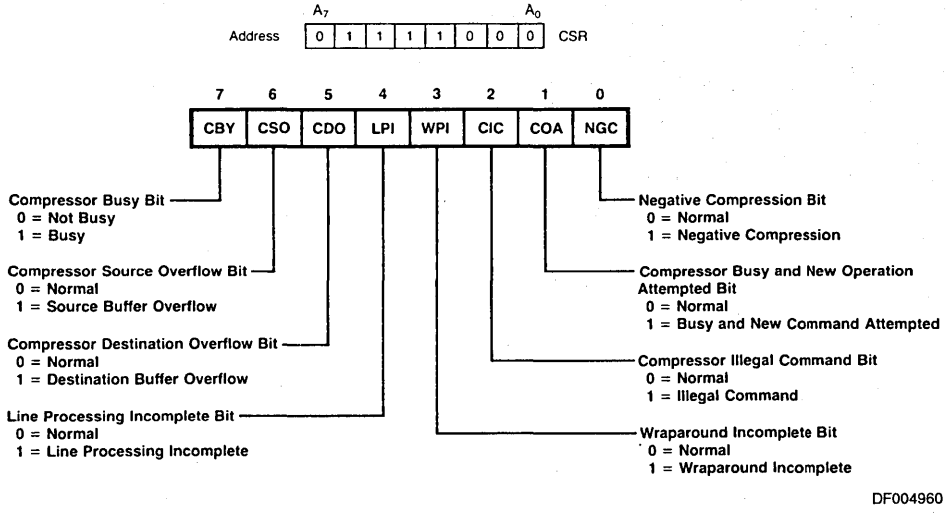


Figure 21. Compressor Status Register (CSR)

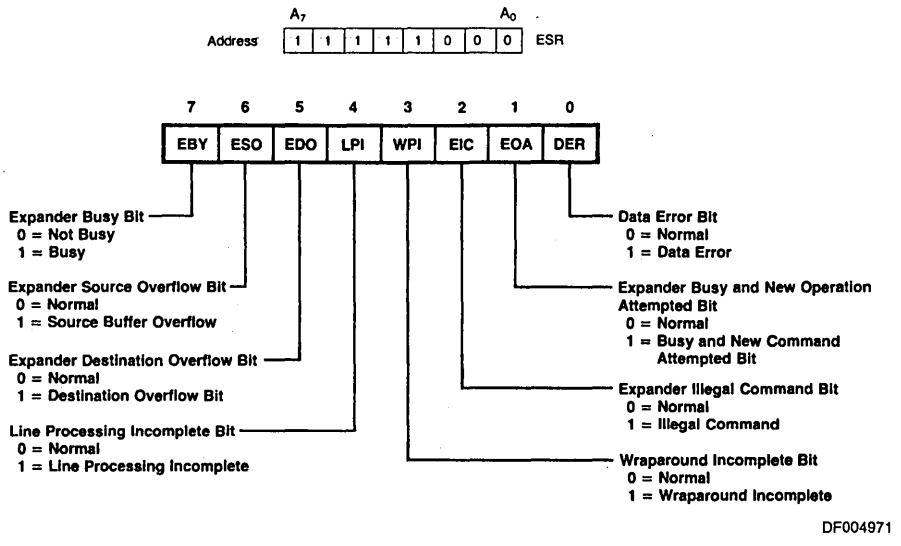


Figure 22. Expander Status Register (ESR)

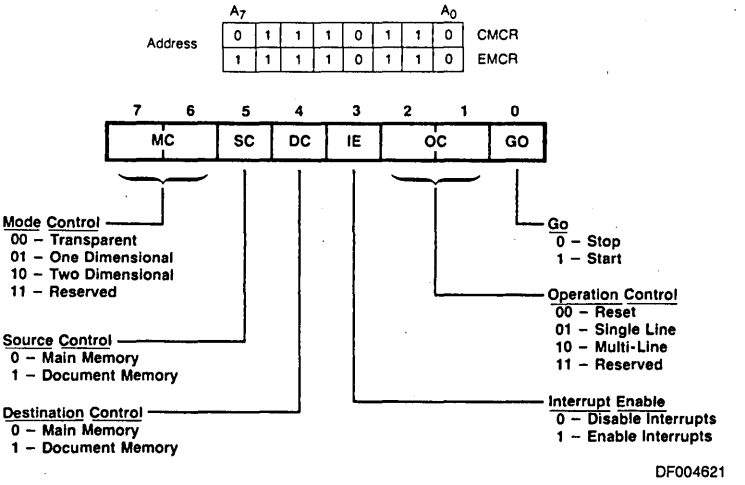
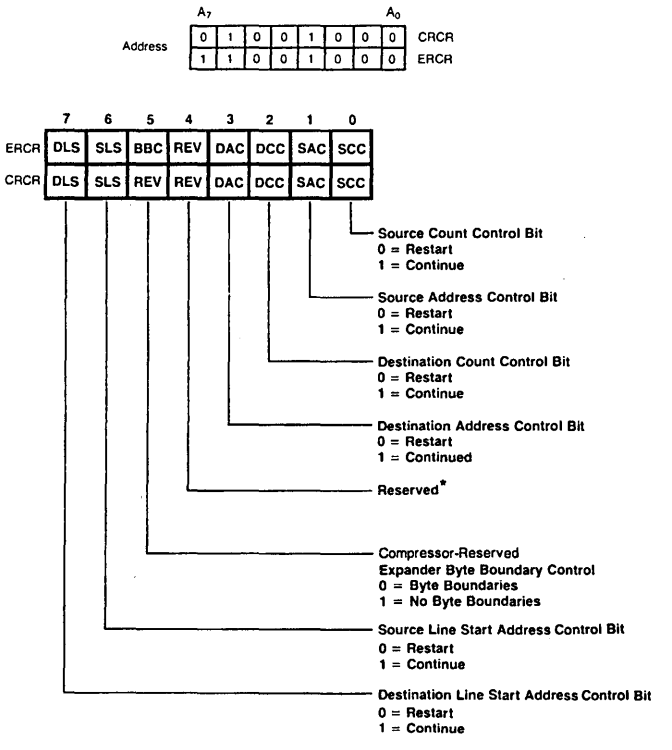


Figure 23. Master Control Register (CMCR, EMCR)



* All Reserved bits should be set to zero.

Figure 24. Restart Control Register (CRCR, ERCR)

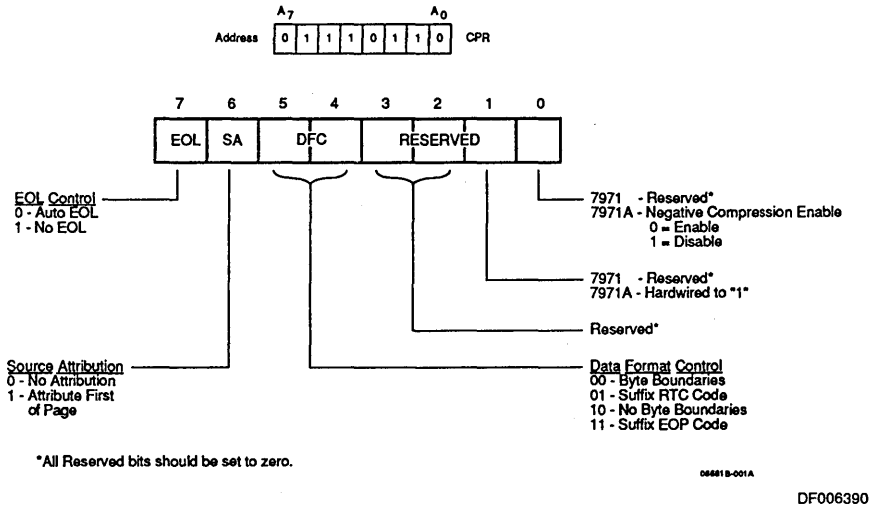
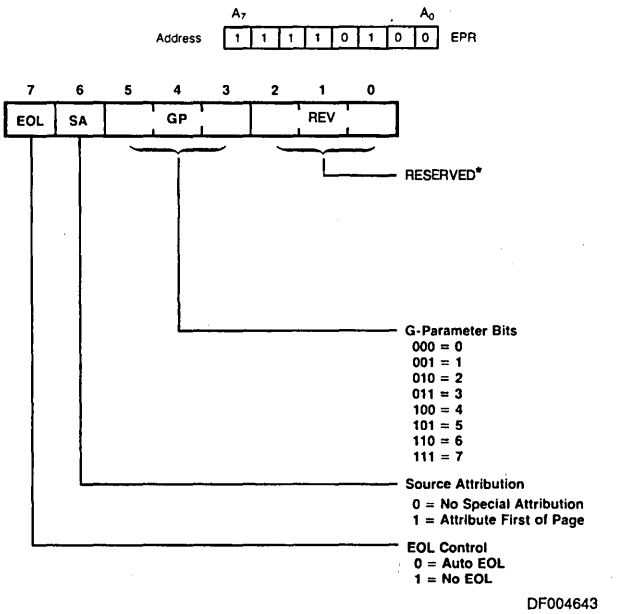


Figure 25. Compressor Parameter Register (CPR)



* All Reserved bits should be set to zero.

Figure 26. Expander Parameter Register (EPR)

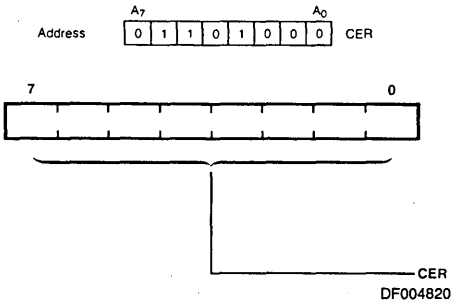


Figure 27. Compressor Express Register (CER)

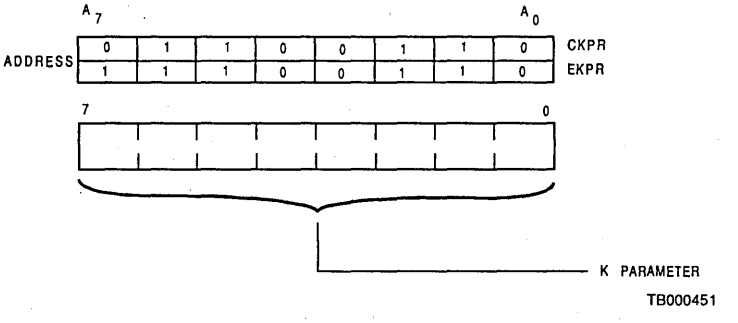


Figure 28. K Parameter Registers (CKPR, EKPR)

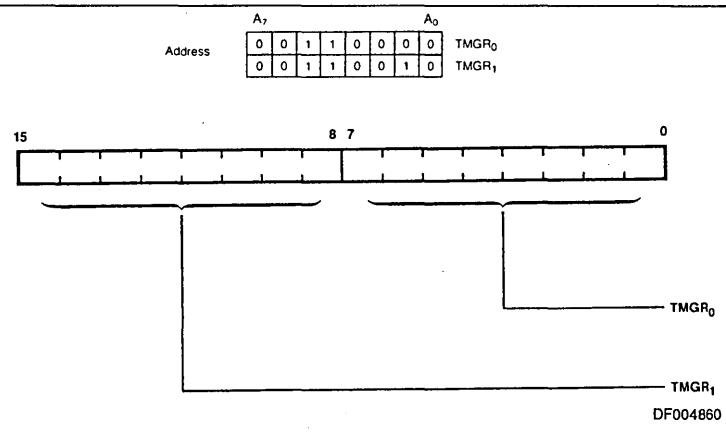


Figure 29. Top Margin Register (TMGR)

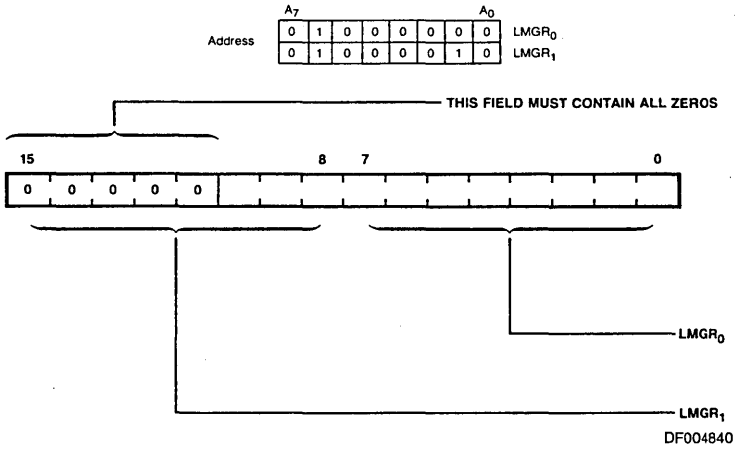


Figure 30. Left Margin Register (LMGR)

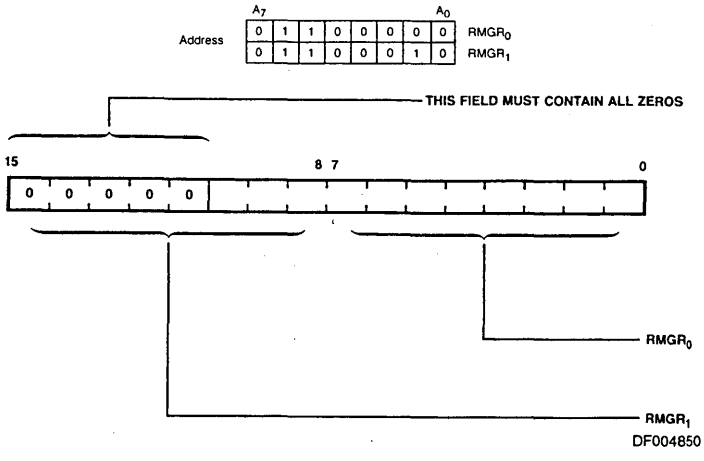


Figure 31. Right Margin Register (RMGR)

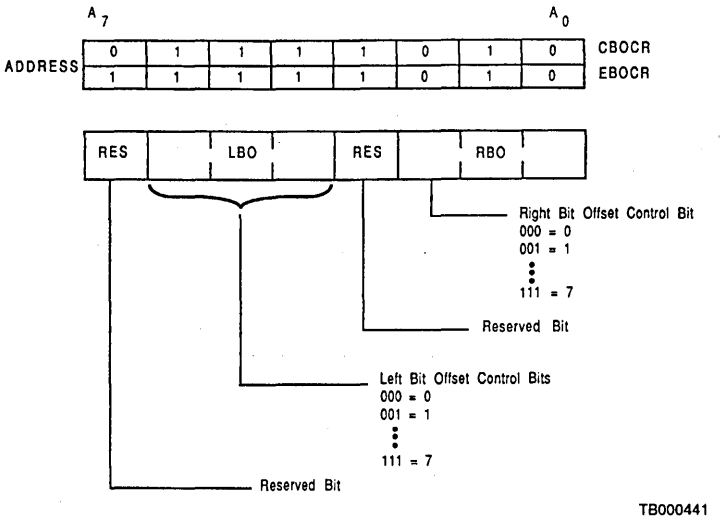


Figure 32. Bit Offset Control Register (CBOCR, EBOCR)

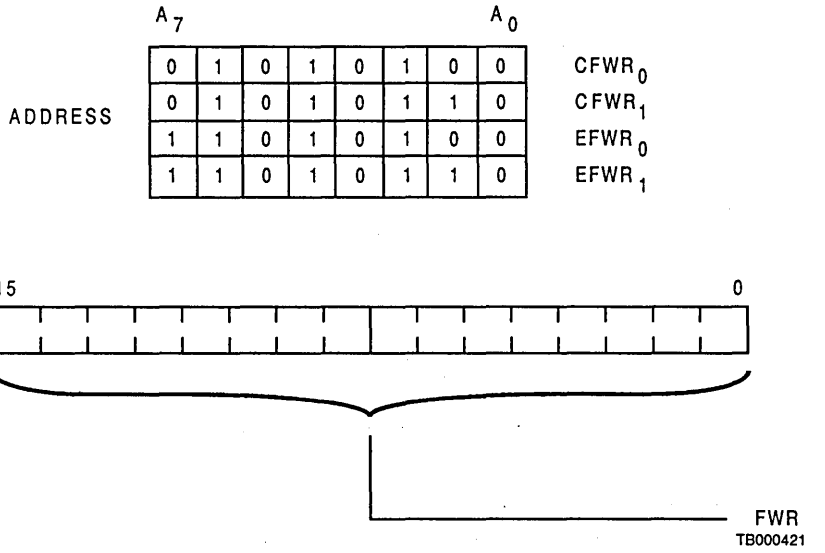


Figure 33. Frame Width Registers (CFWR, EFWR)

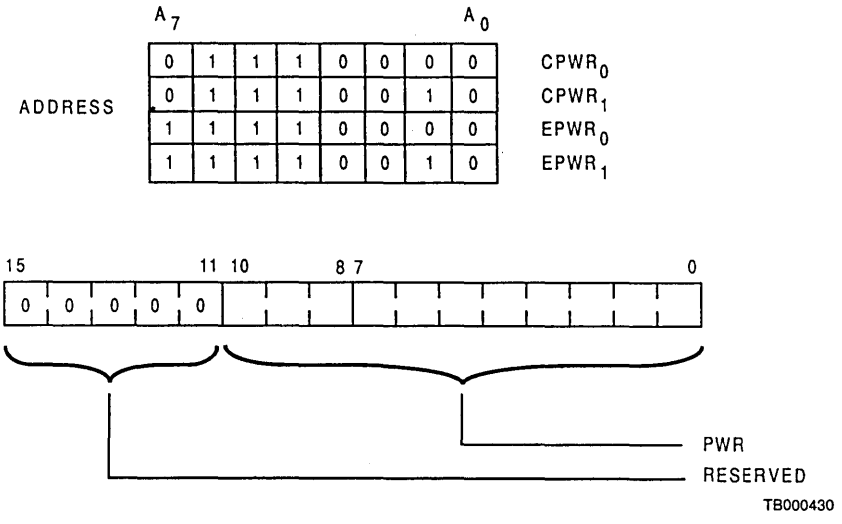


Figure 34. Page Width Registers (CPWR, EPWR)

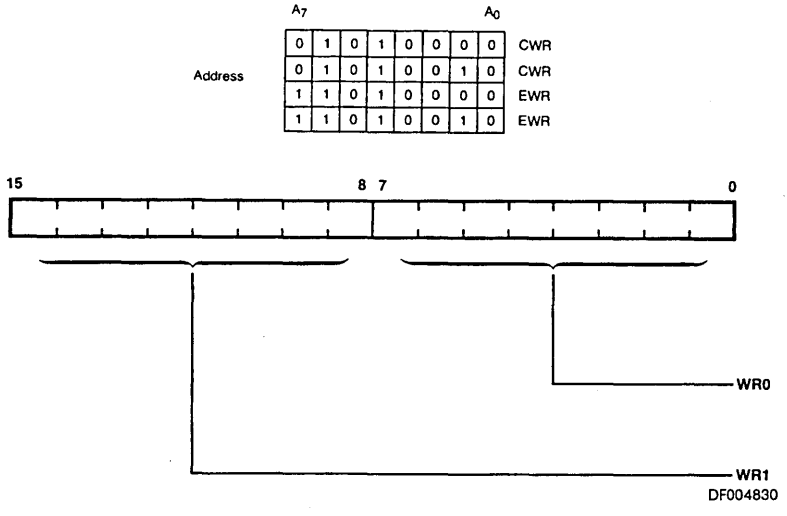


Figure 35. Wraparound Registers (CWR, EWR)

5

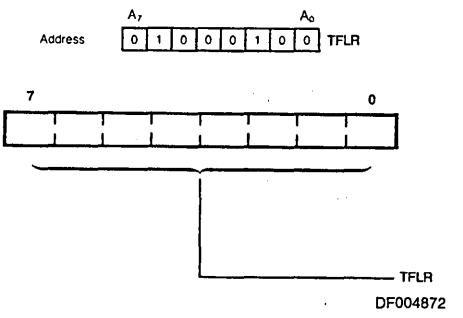


Figure 36. Time Fill Register (TFLR)

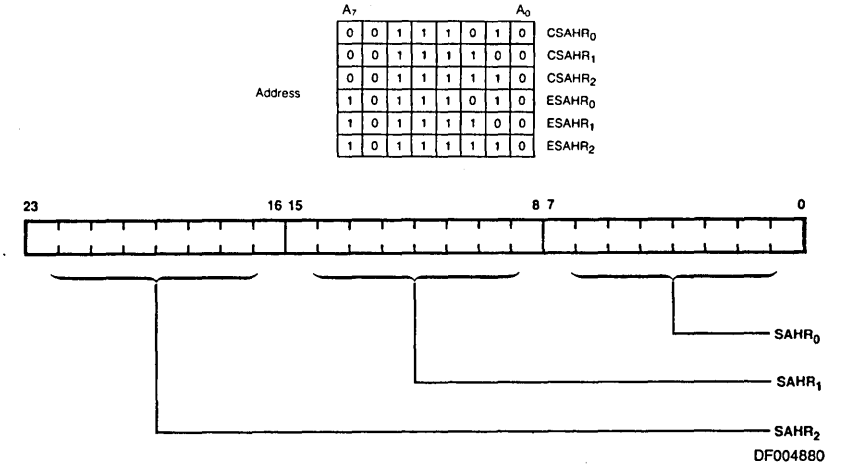


Figure 37. Source Address Holding Registers (CSAHR, ESAHR)

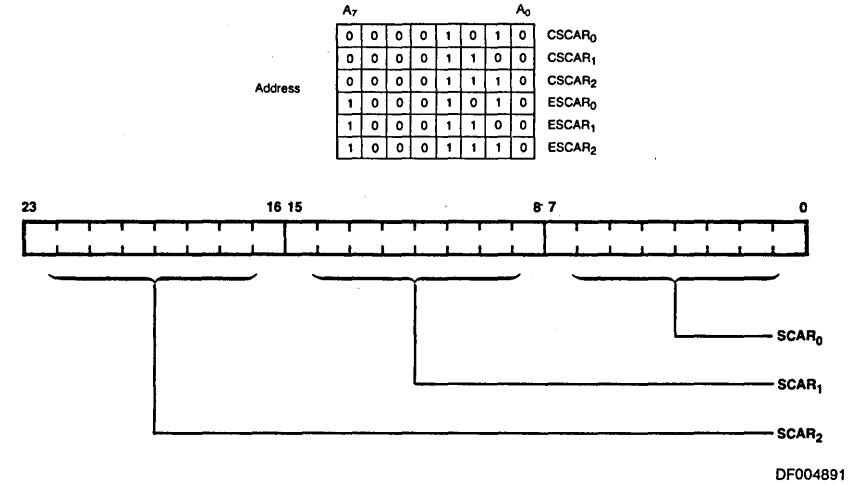


Figure 38. Source Current Address Registers (CSCAR, ESCAR)

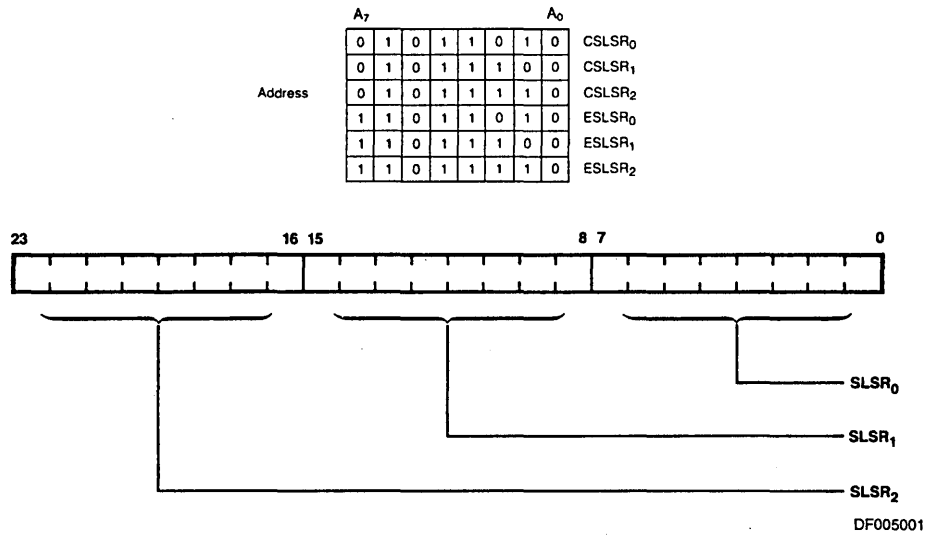


Figure 39. Source Line Start Address Registers (CSLSR, ESLSR)

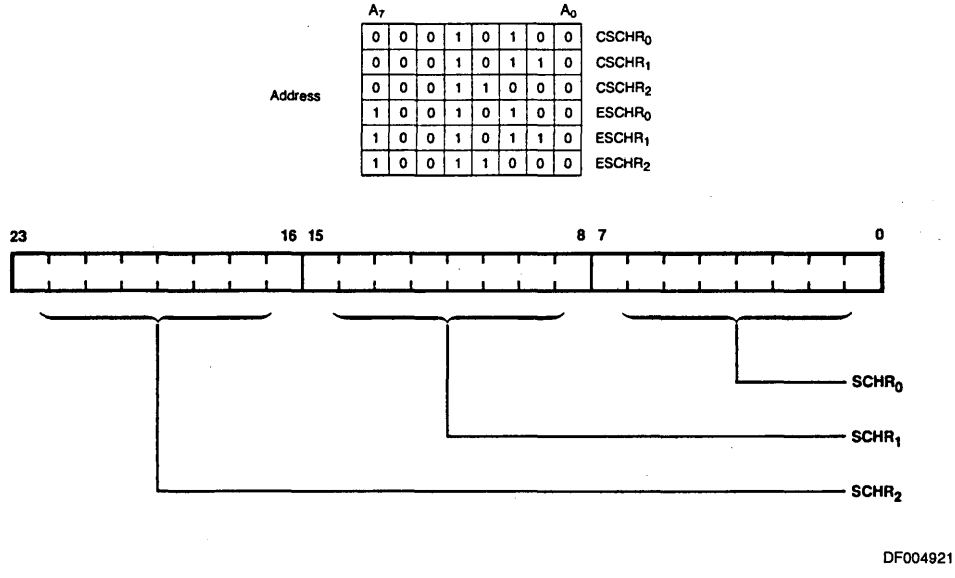


Figure 40. Source Count Holding Registers (CSCHR, ESCHR)

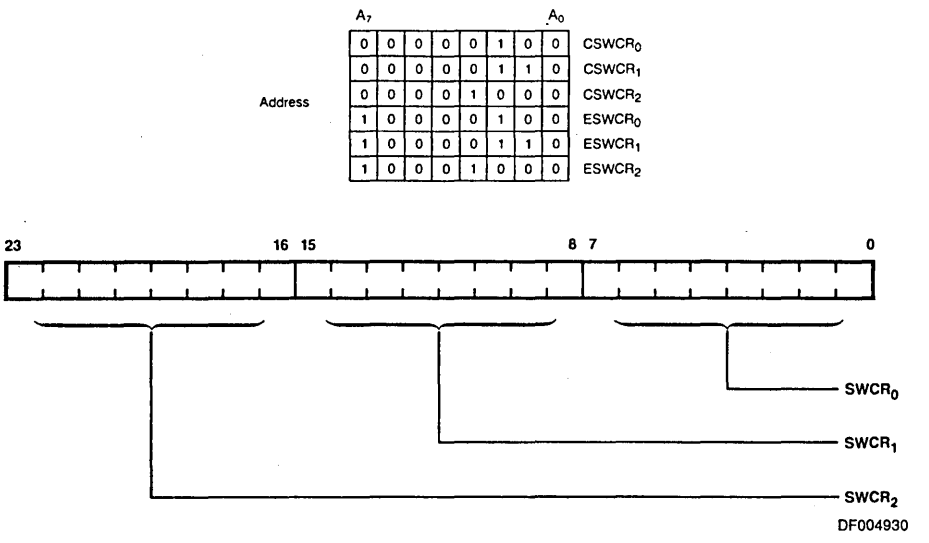


Figure 41. Source Working Count Registers (CSWCR, ESWCR)

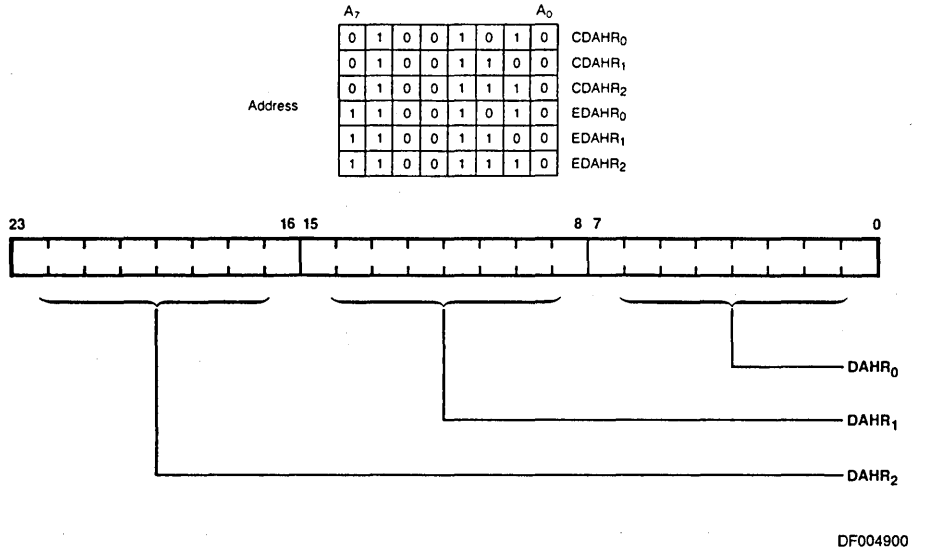


Figure 42. Destination Address Holding Registers (CDAHR, EDAHR)

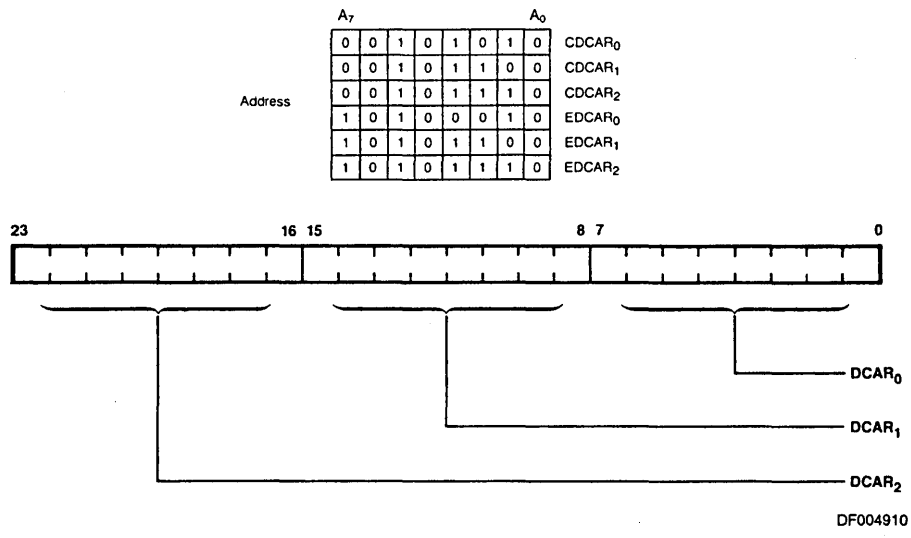


Figure 43. Destination Current Address Registers (CDCAR, EDCAR)

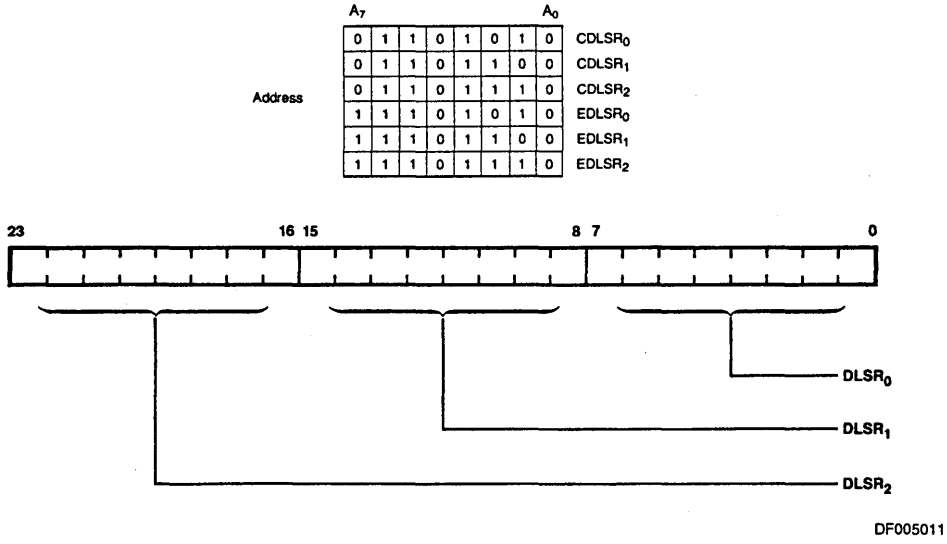


Figure 44. Destination Line Start Address Registers (CDLSR, EDLSR)

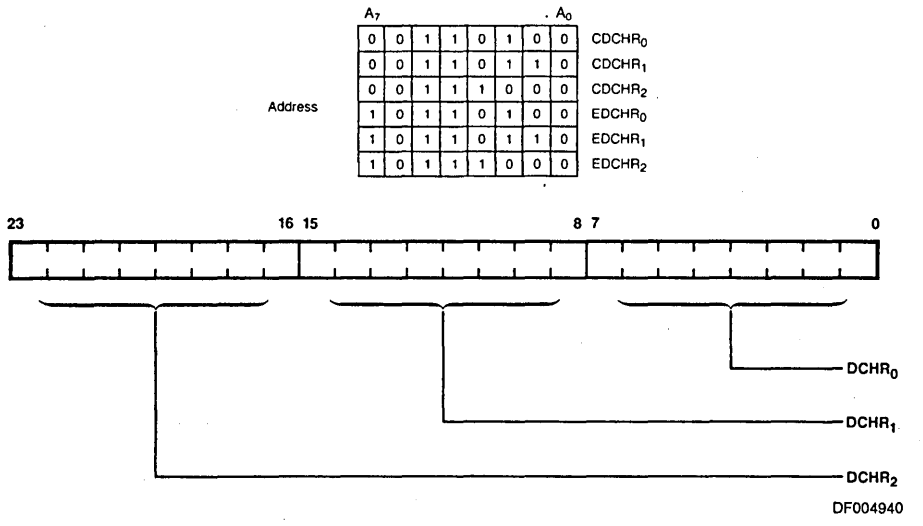


Figure 45. Destination Count Holding Registers (CDCHR, EDCHR)

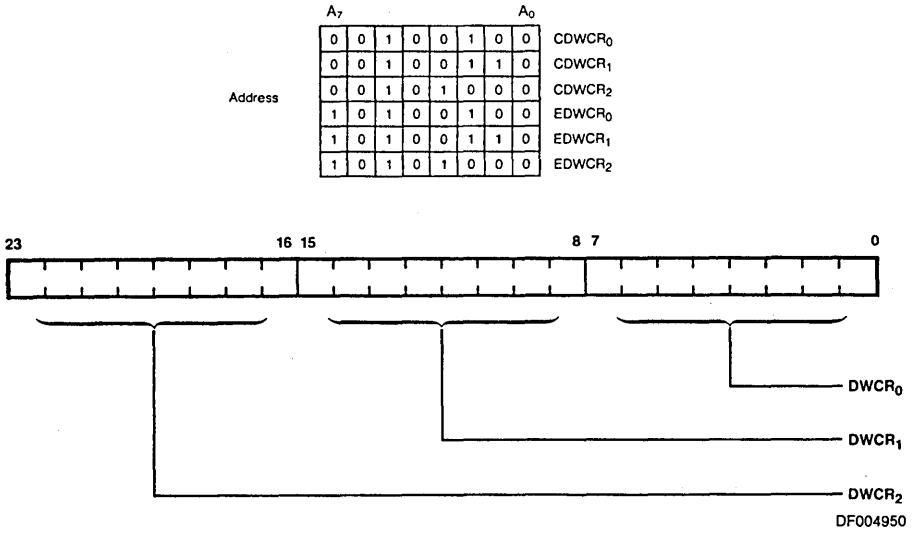


Figure 46. Destination Working Count Registers (CDWCR, EDWCR)

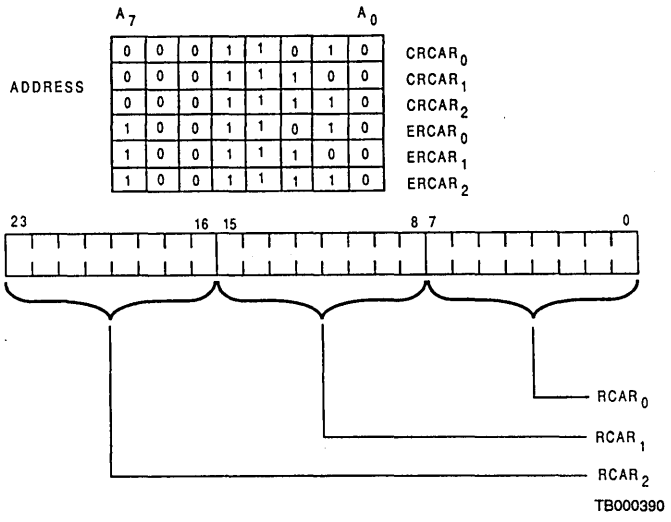
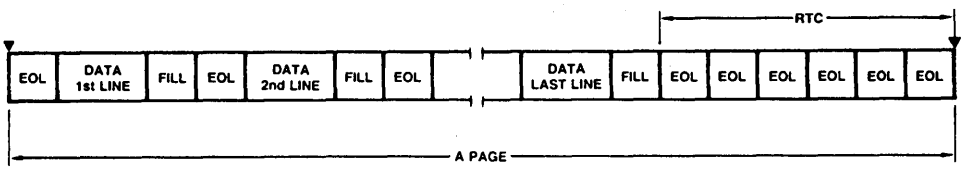


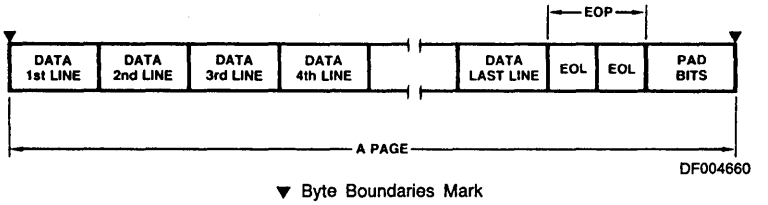
Figure 47. Reference Current Address Registers (CRCAR, ERCAR)

XIII. Am7971A CEP DATA FORMATS



▼ Byte Boundaries Mark

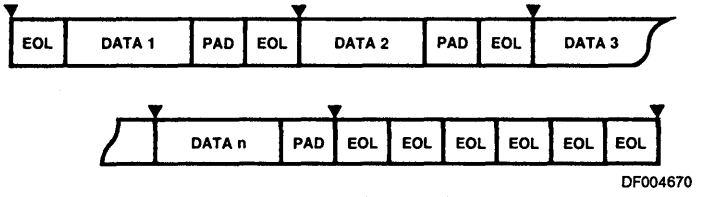
Figure 48. G-3 Compressed Data Format



▼ Byte Boundaries Mark

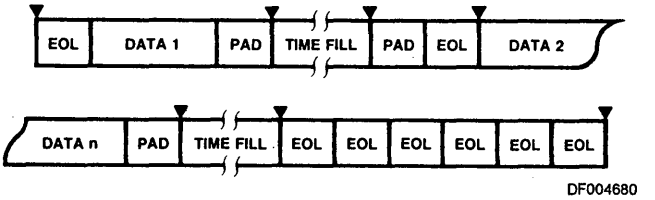
Figure 49. G-4 Compressed Data Format

Compressed Data Format of the 1-D Mode (Figures 50 through 55)



▼ Byte Boundaries Mark
PAD: Consecutive any numbers of 0's (if any)

Figure 50. Byte Boundary Conditioned with Auto EOL and No Time Fill



▼ Byte Boundaries Mark
PAD: 1 to 7 of 0's (if any)

Figure 51. Byte Boundary Conditioned with Auto EOL and Time Fill

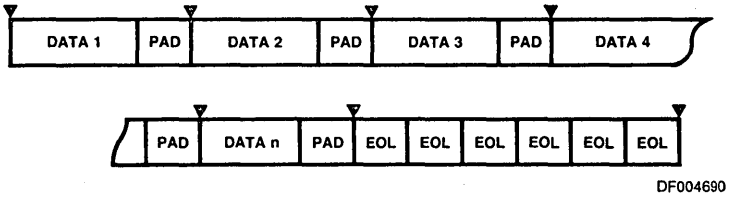
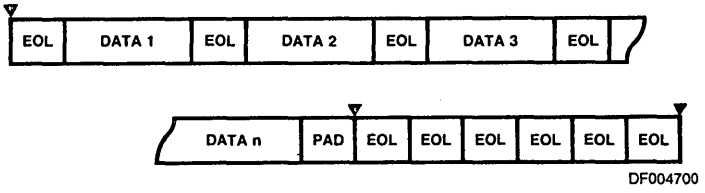


Figure 52. Byte Boundary Conditioned without EOL and Time Fill



▼ Byte Boundaries Mark
PAD: 1 to 7 of 0's (if any)

Figure 53. No Byte Boundary Conditioned with Auto EOL and No Time Fill

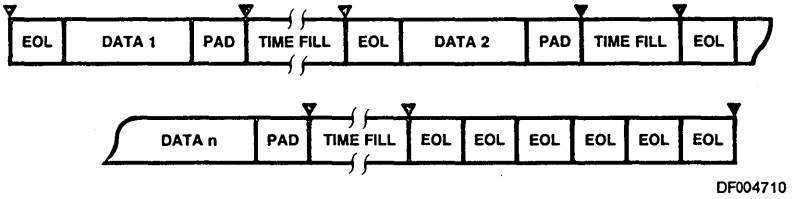
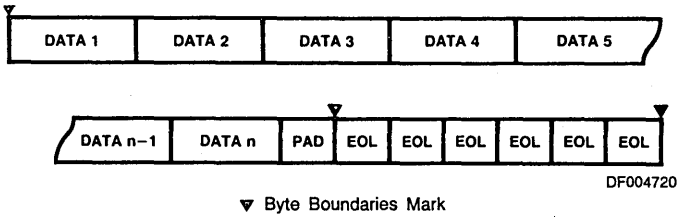


Figure 54. Byte Boundary Conditioned with Auto EOL and Time Fill



▼ Byte Boundaries Mark
PAD: 1 to 7 of 0's (if any)

Figure 55. No Byte Boundary Conditioned without EOL and Time Fill

Compressed Data Format of the 2-D Mode (Figures 56 through 64)

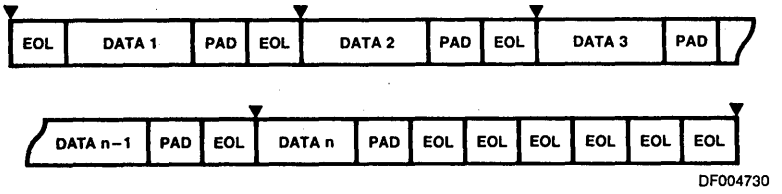


Figure 56. Byte Boundary Conditioned with Auto EOL and No Time Fill

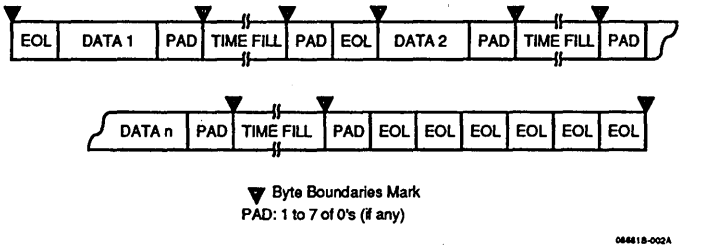


Figure 57. Byte Boundary Conditioned with Auto EOL and Time Fill

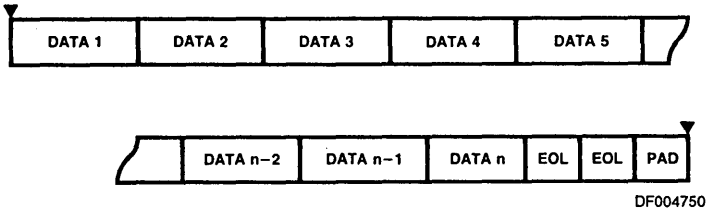
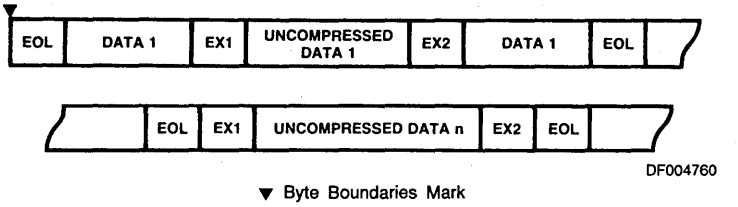


Figure 58. No Byte Boundary Conditioned without EOL and Time Fill (G-4)



▼ Byte Boundaries Mark
PAD: Consecutive any numbers of 0's (if any)
EX1: Extension code (entry code)
EX2: Extension code (exit code)

Figure 59. Uncompressed Data Format

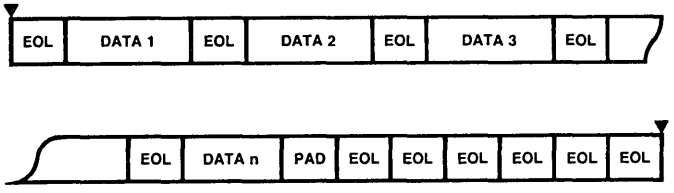
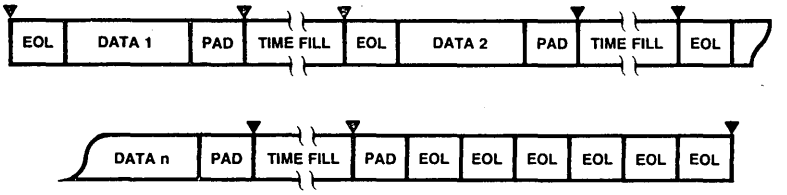


Figure 60. No Byte Boundary Conditioned with Auto EOL and No Time Fill



▼ Byte Boundaries Mark
PAD: 1 to 7 0's (if any)

Figure 61. No Byte Boundary Conditioned with Auto EOL and Time Fill

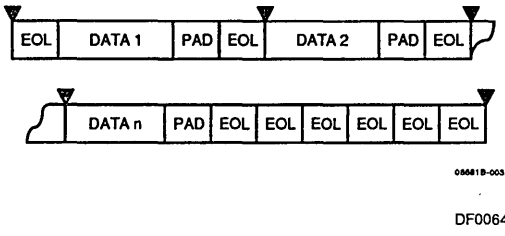


Figure 62. Byte Boundary Conditioned with Auto EOL and No Time Fill

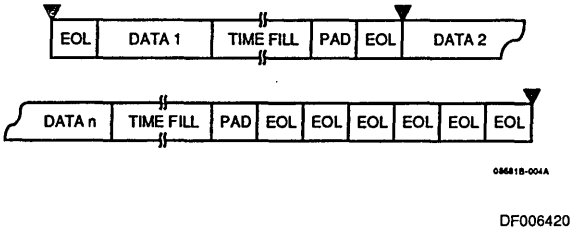
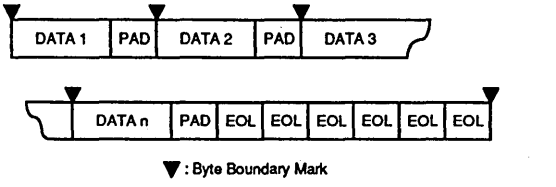


Figure 63. Byte Boundary Conditioned with Auto EOL and Time Fill



DF006430

▼: Byte boundary mark

Figure 64. Byte Boundary Conditioned without EOL and Time Fill

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Temperature Ambient Under Bias — T_C ..... 0 to +70°C
Supply Voltage to Ground
Potential Continuous ..... −0.5 to +7.0 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5.0 V ±5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

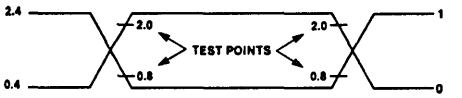
DC CHARACTERISTICS over operating range

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IL}	Input LOW Voltage		−0.5	0.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + 0.5	V
V_{OL}	Output LOW Voltage	I_{OL} = 3.2 mA		0.4	V
V_{OH}	Output HIGH Voltage	I_{OH} = 400 µA	2.4		V
I_{CC}	Power Supply Current (Note 1)	T_A = 0°C		600	mA
I_{LL}	Input Leakage Current	0 V ≤ V_{IN} ≤ V_{CC}		±10	µA
I_{LO}	Output Leakage Current	0.45 V ≤ V_{OUT} ≤ V_{CC}		±10	µA
V_{CL}	Clock Input LOW Voltage		−0.5	+0.8	V

Capacity of all inputs and outputs: 10 pF (Fc = 1 MHz). This parameter is not tested in every device.

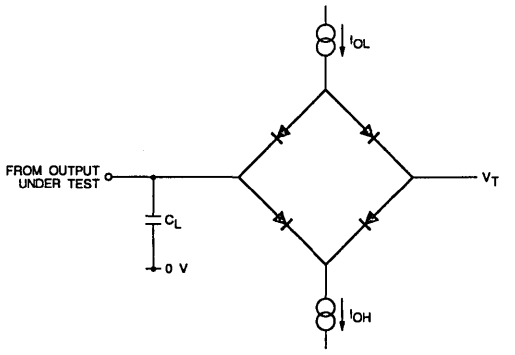
Notes: 1. I_{CC} Typical (T_A = 25°C) = 500 mA
I_{CC} Typical (T_A = 70°C) = 450 mA

SWITCHING TEST WAVEFORM



WF009542

SWITCHING TEST CIRCUIT



TC003862

$$V_T = \frac{V_{OL} + V_{OH}}{2}$$

SWITCHING CHARACTERISTICS over operating range

Timing Requirements

# Parameter	Description	Test Conditions	3 MHz		5 MHz		8 MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1 TCLCL	CLK Cycle Period	From 0.8 V to 0.8 V	330	1000	200	1000	125	1000	ns
2 TCHCL	CLK HIGH Time	From 2.0 V to 2.0 V	130		85		55		ns
3 TCLCH	CLK LOW Time	From 0.8 V to 0.8 V	130		85		55		ns
4 TWHRL	READY Hold Time after WRITE (Note 5)	From 0.8 V to 2.0 V		65		65		60	ns
5 TRDHRL	READY Hold Time after READ (Note 5)	From 0.8 V to 2.0 V		65		65		60	ns
6 TRHVH	Power Supply HIGH to RESET LOW Time		4TCLCL		4TCLCL		4TCLCL		ns
7 TRHRL	RESET HIGH TIME		4TCLCL		4TCLCL		4TCLCL		ns
8 TRLSL	RESET LOW to First $\overline{CS}$		2TCLCL		2TCLCL		2TCLCL		ns
9 THAHCH	HLDA RE Set-up Time (Note 1)		50		30		25		ns
10 THALCH	HLDA FE Set-up Time (Note 1)		50		30		25		ns
11 TAVCV	Address Valid to Control Active		20		20		20		ns
12 TSLRDL	$\overline{CS}$ LOW to $\overline{RD}$ LOW		(Note 6)		(Note 6)		(Note 6)		ns
13 TRDHSH	$\overline{RD}$ HIGH to Address Change		20		20		20		ns
14 TSLWRL	$\overline{CS}$ LOW to $\overline{WR}$ LOW		(Note 6)		(Note 6)		(Note 6)		ns
15 TDVRYH	Data Valid to READY RE (Note 2)		30		30		25		ns
15a TDVWH	Data Valid to $\overline{WR}$ HIGH		30		30		25		ns
16 TWRHDV	DATA Hold Time		20		20		20		ns
17 TWRHSH	$\overline{WR}$ HIGH to Address Change		20		20		20		ns
18 TRYLCH	READY FE Set-up Time		30		20		20		ns
19 TCHDXY	READY Hold Time		30		20		20		ns
20 TRYHCH	READY Active Set-Up Time		30		20		20		ns
21 TDVCH	DATA IN Set-Up Time		45		35		30		ns
22 TRDHDX	READ HIGH to Data not Valid		0		0		0		ns

Note: Switching characteristics are targetted numbers and are subject to change without notice.

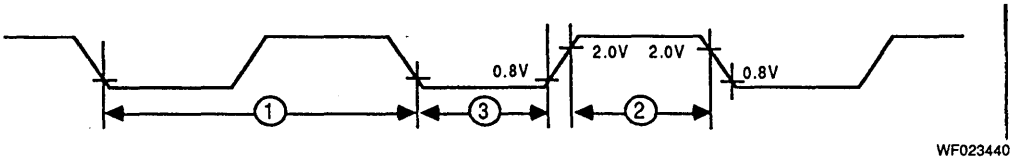
See notes following table on page 47.

SWITCHING CHARACTERISTICS over operating range
Timing Responses

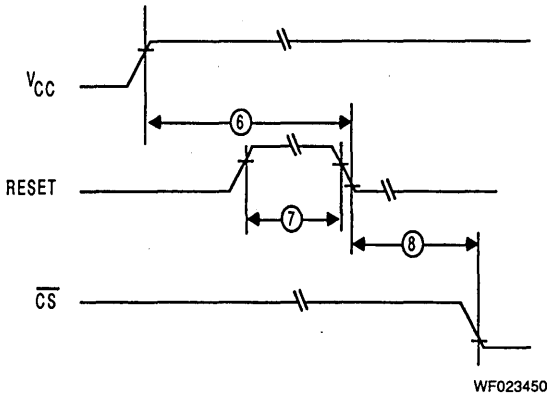
# Parameter	Description	Test Conditions	3 MHz Min.	3 MHz Max.	5 MHz Min.	5 MHz Max.	8 MHz Min.	8 MHz Max.	Unit
24 TCLHRH	HRQ Active Delay	CL = 20 – 100 PF for all Am7971A Outputs (In addition to Am7971A selfload)		120		80		50	ns
25 TCLHRL	HRQ Inactive Delay			120		85		50	ns
26 TSLRYL	READY Active Delay			80		80		50	ns
27 TCLAV	Address Valid Delay			110		80		55	ns
28 TCLLH	ALE/DALE Active Delay			80		65		50	ns
29 TLHLL	ALE/DALE Width		TCLCH – 40		TCLCH – 20		TCLCH – 20		ns
30 TAVAL	Address Valid to ALE LOW		65		55		25		ns
31 TCHLL	ALE/DALE Inactive Delay			80		65		50	ns
32 TLLAX	Address Hold Time to ALE/DALE Inactive		70		50		40		ns
33 TCLAZ	Address Float Delay From Clock FE			70		50		40	ns
34 TCLAX	Address Hold Time		0		0		0		ns
35 TAZRL	Address Float to $\overline{RD}/\overline{DRD}$ Active		(Note 3)		(Note 3)		(Note 3)		ns
36 TCHRL	$\overline{RD}/\overline{DRD}$ Active Delay from Clock RE			60		50		40	ns
37 TRLRH	$\overline{RD}/\overline{DRD}$ Width		TCLCL – 50		TCLCL – 40		TCLCL – 30		ns
38 TCHRH	$\overline{RD}/\overline{DRD}$ Inactive Delay from Clock RE			60		50		40	ns
39 TRHAV	$\overline{RD}/\overline{DRD}$ Inactive to Next Address Active		TCHCL		TCHCL		TCHCL		ns
40 TCLDX	DATA Hold Time		0		0		0		ns
41									ns
42 TCLDV	DATA Valid Delay From Clock FE			90		70		70	ns
43 TCHWL	$\overline{WR}/\overline{DWR}$ Active Delay from Clock RE			60		50		40	ns
44 TWLWH	$\overline{WR}/\overline{DWR}$ Width		TCLCL – 50		TCLCL – 40		TCLCL – 30		ns
45 TCHWH	$\overline{WR}/\overline{DWR}$ Inactive Delay from Clock RE			60		50		40	ns
46 TWHDX	DATA Hold Time After $\overline{WR}/\overline{DWR}$		90		60		50		ns
47 TCADT3 a b	Control Active Delay from Float Control Inactive Delay to Float			65		65		55	ns
48 TCLRH	Clock to RFA HIGH	From 0.8 V to 2.0 V		75		65		55	ns
49 TCLRL	Clock to RFA LOW	From 2.0 V to 0.8 V		75		65		55	ns
50 TRYW	READY Width (Note 4)		2TCLCL – 75		2TCLCL – 75		2TCLCL – 55		ns
51 TCSH	$\overline{CS}$ Hold Time		0		0		0		ns
52 TRDHDV	DATA FLOAT Time			50		50		50	ns
53 THRLHR	HREQ LOW To HRQ HIGH		2TCLCL		2TCLCL		2TCLCL		ns

Switching characteristics are targetted numbers and are subject to change without notice.

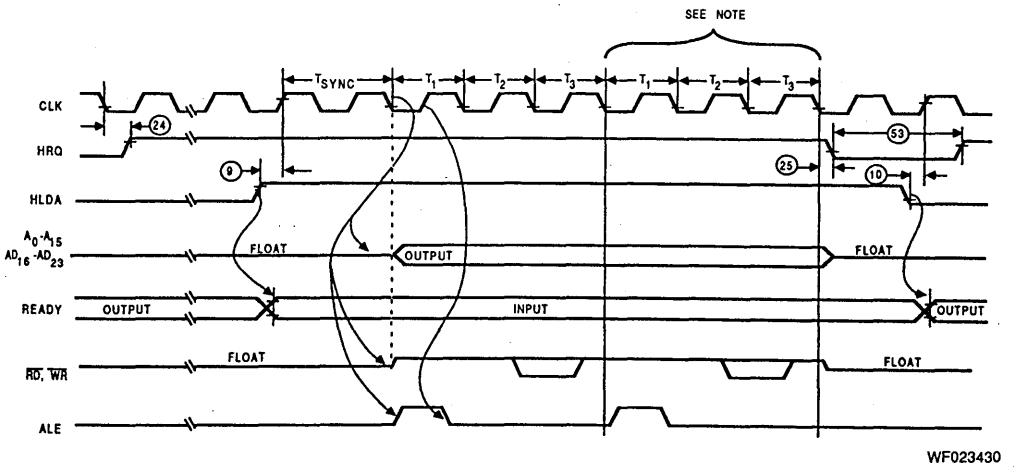
Note: 1. HLDA is an asynchronous input. If THAHCH or THALCH are violated that only means that HLDA might be recognized one clock cycle later.
2. The rising edge of READY is synchronous with the falling edge of CLK. The delay from CLK is 65 ns max.
3. Min. can be computed from ③ – ㊸ max.
4. Maximum is :
   60 x TCLCL when CEP Busy (not tested)
   16 x TCLCL when CEP Not Busy (not tested)
5. If $\overline{CS}$ is HIGH, READY does not return to LOW after the slave access is completed.
6. The slave access is started when both $\overline{RD}/\overline{WR}$ and $\overline{CS}$ are asserted LOW. Thus, $\overline{CS}$ could actually be asserted later than $\overline{RD}$ or $\overline{WR}$. Under this condition, parameter ⑪ applies to $\overline{CS}$.



Timing Diagram 1. Clock Timing

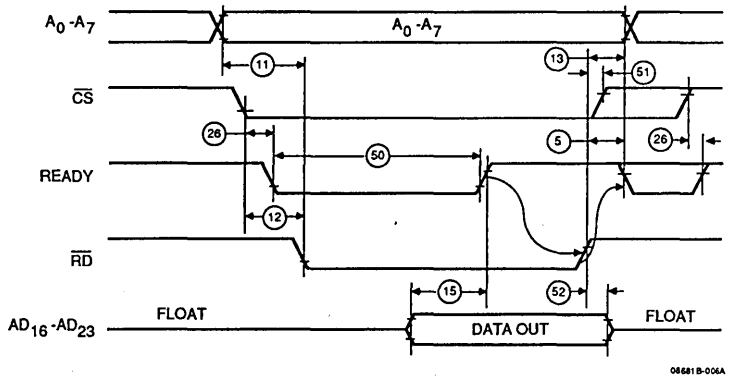


Timing Diagram 2. RESET Timing



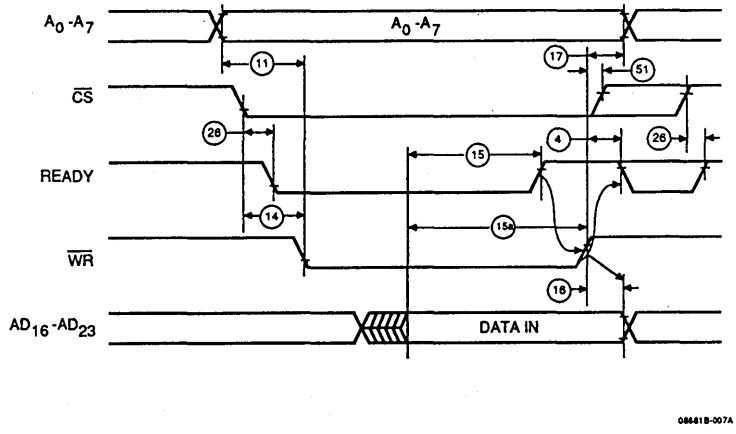
Timing Diagram 3. Bus Exchange Timing (System Interface)

Note: The second transfer cycle occurs only on Am7971A during expansion with Bit offset ≠ 0 with the destination buffer located on the system Bus at the end and beginning of an image line (READ/MODIFY/WRITE).



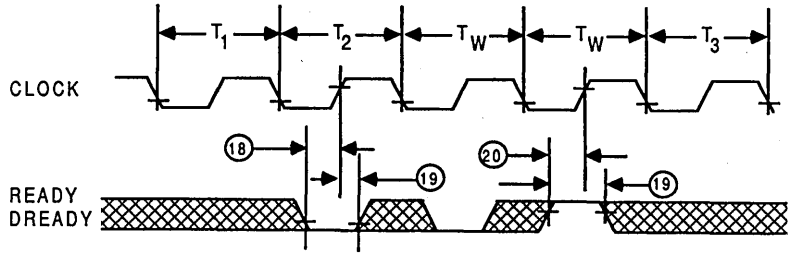
WF026440

Timing Diagram 4. CPU Program Read Timing



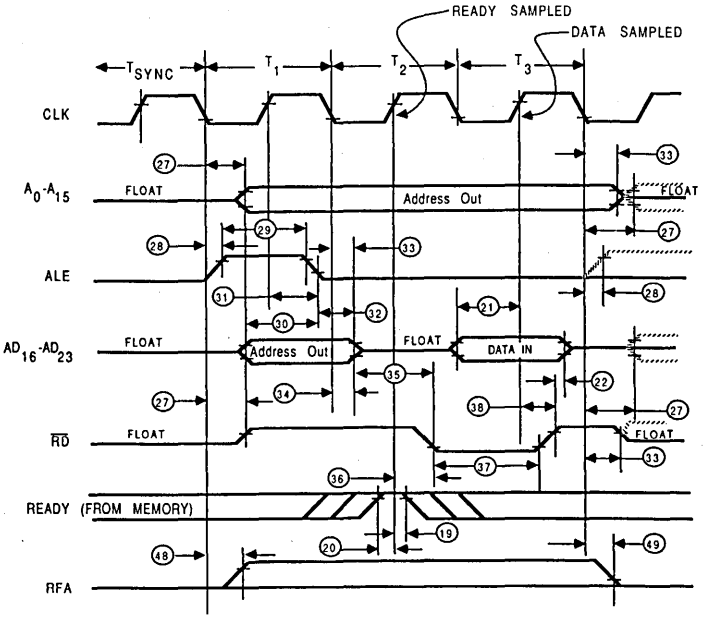
WF026450

Timing Diagram 5. CPU Program Write Timing



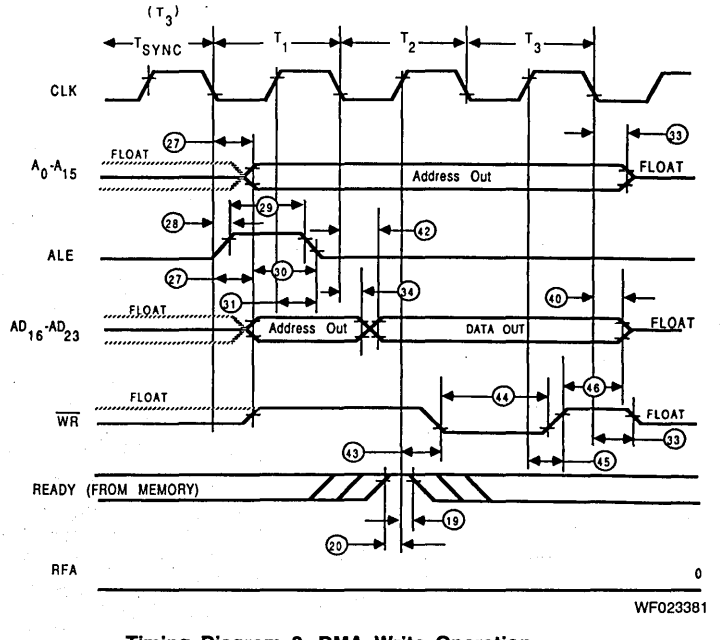
WF023400

Timing Diagram 6. READY, DREADY Input Timing Master Mode, System and Document Interface



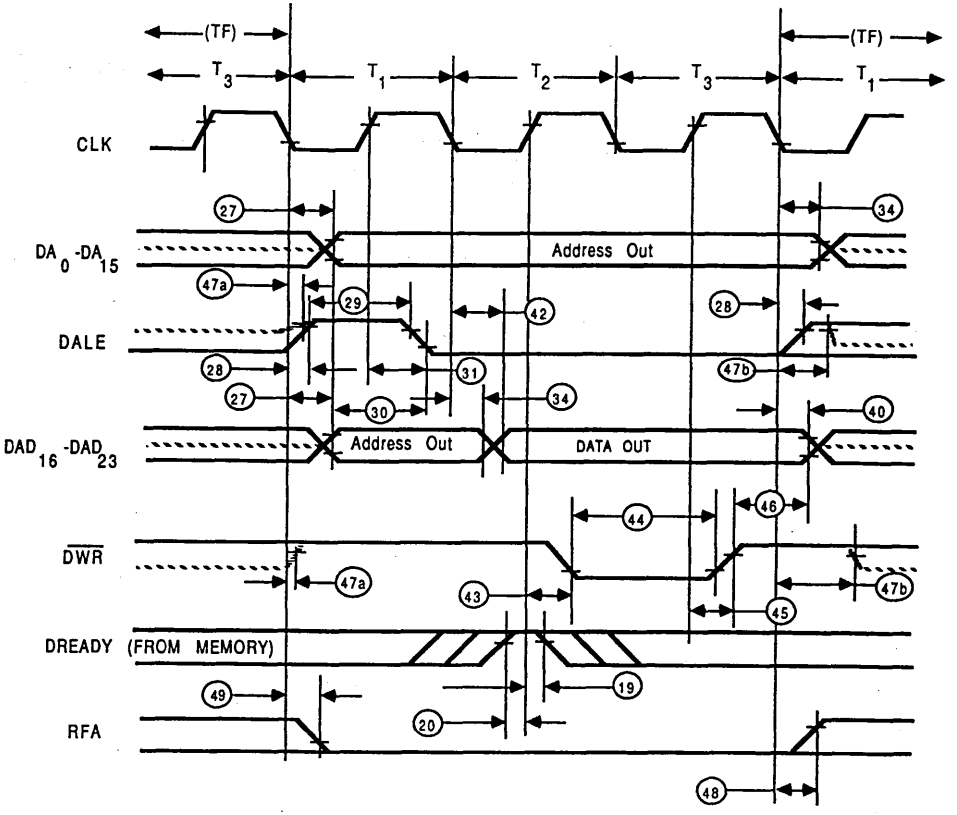
Timing Diagram 7. DMA Read Operation

Indicates additional access during Read-Modify-Write cycle if EBOCR ≠ 0



Timing Diagram 8. DMA Write Operation

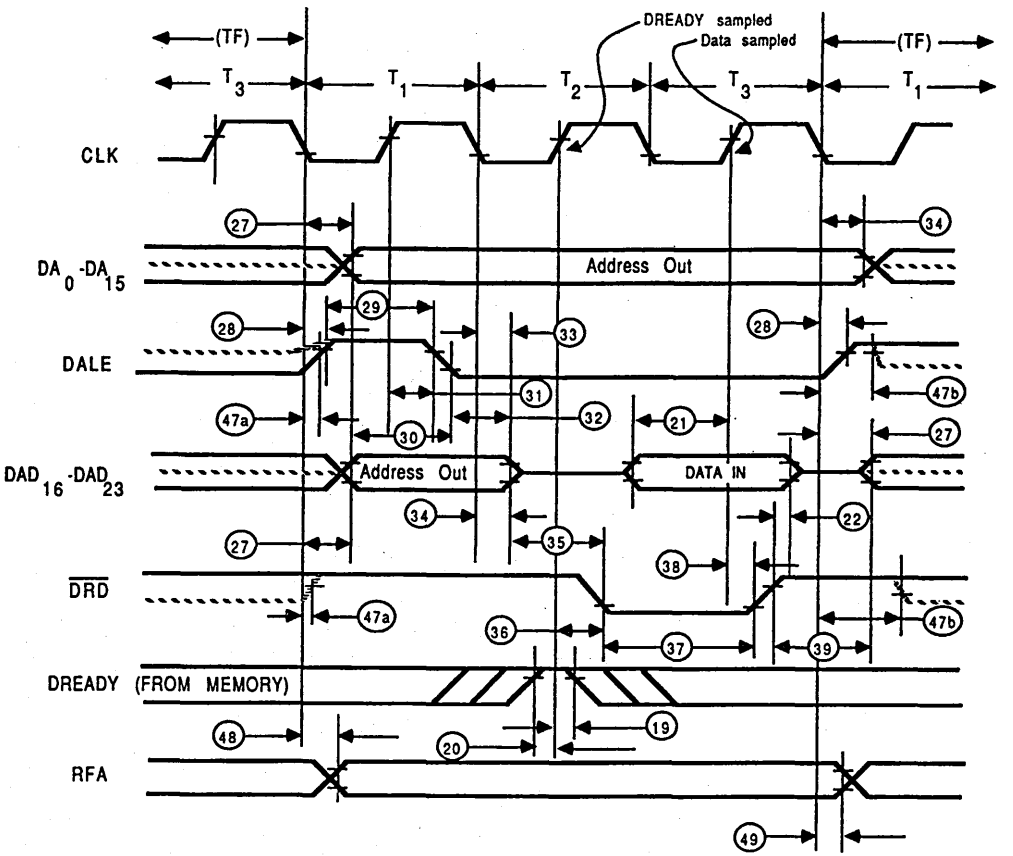
Indicates additional access during Read-Modify-Write cycle if EBOCR ≠ 0



WF023371

Dashed lines show three-stated condition during idle state when Am7971A is not bus master
TF = Float or idle state

Timing Diagram 9. Document Store Bus Write Operation



Dashed lines show three-stated condition during idle state when Am7971A is not bus master
TF = Float or idle state

Timing Diagram 10. Document Store Bus Read Operation

Am7971A CEP ACRONYM LIST

Acronym	Name	Register
ALE	Address Latch Enable	
BBC	Byte Boundary Control bit	ERCR
CBOCR	Compressor Bit Offset Control Register	
CCITT	Int'l Telegraph and Telephone Consultative Committee	
CDAHR	Compressor Destination Address Holding Register	
CDCAR	Compressor Destination Current Address Register	
CDCHR	Compressor Destination Count Holding Register	
CDLSR	Compressor Destination Line Start Address Register	
CDO	Compressor Destination Overflow bit	CSR
CDWCR	Compressor Destination Working Count Register	
CER	Compressor Express Register	
CFWR	Compressor Frame Width Register	
CIC	Compressor Illegal Command bit	CSR
CIE	Compressor Interrupt Enable bit	CMCR
CKPR	Compressor K Parameter Register	
CMCR	Compressor Master Control Register	
COA	Compressor Busy and New Operation Attempted bit	CSR
CPR	Compressor Parameter Register	
CRCAR	Compressor Reference Current Address Register	
CRCR	Compressor Restart Control Register	
CS	Chip Select	
CSAHR	Compressor Source Address Holding Register	
CSCAR	Compressor Source Current Address Register	
CSCHR	Compressor Source Count Holding Register	
CSLSR	Compressor Source Line Start Address Register	
CSO	Compressor Source Overflow bit	CSR
CSR	Compressor Status Register	
CSWCR	Compressor Source Working Count Register	
CWR	Compressor Wraparound Register	
DAC	Destination Address Control bit	CRCR, ERCR
DALE	Destination Address Latch Enable	
DC	Destination Control bit	CMCR, EMCR
DCC	Destination Count Control bit	CRCR, ERCR
DER	Data Error bit	ESR
DFC	Data Format Control bits	CPR
DLS	Destination Line Start Address Control bit	CRCR, ERCR
DMA	Direct Memory Access	
DRD	Document Store Read	
DREADY	Document Store Ready	
DWR	Document Store Write	
EBOCR	Expander Bit Offset Control Register	
EBY	Expander Busy bit	MSR, ESR
ECD	Extension Code Detected bit	MSR
EDAHR	Expander Destination Address Holding Register	
EDCAR	Expander Desintation Current Address Register	
EDCHR	Expander Destination Count Holding Register	
EDLSR	Expander Destination Line Start Address Register	
EDO	Expander Destination Overflow bit	ESR
EDWCR	Expander Destination Working Count Register	
EFWR	Expander Frame Width Register	
EIC	Expander Illegal Command bit	ESR
EIE	Expander Interrupt Enable bit	EMCR
EKPR	Expander K-Parameter Register	
EMCR	Expander Master Control Register	
EOA	Expander Busy and New Operation Attempted bit	ESR
EOL	End-of-line bit	CPR, EPR

EOP	End-of-page (Group IV)	MSR
EPR	Expander Parameter Register	
EPWR	Expander Page Width Register	
ERCAR	Expander Reference Current Address Register	
ERCR	Expander Restart Control Register	
ESA	Expander Source Address bit	EPR
ESAHR	Expander Source Address Holding Register	
ESCAR	Expander Source Current Address Register	
ESCHR	Expander Source Count Holding Register	
ESLSR	Expander Source Line Start Register	
ESO	Expander Source Overflow bit	ESR
ESR	Expander Status Register	
ESWCR	Expander Source Working Register	
EWR	Expander Wraparound Register	
EXT	Extention bits	MSR
GO	Go	CMCR, EMCR
GP	G-Parameter bits	EPR
HLDA	Hold Acknowledge	
HRQ	Hold Request	
ID	Identification bit (Am7970A = 1; Am7971A = 0)	MSR
INTR	Interrupt Request	
LBO	Left Bit Offset Control bits	CBOCR, EBOCR
LMGR	Left Margin Register	
LPI	Line Processing Incomplete bit	CSR, ESR
MC	Mode Control bits	CMCR, EMCR
MH	Modified Huffman (coding)	
MMR	Modified MR (Group IV coding)	
MR	Modified READ (Group III coding)	
MSR	Master Status Register	
NGC	Negative Compression bit	CSR
OC	Operation Control bits	CMCR, EMCR
PEL	Picture Element	
PIXEL	Picture Element	
RBO	Right Bit Offset Control bits	CBOCR, EBOCR
RD	Read	
READ	Relative Element Address (coding)	
READY	Ready	
RESET	Reset	
RFA	Reference Line Access	
RMGR	Right Margin Register	
RTC	Return-to-Control code (six EOLs)	
SA	Source Attribution bit	CPR, EPR
SAC	Source Address Control bit	CRCR, ERCR
SC	Source Control bit	CMCR, EMCR
SCC	Source Count Control bit	CRCR, ERCR
SLS	Source Line Start Address Control bit	CRCR, ERCR
TFLR	Time Fill Register	
TMGR	Top Margin Register	
WPI	Wrap-around Incomplete bit	CSR, ESR
WR	Write	

Am95C71

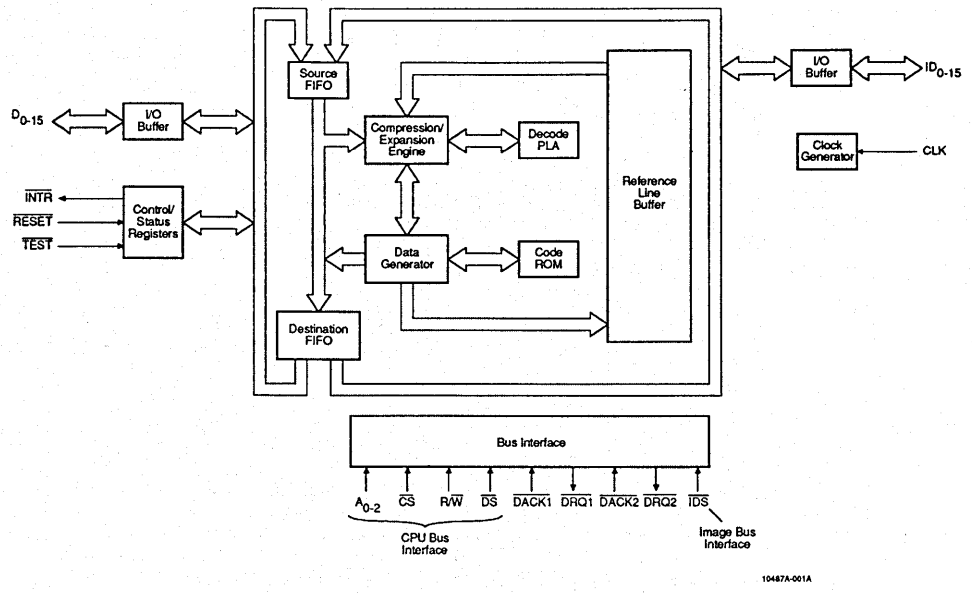
Video Compression/Expansion Processor (VCEP)

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- Throughput exceeding an average rate of 50 Mb/s when compressing or expanding
- Full CCITT Group 3 and Group 4 compression/expansion: allows MH, MR and MMR coding and transparent mode
- Dual-bus architecture with single-bus mode option
- Supports bit-boundary image width up to 8191 pixels in one-dimensional (1D) mode and 6911 pixels in two-dimensional (2D) mode
- Has on-chip 6911-pixel reference-line buffer allowing high-performance 2D coding
- Provides error detection and recovery capability
- Supports programmable k-Parameter for 2D coding
- 16-word FIFOs on input and output
- Half-duplex operation

BLOCK DIAGRAM



BD008180

GENERAL DESCRIPTION

The Am95C71 Video Compression/Expansion Processor (VCEP) is a high-performance CMOS processor which compresses and expands binary image data using the internationally standardized CCITT Group 3 and Group 4 algorithms.

The VCEP supports the Modified Huffman (MH), Modified Read (MR), and Modified-Modified Read (MMR) coding schemes used by the CCITT Groups 3 and 4 standards. MH coding is a one-dimensional technique which identifies and then codes run-lengths of black or white pixels. MR coding compresses a single scan line using MH coding, followed by k-1 scan lines coded in such a way as to reflect differences from the pixel patterns of the previous scan line (two-dimensional or 2D coding); the value of the k-Parameter is defined by the user and will generally be set to a larger number on communication links with lower bit-error rates.

MMR coding is a full 2D-coding scheme which uses an all-white imaginary reference line when coding the first scan line. All lines on the page are coded two-dimensionally. For a typical binary image, MMR coding offers the best compression, followed by MR and then MH. Compressed data may be corrupted during transmission or storage. Error-free (or error-protected) transmission media are used with Group 4 coding, since error recovery is not possible. The CCITT standard refers to MH and MR coding as Group 3 techniques and MMR coding as a Group 4 technique. Group 3 error recovery facility is provided on the VCEP.

The extent of data compression provided by Group 3 and Group 4 compression techniques depends on the specific data patterns contained in the image. Typically, an originally black-or-white (binary) image will yield compression ratios between 5:1 and 50:1, whereas a binary image produced from a grey-scale or color original may compress poorly, even resulting in a compressed file larger than the raw image. Alternatively, the user may program the VCEP into transparent mode where data is simply passed from source FIFO to destination FIFO without compressing or modifying the data.

When 2D (MR or MMR) coding is performed, the previous scan line is used as a reference to code the current line. To significantly increase performance the VCEP stores the reference line in an on-chip buffer.

The VCEP is a slave-mode device with two 16-bit bus interfaces. The user may select either bus to be source or destination and the VCEP to compress, expand or pass through (transparent mode) data. Data is buffered on input and output by 16-word FIFOs. The VCEP, therefore, may be used as a single-bus or dual-bus device, with FIFO buffers on input and output.

The VCEP may either compress, expand, or pass through data; it cannot do these functions simultaneously or in a multiplexed fashion and is therefore termed a half-duplex device.

However, it is possible for the VCEP to multiplex data compression from several sources if a full scan line is processed from each source, and MH coding is selected. Multiplexed expansion is not supported.

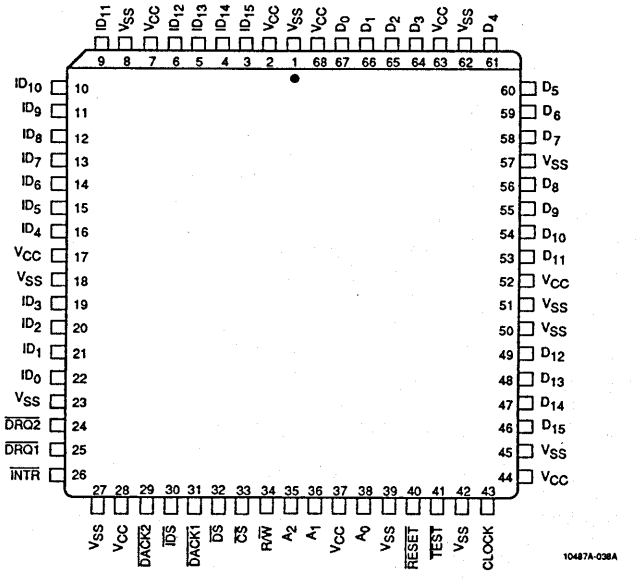
The VCEP has several mechanisms to detect data errors on expansion. For MH and MR modes, if the expanded scan line is longer or shorter than the user-programmed length, the VCEP sets a flag and halts. Illegal codes, negative run lengths in 2D coding, and other illegal fields are detected as errors. Since the VCEP has no on-chip DMA, the host CPU is responsible for error recovery; for example, by replicating the previous scan line when an error is found in the current scan line.

The VCEP has programmable bus burst and dwell counters to allow the user control over the length of the VCEP's data requests and the time between requests.

All registers on the VCEP are set up by the CPU via the VCEP's CPU bus, selecting specific registers with three address lines. In the dual-bus configuration, data is accessed on the Image bus by a slave-mode access which does not require an address.

CONNECTION DIAGRAMS
Top View

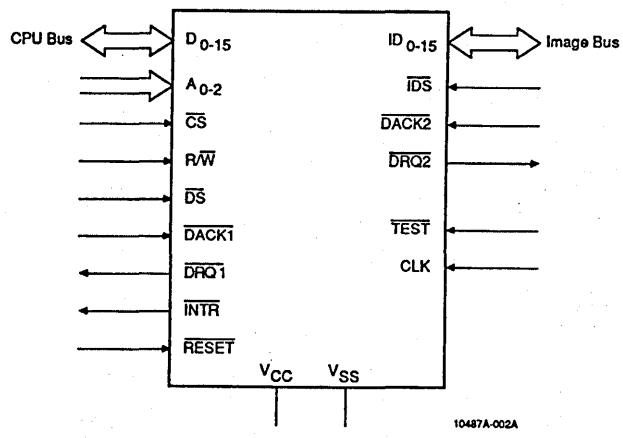
PLCC



CD011640

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



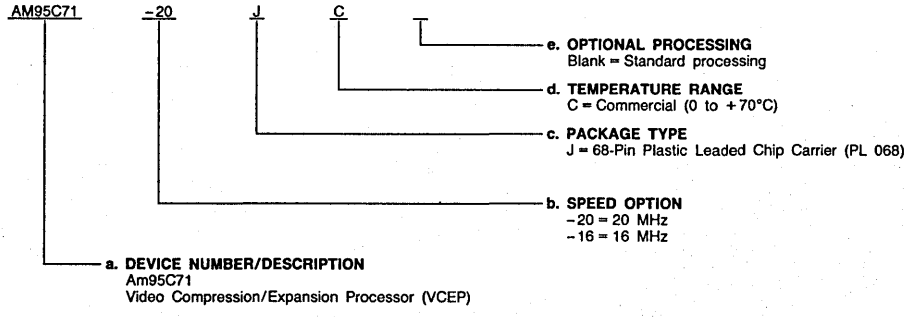
LS003291

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C71-20	JC
AM95C71-16	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

A_{0-2} Address Bus (Input)
A 3-bit address used to select one of seven internal registers. These pins are ignored when performing a fly-by transfer on the CPU bus.

CLK Clock (Input)
Master timing of the VCEP is provided by an external source connected to CLK.

$\overline{CS}$ Chip Select (Input; Active LOW)
Qualifies $\overline{DS}$ when performing a flow-through register access via the CPU bus. Chip Select must be inactive when performing a fly-by transfer on the CPU bus.

D_{0-15} CPU Data Bus (Input/Output; Three-state)
A 16-bit bidirectional data bus used to transfer data, commands and status to or from the VCEP.

$\overline{DACK1}$ Data Transfer Acknowledgement 1 (Input; Active LOW)
Controls data transfers to or from the FIFO Data Port.

$\overline{DACK2}$ Data Transfer Acknowledgement 2 (Input; Active LOW)
Controls data transfers to or from the FIFO Data Port.

$\overline{DRQ1}$ Data Transfer Request 1 (Output; Active LOW)
Controls data transfers to or from the FIFO Data Port.

$\overline{DRQ2}$ Data Transfer Request 2 (Output; Active LOW)
Controls data transfers to or from the FIFO Data Port.

$\overline{DS}$ Data Strobe (Input; Active LOW)
Controls data transfers on the CPU data bus.

ID_{0-15} Image Data Bus (Input/Output; Three-state)
A 16-bit bidirectional data bus used to transfer data to or from the VCEP.

$\overline{IDS}$ Image Data Strobe (Input; Active LOW)
Controls data transfers on the Image Data Bus.

$\overline{INTR}$ Interrupt Request (Output; Active LOW)
$\overline{INTR}$ is asserted when an exception or termination condition occurs and the user has previously set the Interrupt Enable bit in the Command/Status Register. $\overline{INTR}$ is made inactive when the Command/Status Register is read or when the VCEP is reset.

R/$\overline{W}$ Read/$\overline{Write}$ (Input)
Controls the direction of transfer on the CPU bus when accessing one of the internal registers. This signal is ignored when performing a fly-by transfer on the CPU bus.

$\overline{RESET}$ Reset (Input; Active LOW)
$\overline{RESET}$ is an asynchronous, active-LOW input which initializes VCEP to an idle state. $\overline{RESET}$ must be driven LOW for at least four clock cycles to ensure proper operation.

$\overline{TEST}$ Test (Input; Active LOW)
When $\overline{TEST}$ is held LOW, all VCEP outputs are three-stated. In normal use this pin should be tied to V_{CC} or held at a TTL-HIGH level.

FUNCTIONAL DESCRIPTION

Register Description

The VCEP has seven user registers: the FIFO Data Port, Command/Status Register, Parameter Register, Burst Count Register, Dwell/Offset Register, Line Length Register and Line Count Register. These registers are shown in Figure 1-1.

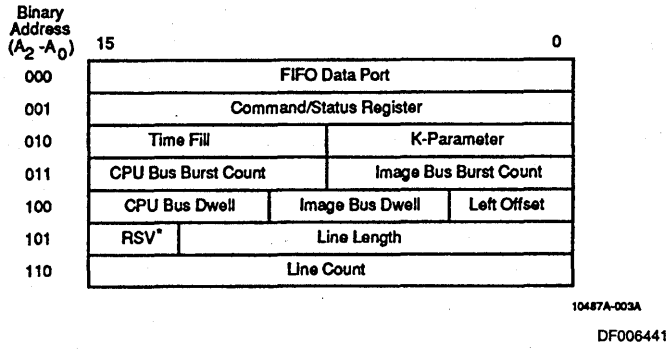


Figure 1-1. VCEP Register Set

OVERVIEW OF REGISTERS

The Command/Status Register (CSR) contains 11 Command bits and 4 Status bits (see Figure 1-2).

GO: The GO bit is used to start VCEP operation in the mode indicated by the OM field compression, expansion or transparent.

Operational Mode (OM): The OM field defines whether VCEP should be in compress or expand (MH, MR or MMR), transparent or reset mode.

Line Mode (LM): When LM = 0 the VCEP will process one scan line before stopping. When LM = 1 the VCEP processes a full image before stopping.

Boundary Mode (BM): This bit defines whether the first code data word of a page is on a word boundary, BM = 0, or on an odd-byte boundary, BM = 1.

Interrupt Enable (IE): When IE = 1, the $\overline{INTR}$ output is asserted whenever the VCEP encounters an exception or termination condition.

Compressor Word-boundary Control (CW): In MH or MR mode, if CW = 1 the VCEP pads the end of the code line to ensure it ends on a word boundary. In MMR mode this bit is ignored.

Source Control (SC): If SC = 0, the CPU bus is selected as the data source. If SC = 1, the Image bus is selected as data source.

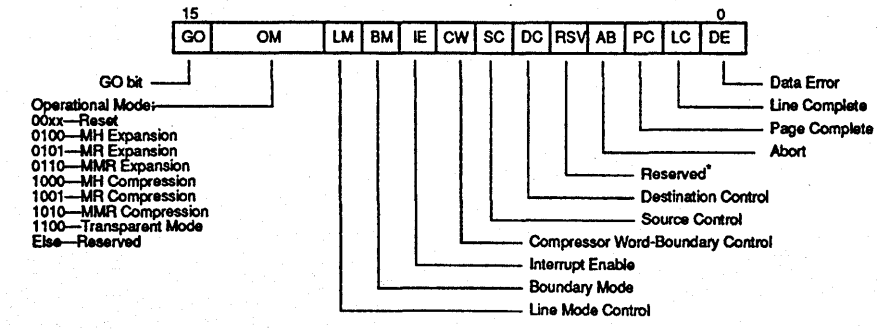


Figure 1-2. Command/Status Register

Destination Control (DC): If DC = 0, the CPU bus is selected as the data destination. IF DC = 1, the Image bus is selected as data destination.

The remaining bits indicate status information:

Abort (AB): If the Command/Status Register is written while the VCEP is busy, the VCEP stops processing and sets AB = 1.

Page Complete (PC): When the VCEP detects the RTC/EOP code in expansion mode it sets this bit. In compression or transparent mode this bit is set when the Line Count Register has been decremented to zero.

Line Complete (LC): When the VCEP is in single-line mode it sets the Line Complete bit each time VCEP has processed a line of data. This bit may also be set when a data error occurs.

Data Error (DE): When the VCEP detects a data error on expansion this bit is set.

The Parameter Register contains an 8-bit Time-Fill value and an 8-bit k-Parameter. Time-Fill is used in MH and MR Compression modes and its value specifies the minimum length of a coded line in words, which may vary from 0 to 255 words. The k-Parameter specifies for MR compression mode how many lines (k-1) will be compressed using two-dimensional coding after a one-dimensionally coded reference line. "k" may vary from 1 to 255 and infinity, indicated by k = 0.

The Burst Count Register contains an 8-bit CPU bus burst count and an 8-bit Image bus burst count.

The Dwell/Offset Register contains a 6-bit CPU bus dwell count, a 6-bit Image bus dwell count, and a 4-bit Left Offset Register (LOR).

The Line Length Register (LLR) specifies the number of pixel elements contained in one line. For one-dimensional coding the total of LLR + LOR must not exceed 8191 pixels. For two-dimensional coding LLR + LOR must not exceed 6911 pixels, due to the internal reference line buffer size. The minimum value of LLR + LOR in either case is 17 pixels.

The Line Count Register (LCR) should be set to the number of scan lines of an image when data compression is used and will be decremented each time the VCEP processes one line in compression or transparent mode. The LCR value is incremented each time the VCEP expands a line. The user may read this value from the LCR Register at any time, while the last value written to the LCR Register is stored elsewhere internally and will be reloaded at the start of a new page.

The FIFO Data Port is a 16-bit I/O port through which image and compressed data is accessed.

All Reserved bits ahould be set to "0" by the user.

Operational Overview

The VCEP has two 16-bit data buses. The control signals associated with each bus depend on whether the VCEP is programmed for single-or dual-bus operation and the type of transfer performed. The CPU bus supports two methods of transferring data into or out of the VCEP. Flow-through operations on the CPU bus use $\overline{CS}$, R/$\overline{W}$, $\overline{DS}$, and A_{0-2} to determine whether a read or write is to be performed o n one of seven internal registers. Fly-by operations on the CPU bus use $\overline{DS}$ and $\overline{DACK1}$ or $\overline{DACK2}$, depending on bus configuration selected, to perform a transfer to or from the FIFO Data Port. The Image bus supports only fly-by operations to or from the FIFO Data Port and uses $\overline{IDS}$ and $\overline{DACK1}$ or $\overline{DACK2}$, depending on bus configuration selected, to perform a transfer.

The VCEP requires data to be both written to and read from the FIFO Data Port. The SC field in the Command/Status Register (CSR) is used to select whether the CPU or Image bus will be used for supplying data to the source buffer via the FIFO Data Port. The DC field of the CSR selects whether the CPU or Image bus will be used for reading data from the destination buffer via the FIFO Data Port. The SC and DC fields are independent and may be programmed to any of four possible configurations. Of these possibilities, two assign a single bus for reading and writing source and destination buffers. The remaining two possibilities have source and destination assigned to different buses.

Table 1 defines the settings of SC and DC, together with use of $\overline{DRQ1}$, $\overline{DACK1}$, $\overline{DRQ2}$ and $\overline{DACK2}$. The data acknowledge signals associated with the CPU bus, $\overline{DACK1}$ and/or $\overline{DACK2}$, are used in fly-by operations only and must both be inactive whenever $\overline{CS}$ is active.

TABLE 1. VCEP BUS ASSIGNMENT

SC	DC	Source Bus	Input Pins	Control Pins	Destination Bus	Output Pins	Control Pins
0	0	CPU	D_{0-15}	$\overline{DRQ1}/\overline{DACK1}$	CPU	D_{0-15}	$\overline{DRQ2}/\overline{DACK2}$
0	1	CPU	D_{0-15}	$\overline{DRQ1}/\overline{DACK1}$	Image	ID_{0-15}	$\overline{DRQ2}/\overline{DACK2}$
1	0	Image	ID_{0-15}	$\overline{DRQ2}/\overline{DACK2}$	CPU	D_{0-15}	$\overline{DRQ1}/\overline{DACK1}$
1	1	Image	ID_{0-15}	$\overline{DRQ1}/\overline{DACK1}$	Image	ID_{0-15}	$\overline{DRQ2}/\overline{DACK2}$

Detailed Functional Description

The VCEP is a streamlined, high-speed compression/expansion engine. The inclusion on-chip of a 6911-pixel reference line buffer allows very efficient two-dimensional coding while a minimal set of user registers combines ease of programming with operational flexibility. By using FIFOs at input and output the VCEP presents a simple, buffered slave-mode interface to CPU and Image buses.

The VCEP may be clocked asynchronously from the system CPU. In addition, the CPU and VCEP clock rates may differ significantly within the following limitations.

When the VCEP clock is slower than the CPU clock, a lower limit is reached if more than one complete data transfer cycle occurs in each VCEP clock cycle. One data transfer may occur on each bus of a dual-bus system since the buses are fully independent.

When the VCEP clock is faster than the CPU clock, the upper limit is defined by the maximum low pulse width of $\overline{DS}$, which is 15 VCEP clock cycles.

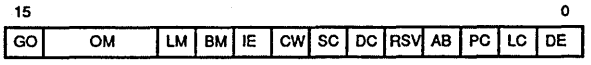
To start VCEP operation after a hardware Reset, the CPU must initialize VCEP registers, specify Operational Mode (OM), select source and destination buses, and choose whether to process single scan lines or full pages. Other parameters may need to be selected depending on choice of operational

mode. Additionally, a software Reset should be asserted prior to processing each page.

The typical pattern of operations after a hardware Reset would be: wait at least eight clock cycles after asserting reset; set up parameters and select mode; assert GO = 1; wait until line processing (if LM = 1) or page processing terminates and LC = 1 or PC = 1; read Status bits; select Reset mode; wait at least eight clock cycles; change parameters if needed and repeat.

COMMAND/STATUS REGISTER (CSR) ADDRESS = 001

The CSR contains 11 Command bits and 4 Status bits, described below. Note that after writing a register, the system should wait a minimum of eight clocks before accessing that register again.



10487A-005A
DF006460

Figure 2. Command/Status Register

Bit 15 — GO: The GO bit is used to start VCEP operation in the mode indicated by the OM field. The user must set up all parameters and operational mode before setting GO = 1. If GO is set and the VCEP is not busy, the VCEP will clear Status bits and start the selected operation. When the operation is complete, the VCEP resets GO and halts. If the user attempts to write the CSR while GO is set and the VCEP is busy, the VCEP will halt the current operation and interrupt the CPU if the Interrupt Enable (IE) bit is set. This condition requires the user to provide the VCEP with a software reset.

When GO = 0, the VCEP is idle and the user may access all registers except the FIFO Data Port, which may yield unpredictable results and if accessed, requires the CPU to reset the VCEP. If the user has programmed the VCEP to compress, expand or pass-through (transparent mode) a complete page of data, Status bits are set and GO is reset when the page has been processed and all data read from the destination buffer by the user, or when an error condition is detected. In single-line mode (LM = 0) Status bits are set and GO is reset when the VCEP has processed a single line of data and the user has emptied the destination buffer. The system may begin processing in single-line mode and then switch to multi-line mode at the end of any scan line.

The user should never assert the GO bit when selecting a software Reset.

When the VCEP finishes an operation and resets GO it will also generate an interrupt if IE = 1.

GO = 0 after a hardware or software Reset.

Bits 14 – 11 — Operational Mode (OM): The OM field defines whether VCEP should compress, expand, pass-through data (transparent mode) or do a software Reset. When compression or expansion is selected, the OM field also defines whether MH, MR or MMR coding is to be used. Table 2 defines the OM bits.

TABLE 2. OPERATIONAL MODE BIT DEFINITIONS

Bit 14	Bit 13	Bit 12	Bit 11	Operational Mode
0	0	0	0	Software Reset
0	0	0	1	Software Reset
0	0	1	0	Software Reset
0	0	1	1	Software Reset
0	1	0	0	MH Expansion
0	1	0	1	MR Expansion
0	1	1	0	MMR Expansion
0	1	1	1	Reserved
1	0	0	0	MH Compression
1	0	0	1	MR Compression
1	0	1	0	MMR Compression
1	0	1	1	Reserved
1	1	0	0	Transparent Mode
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

Each Operational Mode is explained in considerable detail in the section on Operating Modes.

Bit 10 — Line Mode (LM): When LM = 0 the VCEP is in single-line mode and when set to LM = 1 the VCEP is in multi-line mode. In single-line mode, when the user selects an operational mode besides Reset, the VCEP will process a single line, waiting until the destination buffer has been emptied, before it clears GO, sets Line Complete (LC = 1), and halts. When in multi-line mode and when compressing or in transparent mode, the VCEP will process data until the Line Count Register is zero. When it detects LCR = 0 it waits until all data has been read from the destination buffer, clears GO, sets Page Complete (PC = 1), and halts.

Bit 9 — Boundary Mode (BM): This bit defines whether the first code data word of a page is on a word boundary, BM = 0, or on an odd-byte boundary, BM = 1. When BM = 1 and VCEP is compressing, it inserts eight zeros before the code data so that the code begins the page on an odd-byte boundary. If BM = 0 no zeros are inserted. When BM = 1 and VCEP is expanding, it ignores the first eight bits of the first code word of the page and begins expanding from the second code byte of the page (bit 8). When BM = 0 and the VCEP is expanding, it begins from bit 0 of the first code word.

Bit 8 — Interrupt Enable (IE): When IE = 1, the $\overline{INTR}$ output is asserted whenever VCEP encounters an exception or termination condition. When IE = 0, $\overline{INTR}$ is never asserted.

Bit 7 — Compressor Word-boundary Control (CW): In MH or MR Compression mode, if CW = 1 the VCEP will insert 0 to 15 pad bits (zeros) at the end of each line of code data to ensure that the line ends on a word boundary. In MMR Compression mode this bit is ignored.

Bit 6 — Source Control (SC): If SC = 0, the CPU bus is selected as the data source. If SC = 1, the Image bus is selected as data source.

Bit 5 — Destination Control (DC): If DC = 0, the CPU bus is selected as the data destination. IF DC = 1, the Image bus is selected as data destination.

Bit 4: Reserved.

The remaining bits indicate status information. Status bits are reset when GO = 1 or when the system performs a software Reset of the VCEP.

Bit 3 — Abort (AB): If the Command/Status Register is written while the VCEP is busy, the VCEP will abort processing and set this bit. To restart the VCEP, the user must perform a software or hardware Reset and wait at least eight clocks before setting an active operational mode and writing GO = 1. If an Abort occurs, this is a non-recoverable condition and all data currently in process will be lost.

Bit 2 — Page Complete (PC): When VCEP detects the RTC/EOP code in expansion mode it sets this bit. In compression or transparent mode this bit is set when the Line Count Register has been decremented to zero. Either way this Status bit indicates that the VCEP has processed a page of data.

Bit 1 — Line Complete (LC): When the Line Mode bit is reset (LM = 0) the VCEP is in single-line mode and the Line Complete bit is set when the VCEP has processed a line of data.

Bit 0 — Data Error (DE): When VCEP detects a data error during expansion, this bit is set. Data error conditions are described in the section on Error Detection and Recovery.

A summary of status information is given in Table 3.

TABLE 3. SUMMARY OF STATUS INFORMATION

AB	PC	LC	DE	Explanation
1	X	X	X	VCEP Aborted the Current Operation
0	1	0	0	Page Complete without Error in Multi-line Mode
0	0	1	0	Line Complete without Error in Single-line Mode
0	0	0	1	Data Error (see above)
0	0	1	1	Data Error Detected in EOL Code
0	1	0	1	RTC/EOP Detected with Data Error

OPERATING MODES

The system must specify LLR, LCR, LOR, Time-Fill, and k-Parameter before setting the GO bit in any mode, except where specified.

Compression

When compressing data, the VCEP reads image data from the source buffer, converts it to code, and writes it to the destination buffer.

If BM = 1 in the CSR, the VCEP puts the first EOL code on an odd-byte boundary by inserting eight zeros into the first word. If CW = 1, the VCEP compresses each scan line and adds from one to fifteen zeros to ensure each line ends on a word boundary. The VCEP also inserts bits before the next line's EOL code if the current line is shorter than the minimum length specified by the Time-Fill Parameter Register.

The user specifies scan line length in the Line Length Register (LLR) and as the VCEP reads image data from the source buffer, it will ignore the first LOR bits of each scan line, where the user defines LOR, the Left Offset Register, which may vary from 0 to 15 bits. The user must also program the VCEP's Line Count Register (LCR), specifying the number of scan lines in a page. Once the VCEP has compressed LCR scan lines this is the end of a page.

In all compression modes once the VCEP has finished compressing a page it will wait until the destination FIFO has been emptied before it sets PC = 1 in the CSR and halts. In addition, if LM = 0 in the CSR (single-line mode), the VCEP waits until its destination FIFO is emptied after compressing each scan line, sets LC = 1, and halts. If LM = 0 and the code data does not end on a word boundary, the VCEP retains the last partial word if no padding or Time-Fill bits are added.

MH Compression

When the CPU sets OM = 1000, the VCEP is in MH Compression mode, where each scan line is 1D-compressed and prefixed with an EOL (End-of-Line) code; the k-Parameter is ignored. Once it has compressed a scan line and added pad bits as needed, it decrements the Line Count Register. When this count reaches zero, the VCEP adds six EOL codes, indicating the RTC (Return-to-Control) code to indicate end-of-page. MH Compression mode data format is shown in the Data Formats section.

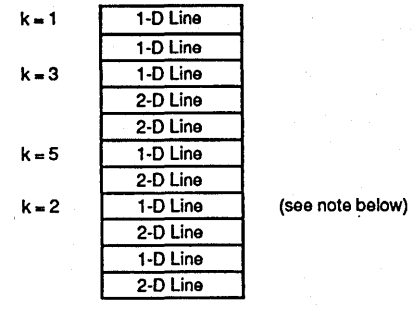
MR Compression

When OM = 1001 in the CSR, the VCEP is in MR Compression mode, where it 1D-compresses the first scan line and 2D-compresses the following (k-1) lines. The code for each scan line is preceded by an EOL code with Tag bit. Tag = 1 for 1D-compressed lines; Tag = 0 for 2D-compressed lines. If k = 0, all but the first scan line are 2D-compressed. The value of k is stored internally and so does not need to be loaded each time the user sets GO = 1, although k may be changed while compressing a page in single-line mode.

When beginning page compression, the VCEP outputs an EOL code with Tag bit set, followed by the 1D code for the first scan line. The VCEP then decrements the Line Count Register and adds the RTC code (six EOL codes with Tag = 1) if the count is zero, to indicate end-of-page. MR compression mode data format is shown in the section on Data Formats.

Using the k-Parameter: The k-Parameter specifies how many lines (k-1) will be compressed using two-dimensional coding after a one-dimensionally coded reference line. "k" may vary from 1 to 255 and infinity. This register is valid only in MR Compression mode and is ignored in all other modes. The most-significant bit of the k-Parameter is bit 7. To set k at infinity, write k = 0. When k = 0, the first scan line of a page is 1D-coded and all remaining scan lines are 2D-coded.

When LM = 0, and single-line mode is used with MR Compression, the user may change the value of k while compressing a document. The effect of k-Parameter on whether a line is 1D- or 2D-coded is shown in Figure 3.



Note: This line is 1D-compressed because the value of the k-Parameter was changed between lines with the VCEP in single-line mode. Changing the k-Parameter is not permitted if multi-line mode is used.

10487A-006A

TB001240

Figure 3. Effect of k-Parameter on Coding

MMR Compression

When OM = 1010 in the CSR, the VCEP is in MMR Compression mode and will 2D-compress all scan lines. The k-Parameter and Time-Fill parameter are ignored.

When the VCEP begins page compression it 2D-compresses the first scan line with respect to an imaginary white reference line (all zeros). For subsequent lines, the VCEP uses as reference the data in its internal line buffer. At the beginning of page, VCEP will pad the first code with eight zeros if BM = 1. In MMR Compression mode, no EOL code precedes each coded line.

Once the VCEP has compressed a scan line, it decrements the Line Count Register. When this count reaches zero, the VCEP adds two EOL codes, indicating the EOP (End-of-Page) code. MMR Compression mode data format is shown in the section on Data Formats.

Expansion

In all expansion modes, the VCEP reads code data that the user has written to the source buffer, expands it, and places it in the destination buffer. The k-Parameter and Time-Fill parameter are ignored.

Prior to beginning page expansion, the user defines scan line length in the Line Length Register (LLR), and as the VCEP expands code data and writes it to the destination buffer, it compares current line length to LLR, allowing error detection. The VCEP will also write the first expanded word of each scan line offset by LOR bits, where the user defines LOR (Left Offset Register), which may vary from 0 to 15 bits. The VCEP increments the Line Count Register (LCR) after expanding a scan line. At the end of the page, the user may read LCR to determine the number of scan lines in a page. When the user sets GO = 1 to begin processing an image, LCR is loaded with the last count programmed (since this is retained internally between pages). The VCEP then begins expanding a new page. If BM = 0 the VCEP will search from bit 0 of the first code word; if BM = 1, the VCEP searches from bit 8.

MH Expansion

When OM = 0100 in the CSR, the VCEP is in MH Expansion mode, where it 1D-expands all code data.

MR Expansion

When OM = 0101 in the CSR, VCEP is in MR Expansion mode, where it 1D- or 2D-expands code data, depending on the detected value of the Tag bit following EOL codes. If Tag = 1, data is 1D-expanded, while Tag = 0 will cause 2D expansion.

MMR Expansion

When OM = 0110, the VCEP is in MMR Expansion mode, where it will 2D-expand all code data.

Transparent Mode

When OM = 1100, the VCEP is in Transparent mode, where it transfers data from source to destination buffers without modification. Transparent mode is a simple mechanism to allow data flow between CPU and Image buses. The Time-Fill and k-Parameters are ignored and LOR should be set to zero. The VCEP still uses LLR and LCR values. When the VCEP

reaches the end of the page (LCR = 0), it waits until the destination buffer has been emptied, sets PC = 1 in the Status Register, and halts. In single-line mode (where LM = 0), when the VCEP detects the end of the scan line (LLR reaches the programmed value), it waits until the destination buffer has been emptied, sets LC = 1, and halts.

Reset

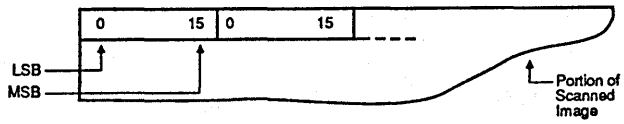
When OM = 00xx, the VCEP is reset. This software Reset has the same effect on the VCEP as a hardware Reset, clearing source and destination buffers, all Status bits, and the GO bit. Other register contents are unchanged. The GO bit should not be set when performing a software Reset. Once the user has performed a software Reset, the CSR should not be accessed for at least eight clock (CLK) cycles.

DATA FORMATS

Image Data Format

The VCEP processes (compresses or expands) images digitized as described below:

1. A single image is represented by an array of black and white pixels.
2. Each row of pixels is represented by ones and zeros:
   White pixel = 0
   Black pixel = 1
3. The image bit stream is assumed to be generated by a scanner which moves from left to right across the page while scanning each line.
4. Bit 0 of each word of the resulting bit stream is considered to be the least-significant bit.
5. Bit 0 of a code word is the least-significant bit.



DF006470

Figure 4. Scanned Data Stream to/from VCEP

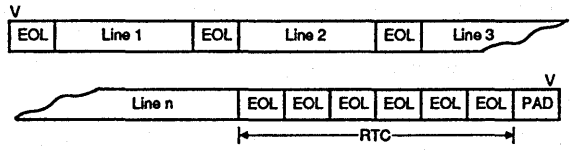
Compressed Data Format

The compressed data formats for MH, MR and MMR modes are shown below.

MH Code Compressed Data Format without Time-Fill Bits

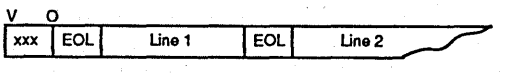
If BM = 1 in the CSR, the prefixed EOL starts from bit 8 of the first word, which is the odd-byte boundary (Figure 6).

If CW = 1 in the CSR, the VCEP's compressor adds pad bits (0 to 15 zeros) to the end of the coded line, to make the coded line end on a word boundary (Figure 7).



DF006720

Figure 5. MH Coded Data without Time-Fill

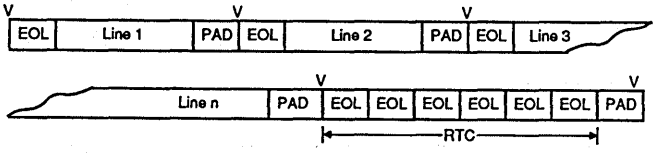


Where O : Odd-Byte Boundary
xxx : Eight "Don't Care" Bits

10487A-009A

DF006490

Figure 6. MH Coded Data with BM = 1



10487A-010A

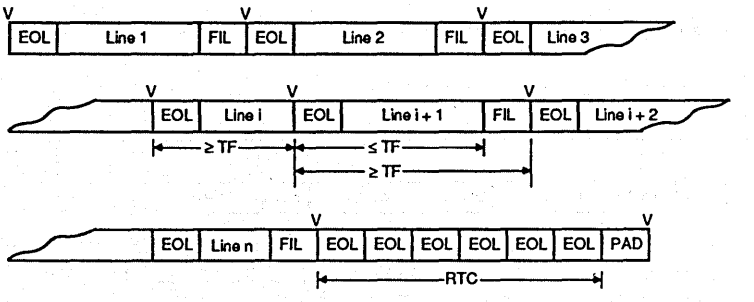
DF006730

Figure 7. Padded MH Coded Data

MH Mode Compressed Data Format with Time-Fill Bits

If BM = 1 in the CSR, the prefixed EOL of the first line starts from bit 8 of the first word, on the odd-byte boundary (Figure 9).

If CW = 1 in the CSR, the VCEP's compressor will insert pad bits (0 to 15 zeros) to the last code of the line, such that the coded line ends on a word boundary. Since the Time-Fill bits will always end on a word boundary, no pad bits will be added when Time-Fill occurs (Figure 10).

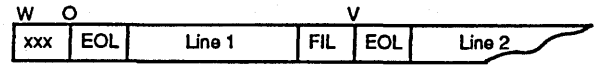


Where V : Word-Boundary Mark
EOL : End-of-Line Code Word
FIL : Time-Fill Bits (Variable Number of Zeros)
PAD : Return-to-Control Code Word
TF : Time-Fill Parameter Specified in Parameter Register

10487A-011A

DF006510

Figure 8. MH Coded Data with Time-Fill

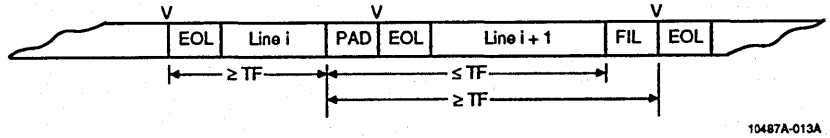


Where O : Odd-Byte Boundary
xxx : Eight "Don't Care" Bits

10487A-012A

DF006521

Figure 9. MH Coded Data with Time-Fill and BM = 1



10487A-013A

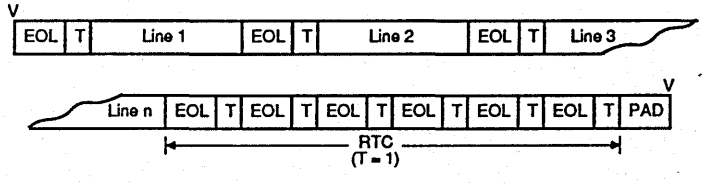
DF006530

Figure 10. Padded MH Coded Data with Time-Fill

MR Mode Compressed Data Format without Time-Fill Bits

If BM = 1 in the CSR, the prefixed EOL of the first coded line will be at the odd-byte boundary of the first word (Figure 12).

If CW = 1 in the CSR, the VCEP's compressor will insert pad bits (0 to 15 zeros) at the end of the coded line to ensure it ends on a word boundary (Figure 13).

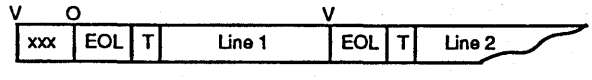


Where V : Word-Boundary Mark
EOL : End-of-Line Code
T : Tag Bit
PAD : 0 to 15 Zeros
RTC : Return-to-Control Code Word

10487A-014A

DF006540

Figure 11. MR Coded Data without Time-Fill



Where O : Odd-Byte Boundary
xxx : Eight "Don't Care" Bits

10487A-015A

DF006550

Figure 12. MR Coded Data with BM = 1

5

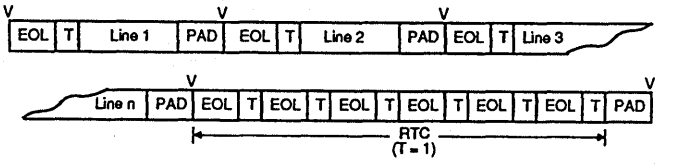
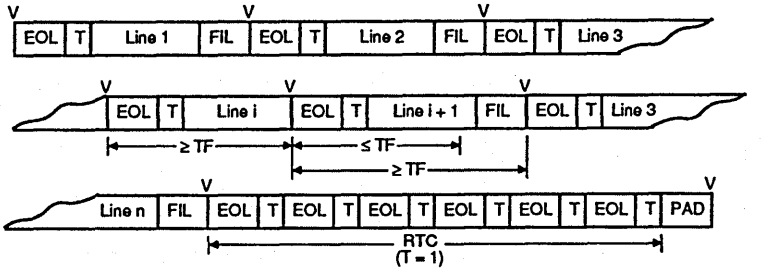


Figure 13. Padded MR Coded Data without Time-Fill

MR Mode Compressed Data Format with Time-Fill Bits

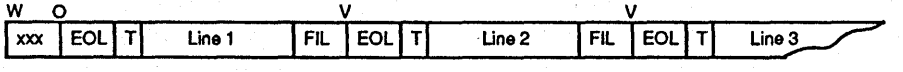
If BM = 1 in the CSR, the prefixed EOL of the first coded line will begin on an odd-byte boundary (Figure 15).

If CW = 1 in the CSR, the VCEP's compressor inserts pad bits (0 to 15 zeros) to the end of the coded scan line, such that it ends on a word boundary. Since Time-Fill bits will always end on a word boundary, no pad bits are necessary when Time-Fill occurs (Figure 16).



Where V : Word-Boundary Mark
EOL : End-of-Line Code
FIL : Time-Fill Bits (Variable Number of Zeros)
T : Tag Bit
PAD : 0 to 15 Zeros
RTC : Return-to-Control Code Word
TF : Time-Fill Value

Figure 14. MR Coded Data with Time-Fill



Where O : Odd-Byte Boundary
xxx : Eight "Don't Care" Bits

Figure 15. MR Coded Data with Time-Fill and BM = 1

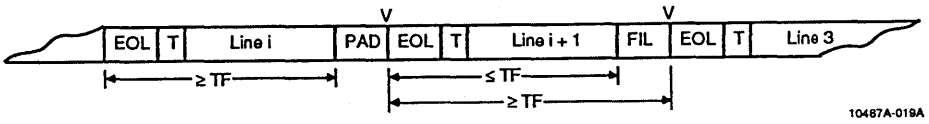


Figure 16. Padded MR Coded Data with Time-Fill

MMR Mode Compressed Data Format

If BM = 1 in the CSR, the first code word of the first scan line will begin on an odd-byte boundary (Figure 18).

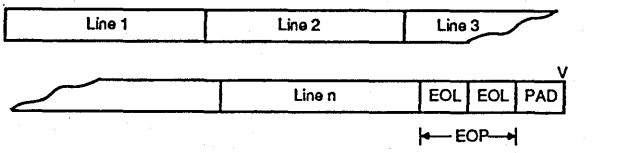


Figure 17. MMR Coded Data

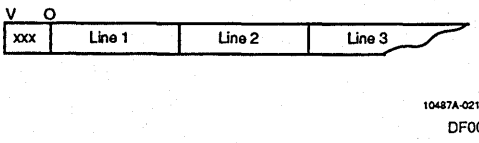


Figure 18. MMR Coded Data with BM = 1

End-of-Line and End-of-Page Codes

Note: Serial data streams are shown here from least- to most-significant bits in all cases.

When Modified Huffman (MH) coding is expanded, the VCEP recognizes the end of a page (known as Return-to-Control or RTC), when it detects three consecutive EOL code words:

(MH code RTC) = 0000 0000 0001
0000 0000 0001
0000 0000 0001

When the VCEP is in MH Compression mode it will append six EOL codes to the end of each page.

When Modified Read (MR) coding is expanded, the VCEP recognizes the end of a page (RTC) when it detects three EOL codes, as in MH coding, but each EOL code has a single Tag bit (set to one) added:

(MR code RTC) = 0000 0000 0001 1
0000 0000 0001 1
0000 0000 0001 1

When the VCEP is in MR Compression mode it will append six EOL codes with Tag bits set, to the end of each page.

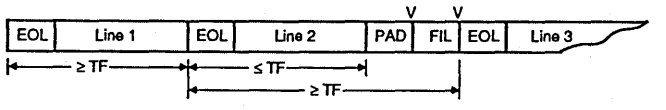
When MMR coding is used, the end of a page (now called EOP) is indicated by two EOL code words:

(MMR code EOP) = 0000 0000 0001
0000 0000 0001

When the VCEP is in MMR Compression mode it will append two EOL codes to the end of each page.

Time-Fill Parameter

Parameter Register (PMR), Bits 15 – 8 Time-Fill (TF): This value specifies the minimum length of a coded line in words, which may vary from 0 to 255 words or from 0 to 4080 bits. VCEP will pad all coded lines less than the value specified here by inserting zeros after the end of the coded data and before the EOL code, with the Time-Fill bits always ending on a word boundary. Time-Fill is performed only when MH and MR coding are used. Minimum line length is calculated as the sum of lengths of the coded scan line, pad bits (to ensure the last code word ends on a word boundary) and the EOL code. An example of Time-Fill in the coded data stream is given in Figure 19.



Where V : Word-Boundary Mark
EOL : End-of-Line Code
FIL : Time-Fill Bits
PAD : 0 to 15 Zeros
TF : Time-Fill Value

10487A-022A

DF006620

Figure 19. Example of Time-Fill

The Time-Fill parameter may specify a minimum line length of 0 to 255 words, where bit 15 in the Parameter Register is the most-significant bit and bit 8 is the least-significant bit.

ERROR DETECTION AND RECOVERY

Detected error conditions are as follows. In MH/MR Expansion modes:

1. The expanded scan line length is longer than the Line Length Register before EOL is detected.
2. The expanded scan line length is shorter than the Line Length Register when the EOL is detected.
3. The expanded scan line length is shorter than the Line Length Register when the RTC/EOP is detected.
4. The codes indicate at any time that a negative run length has occurred.
5. An illegal code word is detected. These codes are:

   0000 0000 1,
   0000 0000 01,
   and 0000 0000 001.

   Also, if an error occurs in the last code word in a line such that it ends in one or more zeros, this may cause false EOL recognition and status will be LC = 1 and DE = 1.
6. In 2D-coding, the following illegal codes are detected:

   0000 001,
   and 0000 0001.
7. Two consecutive EOL codes are detected followed by a non-EOL code word. In the event of this error LC is also set.

In MMR Expansion mode:

1. The expanded scan line length is longer than the Line Length Register.
2. The codes indicate at any time that the run length is negative.
3. An illegal code word is detected. These codes are:

   0000 001,
   0000 0001,
   0000 0000 1,
   0000 0000 01,
   and 0000 0000 001.
4. An EOL code is detected except as part of a pair indicating EOP.
5. Fill bits are present in the code data.
6. The expanded scan line length is shorter than the Line Length Register when an EOP code is detected.

If an error is detected within a line and the Interrupt Enable bit is set (IE = 1 in CSR), the system will be interrupted; otherwise the system must poll the Status bits in the CSR. Either way, Table 4 shows the possible status code combinations when an error is detected.

TABLE 4. ERROR STATUS CODES

AB	PC	LC	DE	Explanation
0	0	0	1	Data Error
0	0	1	1	Data Error Detected in EOL Code
0	1	0	1	RTC/EOP Detected with Data Error

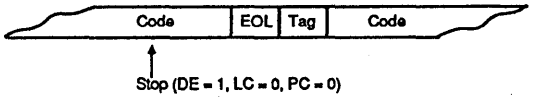
The DE bit is reset after the user issues a command with GO = 1, or when the user issues a software Reset.

When the VCEP detects an error, it stops expanding data, waits until the destination buffer has been emptied, and then sets Status bits and goes idle. When the user executes the error recovery routine for MH or MR modes, the VCEP could be programmed to be in single-line mode to facilitate the user's administration of error recovery. Once the error recovery is complete, the user may switch the VCEP into multi-line mode to process the remainder of the page.

When an error is detected during MMR mode expansion, no recovery is possible; since the page has been encoded two-dimensionally, it is not possible to "pass over" the error without its having a cumulative effect on subsequent data. For this reason (inherent in MMR coding rather than the VCEP) the CCITT has specified that systems using MMR coding should implement error protection apart from the compression code, allowing an effectively error-free transmission environment for the MMR code.

When MH- or MR-coded images are expanded, the three combinations of codes in Table 4 describe four possible error conditions, which together with recovery procedures will be defined below:

Error Condition 1



10487A-023A

DF006630

Figure 20. Error Condition 1

Status: DE = 1, LC = 0, PC = 0

1. The expanded line length is longer than the Line Length Register before the EOL code is detected.
2. A code indicating negative run length is detected.
3. An illegal code (but not a false EOL code) is detected. These are:

   0000 0000 1,
   0000 0000 01,
   and 0000 0000 001.

4. An illegal code is detected in horizontal mode where 1D-codes are expected.
5. In 2D-coding, the following illegal codes are detected:

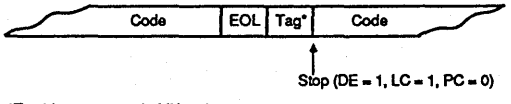
   0000 001,
   and 0000 0001.

Recovery routine:

1. System must replace current scan line containing error with the last correctly expanded scan line.
2. System sets GO = 1.
3. VCEP will search for the next EOL to determine where to restart expansion.

Note that for MR coding, errors will rapidly compound in 2D-coded sections of the image; each time a 2D line is detected in error, it is flagged with a DE = 1 status. Therefore, even for MR coding, the user is only required to administer error recovery on a line-by-line basis.

Error Condition 2



*Tag bit not present in MH code

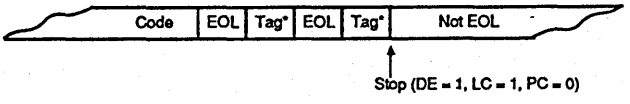
10487A-024A

DF006760

Figure 21. Error Condition 2

Status: DE = 1, LC = 1, PC = 0

1. The expanded line length is shorter than the Line Length Register when the EOL is detected.
2. False EOL codes are detected.
3. Two consecutive EOLs are detected (see Figure 22).



*Tag bit not present in MH code

10487A-025A

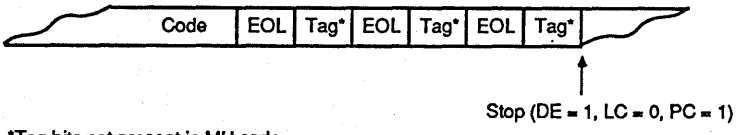
DF006660

Figure 22. Error Condition 2 with Two EOLs

Recovery routine:

1. System must replace current scan line containing error with the last correctly expanded scan line.
2. System sets GO = 1.
3. The VCEP will start expanding the next line immediately, as it has already detected the EOL code. The DE bit is cleared when the system sets GO = 1.

Error Condition 3



*Tag bits not present in MH code

10487A-026A
DF006670

Figure 23. Error Condition 3

Status: DE = 1, LC = 0, PC = 1

The VCEP detects a short line (less than LLR) when the end of page (RTC) has been detected. Note that although the VCEP detects the Tag bits within EOL codes, it will not indicate an error based on the value of these Tag bits.

MH and MR recovery routine:

1. System must replace current scan line containing error with the last correctly expanded scan line.
2. System must issue a software Reset to the VCEP.
3. After allowing a minimum of eight clock cycles from issuing the reset, the system may set up parameters for a new page.

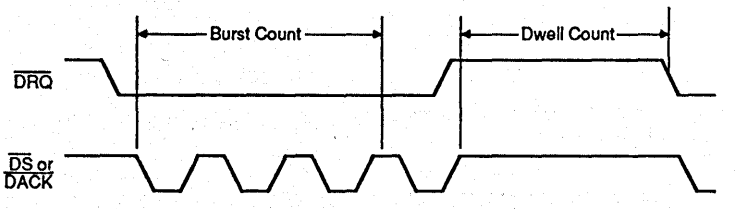
BUS BURST AND DWELL CONTROL

The VCEP is capable of processing up to 16 bits of data every three clock cycles. This may require a bus master to be able to regulate the amount of time for which the VCEP may request data transfer. The VCEP, therefore, has two identical and independent Burst-Dwell counters (one for each bus) that can be used to help regulate bus activity. The burst count specifies the maximum time the VCEP will hold a request active, while the dwell count specifies the minimum time the VCEP will wait, after taking request lines inactive and before requesting service again.

The Burst Count Register contains two 8-bit burst count values, one for the CPU bus and one for the Image bus. The burst count specifies the maximum number of clock cycles the VCEP will hold the request lines active and is given by:

(Burst Count Value 1) x 4

The counter starts decrementing at the beginning of the first access, to or from the FIFO Data Port, in either fly-by or flow-through mode. Once the burst counter expires all active requests associated with a particular bus will remain active until the beginning of the next transfer on that bus. At that time they will be taken inactive (see Figure 24).



Note: $\overline{DRQ}$ may be either $\overline{DRQ1}$ or $\overline{DRQ2}$; $\overline{DACK}$ then refers to $\overline{DACK1}$ or $\overline{DACK2}$, respectively.

10487A-27A
WF026550

Figure 24. Bus Burst and Dwell Timing

A burst count of one will ensure that the VCEP only does a single transfer each time it accesses the bus. If the burst counter is set to zero it is disabled, effectively allowing an unlimited burst time.

When all requests associated with a particular bus are taken inactive before the burst counter expires (because the source or destination buffers have been filled or emptied), the VCEP will take its request line inactive and give up the bus. In the case where the burst counter expires while a source buffer request is active, the system may have given the complete (coded or uncoded) image to the VCEP, but since the burst counter has expired, the VCEP is waiting for one more transfer before removing its data transfer request. In this case a software Reset is necessary, making all request lines inactive.

If the burst count is not set to zero, the burst counter may expire before all requests have been completely serviced. The source buffer requests service when there are at least eight empty locations in the source buffer. The destination buffer requests service when at least eight words are in the destination buffer or the buffer is being emptied at the end of a line (single-line mode) or page. When the burst counter expires prior to completely servicing these requests, active requests are removed for the duration of the dwell count and are then reactivated.

The Dwell-Offset Register contains two 6-bit count values, one for each bus, and a 4-bit left-offset value applied to the first word of each scan line. The dwell count specifies the minimum number of clock cycles to wait (from the end of a burst cycle), before the VCEP will reactivate service requests ($\overline{DREQ1}$ and/or $\overline{DREQ2}$). The minimum dwell time is given by:

(Dwell Count Value) x 8

The dwell counter starts at the end of the last bus access of a burst, to or from the FIFO Data Port (see Figure 24). Setting the dwell count to zero will result in a minimum dwell time. For both burst and dwell counts, the actual times the VCEP requests data transfer may vary due to synchronization of the data strobe ($\overline{DS}$ or $\overline{IDS}$) and data acknowledge ($\overline{DACK1}$ or $\overline{DACK2}$) signals to the VCEP's input clock.

GUIDE TO VCEP THROUGHPUT CALCULATIONS

VCEP throughput in compression or expansion is dependent on the specific data input. The stated throughput of 50 Mb/s is based on the average throughput achieved over the CCITT standard documents numbers 4 and 7, when scanned at 200 dpi; these are the most complex of the eight standard documents, with the lowest compression ratio and therefore the lowest throughput on the VCEP. The other six CCITT documents should achieve higher throughput, as would documents scanned at higher resolutions, since they will compress better.

The most specific information on throughput may be obtained by analysis of a given image file. The VCEP, whether compressing or expanding, processes one run length of up to 16 bits in three clock cycles. If the maximum clock rate of 20 MHz is used, one run length up to 16 bits will be processed every 150 ns. This means that if an image consisting of alternating black and white pixels is compressed by the VCEP, it will be processed at one bit per 150 ns, or 6.67 Mb/s. At the other extreme, an all-white image will be processed at 16 bits per 150 ns, or 106.7 Mb/s. Typical documents rarely contain alternating run lengths of unit length. While each document is unique, our analysis shows that the worst sections of the CCITT documents contain five run lengths per 16 bits. In this practical worst-case condition the VCEP would still throughput data in excess of 20 Mb/s.

The above discussion centers on the VCEP's engine speed, but other factors must be taken into account, including the smoothing effect of the source and destination buffers which each hold sixteen words.

The most important issue affecting throughput will often be the handling of raw image data. Assuming a dual-bus VCEP system in expansion mode (for optimum throughput), image data must be removed by the system from the VCEP's destination buffer at a rate at least equal to the engine speed to provide maximum system throughput. This area will be addressed below.

To ensure maximum throughput, a dual-bus system is required with the code buffer on the CPU bus and the image buffer on the Image bus. Also, certain settings of VCEP programmable parameters are necessary. The bus burst value should be maximum especially on the Image bus, and dwell values should be minimum.

The above throughput numbers apply to compression and expansion when in MMR mode. Since MMR coding is the most complex, MR and MH code compression or expansion will yield higher throughput. Also, throughput in Transparent mode will be 16 bits per three clocks.

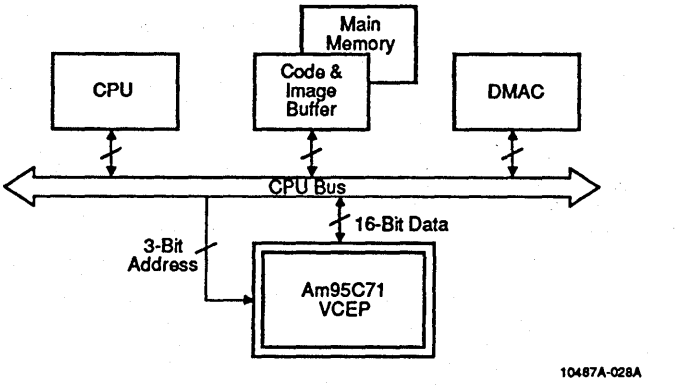


Figure 25. Single-Bus Configuration

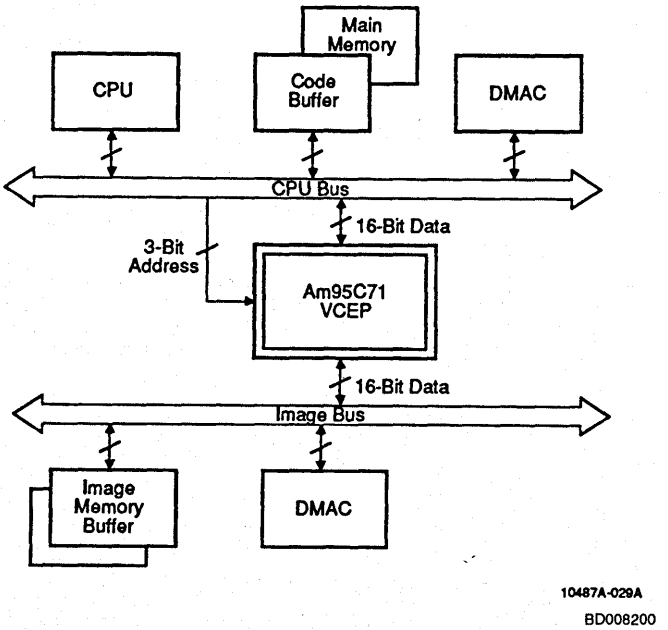


Figure 26. Dual-Bus Configuration

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Maximum V_{CC} Relative to V_{SS} ..... −0.3 to +7.0 V
DC Voltage Applied to Any Pin
Relative to V_{SS} ..... −0.5 to V_{CC} + 0.3 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IL}	Input LOW Voltage		−0.5	+ 0.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + 0.5	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA		+ 0.4	V
V_{OH}	Output HIGH Voltage	I_{OH} = 250 µA	2.4		V
I_{LI}	Input Leakage Current	0 < V_{IN} < V_{CC}		±10	µA
I_{LO}	Output Leakage Current	0.45 < V_{IN} < V_{CC}		±10	µA
I_{CC}	Power Supply Current			250	mA

PRELIMINARY

CAPACITANCE

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
C_{IN}	Input Pin Capacitance	f_C = 1 MHz		10	pF
C_{OUT}	Output Pin Capacitance			15	pF
$C_{I/O}$	I/O Buffer Capacitance			20	pF

SWITCHING CHARACTERISTICS over operating range (T_A = 0 to +70°C, V_{CC} = 5 V ± 5%)

No.	Parameter Symbol	Parameter Description	95C71-20		95C71-16		Unit
			Min.	Max.	Min.	Max.	
1	t_{PD}	Clock Period	50	500	62	500	ns
2	t_{PWH}	Clock HIGH Time	23		28		ns
3	t_{PWL}	Clock LOW Time	23		28		ns
4	t_R	Clock Rising Time		5		5	ns
5	t_F	Clock Falling Time		5		5	ns
6	t_S	Address Valid to $\overline{DS}$ FE Setup Time	10		12		ns
7	t_S	$\overline{CS}$ Valid to $\overline{DS}$ FE Setup Time	0		0		ns
8	t_S	R/$\overline{W}$ Valid to $\overline{DS}$ FE Setup Time	10		12		ns
9	t_{SKEW}	$\overline{DS}$ FE to Data Output Valid Delay		80		85	ns
10	t_{SKEW}	$\overline{DS}$ FE to $\overline{DRQ}$ RE Delay		60		70	ns
11	t_H	$\overline{DS}$ RE to Data Output Hold Time	0		0		ns
12	t_{SKEW}	$\overline{DS}$ RE to Data Out Float Delay		50		55	ns
13	t_H	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time	0		0		ns
14	t_H	$\overline{DS}$ RE to $\overline{CS}$ Valid Hold Time	0		0		ns
15	t_H	$\overline{DS}$ RE to Address Valid Hold Time	0		0		ns
16	t_{PWL}	$\overline{DS}$ LOW Width (Note 2)	70		90		ns
17	t_S	Data In Valid to $\overline{DS}$ RE Setup Time	50		60		ns
18	t_H	$\overline{DS}$ RE to Data Valid Hold Time	0		0		ns
19	t_{PWH}	$\overline{CS}$ HIGH Width	65		70		ns
20	t_S	$\overline{DACK}$ FE to $\overline{DS}$ FE Setup Time	0		0		ns
21	t_{PWH}	$\overline{DS}$/$\overline{DACK}$ HIGH Width	65		70		ns
22	t_{SKEW}	$\overline{DS}$ RE to $\overline{INTR}$ RE Delay Time		100		100	ns
23	t_{PWL}	$\overline{RESET}$ LOW Width	(Note 1)		(Note 1)		ns

PRELIMINARY

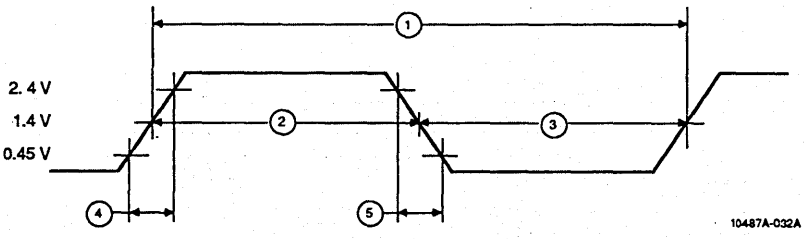
Notes: 1. Minimum $\overline{RESET}$ LOW Width is four clock periods (see parameter 1).
2. No more than one complete data transfer cycle can occur in each VCEP clock cycle. One data transfer may occur on each bus of a dual-bus system since the buses are fully independent. In addition, the maximum low pulse of $\overline{DS}$ cannot exceed 15 VCEP clock cycles.

SWITCHING WAVEFORMS

Key to Switching Waveforms

WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

KS000010

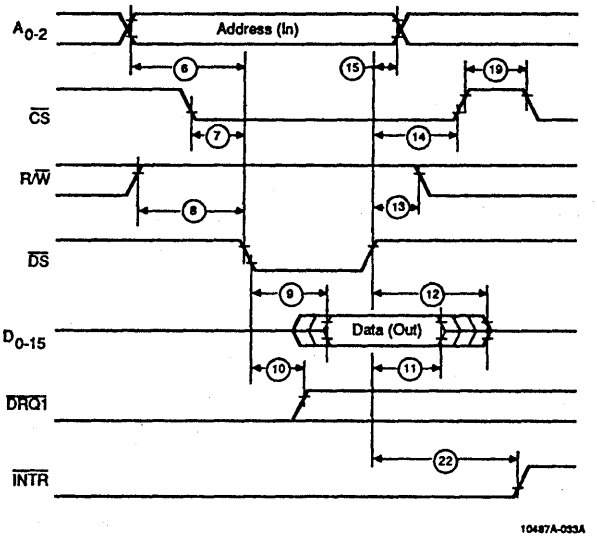


WF026560

Clock Timing

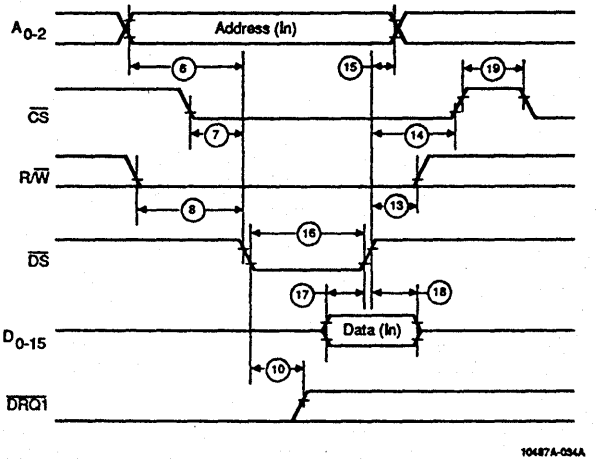
5

SWITCHING WAVEFORMS (Cont'd.)



WF026570

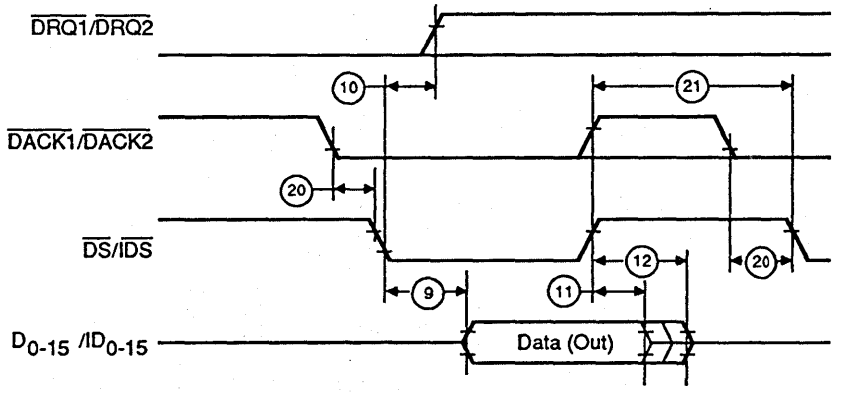
CPU Program/DMA Flow-Through Mode Read Timing



WF027010

CPU Program/DMA Flow-Through Mode Write Timing

SWITCHING WAVEFORMS (Cont'd.)

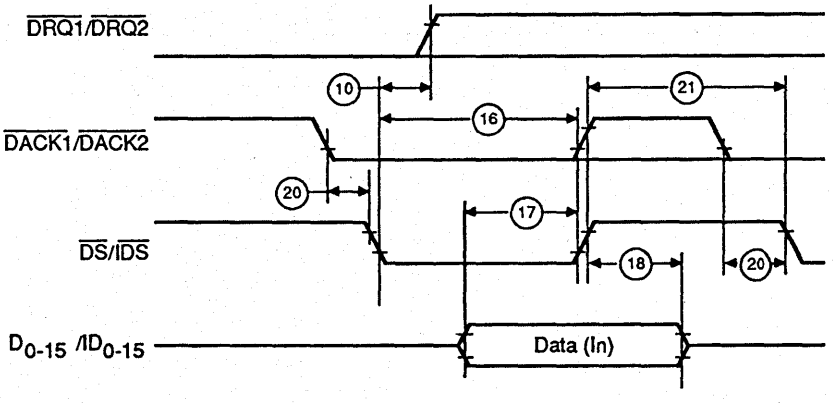


Note: Parameters 11, 12, and 21 are with respect to DS/IDS or DACK1/DACK2 inactive, whichever is sooner.

10487A-035A

WF026590

DMA Fly-by Mode Read Timing



Note: Parameters 17, 18, and 21 are with respect to DS/IDS or DACK1/DACK2 inactive, whichever is sooner.

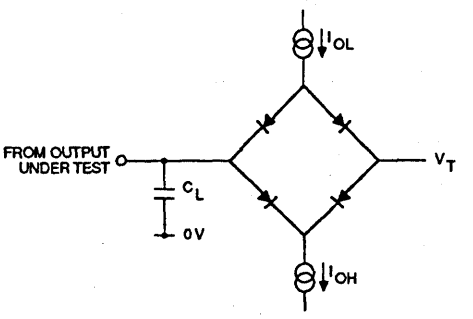
10487A-036A

WF026600

DMA Fly-by Mode Write Timing

5

SWITCHING TEST CIRCUIT

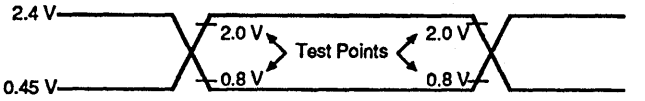


AF004810

Notes: 1. C_L = 100 pF (includes text fixture capacitance).

SWITCHING TEST WAVEFORM

(Input)



10487A-030A
WF026611

Preliminary

Am95C75

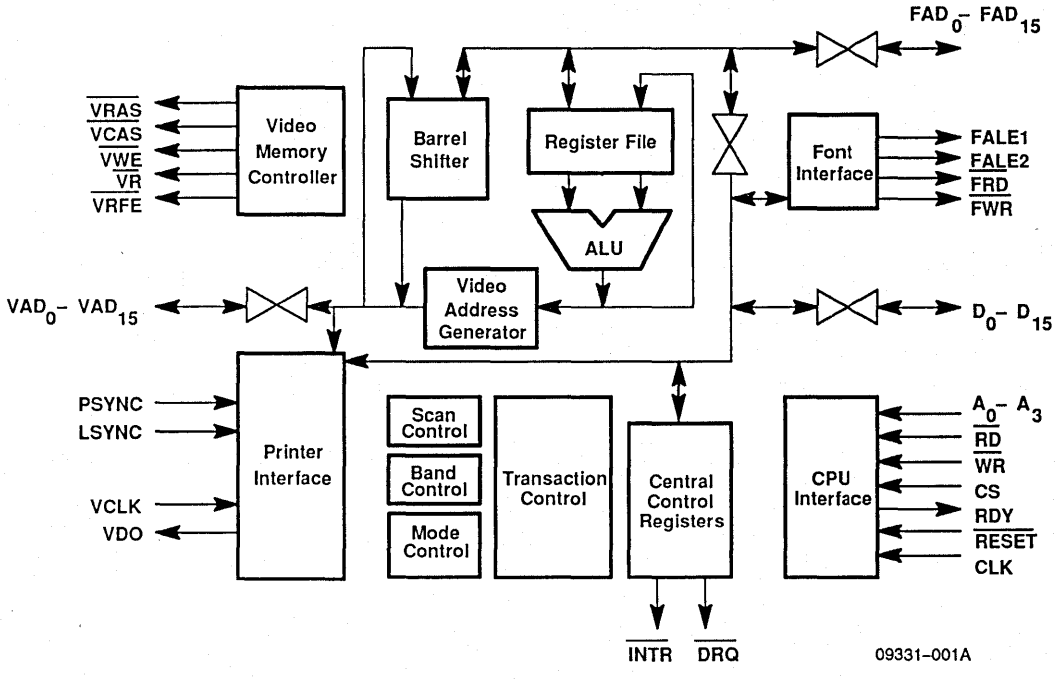
Raster Printer Controller (RPC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- 20 Mb/s printing of combined text and graphics
- Triple-bus architecture for independent host, font memory and page buffer interfaces and direct connection to print engines
- Supports any combination of transparent, opaque and textured images at pixel addressable positioning
- 24-bit addressing provides for page buffers up to 16K x 16K pixels and 16M words of font memory space
- Supports two band buffers with automatic virtual address conversion to reduce memory requirements
- On-board programmable DRAM controller with refresh
- Synchronous or asynchronous interface to print engines with a broad range of page size and margins
- Operating modes for direct interface to Am95C76 Orthogonal Rotation Processor (ORP)

BLOCK DIAGRAM



5

GENERAL DESCRIPTION

The Am95C75 Raster Printer Controller (RPC) is a high-performance CMOS processor for controlling the real-time requirements of a raster printing system. It is designed to assemble text and graphical images into a partial or full page buffer from either of two memory spaces or from the host and serially transmit that image information to an asynchronous print engine. A high degree of programmable options assures simple interfacing to a broad range of host, memory, and printer configurations.

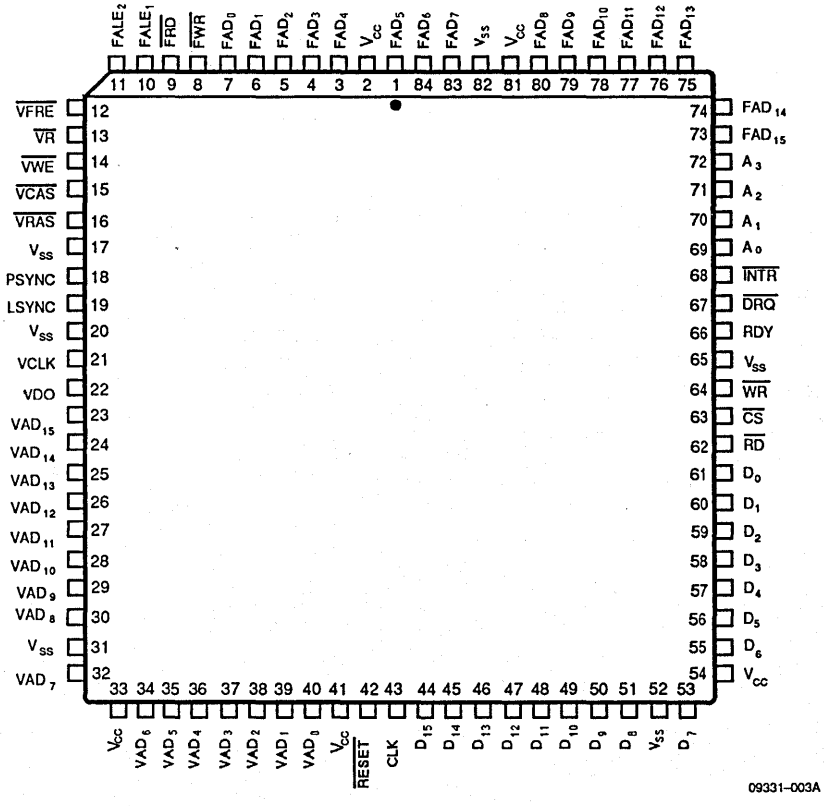
The RPC receives source and destination address information (or image data) from the host CPU or DMA controller, and performs block transfers of data to rectangular areas at the destination address in the buffer. The image blocks are of arbitrary size and pixel alignment and may be combined with background information using the overlay options. Characters may be textured using the additional source address of a texture word or array. In addition to the buffer assembly operations, the RPC can transfer font or graphic information with the host to or from the static RAM or ROM Font memory or the DRAM Video memory of the buffer.Two other modes support reading and writing of the Am95C76 Orthogonal Rotation Processor (ORP) which resides in Font memory space.

Printing can begin when a page or band is completely assembled and is controlled by Page and Line Sync signals provided by the print engine. Pixel data is serialized at a rate controlled by the Video clock (VCLK). Page Size and Margins are determined by the values programmed in the RPC control registers. In systems where two band buffers are used, one band may be assembled while the other band is being scanned out. The RPC manages all address conversion so that the host need only provide a destination address for each image block that corresponds to the virtual position of that block on the full page. Band control logic in the RPC insures that the bands alternate properly according to the scanning requirements of the printer. Image blocks or fonts that are sliced by a band boundary cause interrupts that allow the remaining portion to be transferred into the next band.

The system bus interface of the RPC allows the control registers to reside in host I/O or memory space and operations are initiated automatically when the address information for the transfer is provided. An internal address pointer sequences through the control registers required to setup a block transfer so that the host may write them to a single port address if that is desired. The RPC will interrupt the CPU or request a DMA when the transfer is complete. Interrupt options allow the host to be updated on transfer, printing, and error conditions by reading status information.

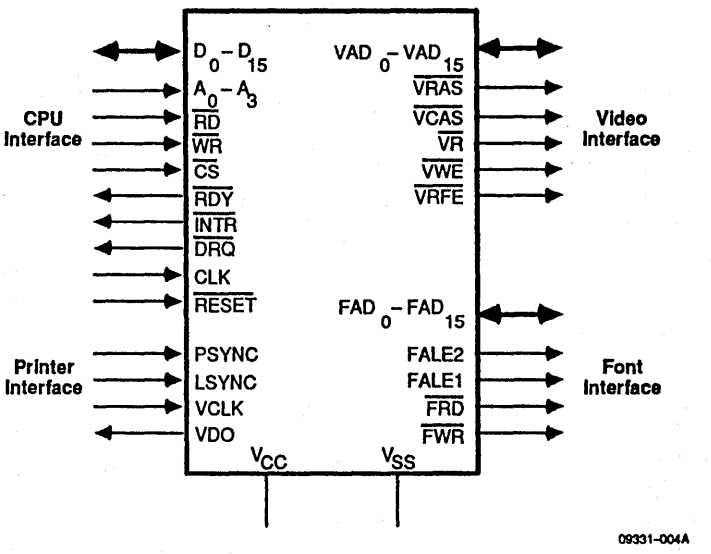
CONNECTION DIAGRAM

PLCC



09331-003A

LOGIC SYMBOL

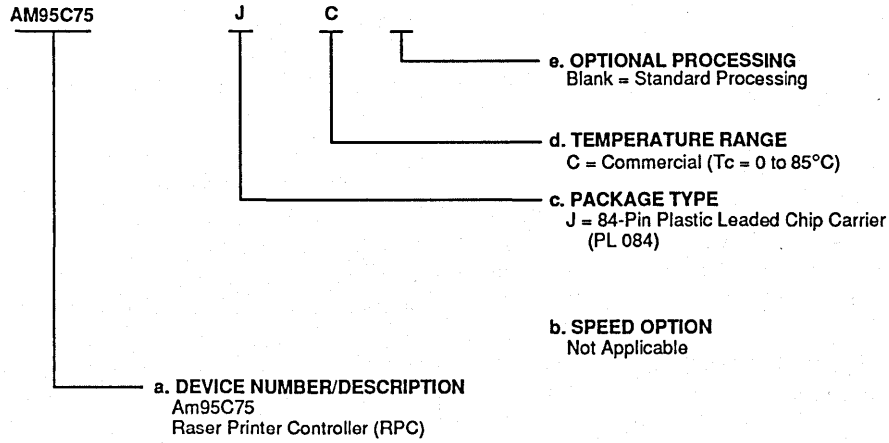


ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM95C75	JC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

5

PIN DESCRIPTION

Interface Signal Description

The RPC has four independent groups of pins to interface to the other resources of the printing system. The CPU Interface connects to the host to provide control of all operations. The Font Memory Interface connects to an optional static memory of RAM or ROM for storage of character fonts, textures and other blocks of data. The Font Memory Interface also interfaces directly to the Am95C76 ORP. The Video Memory Interface connects to standard Dynamic RAMs for the page buffer or for storage of downloaded fonts or other data. This latter case permits a system with a single memory architecture. The Printer Interface accepts timing information from the print engine and supplies serial pixel data for the rasterized page.

CPU Interface Signals

A_0–A_3
Address Bus (Input)

The 4-bit Address Bus is used to select one of sixteen internal registers for reading or writing. It must be held valid while the access is taking place.

CLK
Clock (Input)

The Clock input provides the timing reference for all operations controlled by the RPC except printing. All memory operations on the Font or Video Interfaces will occur relative to Clock edges. Accesses on the CPU Interface may be asynchronous to the Clock.

$\overline{CS}$
Chip Select (Input; Actve LOW)

Chip Select must be held LOW to qualify a read or write access by the host. Chip Select may be held LOW for multiple accesses.

D_0–D_{15}
Data Bus (Input/Output; Three-State)

The 16-bit bidirectional Data Bus is used for all read and write accesses by the host system. The bus is an output whenever $\overline{CS}$ and $\overline{RD}$ are active and is an input whenever $\overline{CS}$ and $\overline{WR}$ are active.

$\overline{DRQ}$
DMA Request (Output; Active LOW)

The Data Request output is driven LOW when the RPC is ready to accept the address information required to transfer an image block and is driven HIGH when the last word of address information is received. Five words are required for a normal Dispatch operation and eight words are required for a textured Dispatch operation. Data Request will only be driven LOW when the RPC is in Dispatch, ORP Load or ORP Read Mode. It is intended for direct connection to a DMA controller.

$\overline{INTR}$
Interrupt (Output; Active LOW)

The Interrupt output is driven LOW whenever the RPC detects a condition that requires the attention of the CPU and the Interrupt Enable bit is set. Interrupt is driven HIGH when status is read by the CPU to identify the cause of the interrupt. All interrupt status bits are cleared after the register is read.

$\overline{RD}$
Read (Input; Active LOW)

The active-LOW Read input determines when the contents of the selected internal register are driven onto the Data Bus.

RDY
Ready (Output)

The Ready output is driven LOW at the start of a Read or Write access when the data is not immediately available or cannot be accepted. The host must insert wait states and hold Read or Write active until Ready goes HIGH to complete the access successfully. If Read or Write go HIGH while Ready is being driven LOW, the access is aborted and Ready will go HIGH. In this abort case, there is no guarantee that the access was accepted or ignored.

$\overline{RESET}$
Reset (Input; Actve LOW)

The active-LOW $\overline{RESET}$ input causes the RPC to terminate any operation in progress and enter the idle state. $\overline{RESET}$ must be held LOW for a minimum of four CLK cycles. After Reset, all bidirectional buses will be in a high impedance state and all internal registers must be programmed to begin operation.

$\overline{WR}$
Write (Input; Active LOW)

The active-LOW Write input determines when the contents of the Data Bus are loaded into the selected internal register.

Font Memory Interface Signals

FAD_0–FAD_{15}

Font Address/Data Bus (Input/Output; Three-State)

The 16-bit bidirectional Font Address/Data Bus carries all address and data information for Font Memory accesses. The presence of information on this bus is always controlled by the signals described below.

FALE1

Font Address Latch Enable 1 (Output)

The FALE1 output is driven HIGH when the least significant 16 bits of the 24-bit Font Memory address are being driven onto FAD_0–FAD_{15}. This address information must be latched on the HIGH-to-LOW transition of FALE1. The lower address cycle, containing FALE1, occurs on every memory cycle.

FALE2

Font Address Latch Enable 2 (Output)

The FALE2 output is driven HIGH when the most significant 8 bits of the 24-bit Font Memory address are being driven onto FAD_0–FAD_7. This address information must be latched on the HIGH-to-LOW transition of FALE2. The upper address cycle, containing FALE2, is sometimes skipped when the upper address is unchanged from the previous memory cycle.

$\overline{FRD}$

$\overline{\text{Font Read}}$ (Output; Active LOW)

The active-LOW Font Read output is driven LOW during a Font Memory access when data is to be driven onto the FAD_0–FAD_{15} bus by the memory. $\overline{FRD}$ stays low for 1 to 8 CLK cycles depending on the number of Font Memory Wait states programmed.

$\overline{FWR}$

$\overline{\text{Font Write}}$ (Output; Active LOW)

The active-LOW Font Write output is driven LOW when valid data is being driven onto the FAD_0–FAD_{15} bus by the RPC to be strobed into the Font Memory. $\overline{FWR}$ stays low for 1 to 8 CLK cycles depending on the number of Font Memory Wait states programmed. The FAD data out is valid for the duration of $\overline{FWR}$.

Video Memory Interface Signals

VAD_0–VAD_{15}

Video Address/Data Bus (Input/Output; Three-State)

The 16-bit bidirectional Video Address/Data Bus carries all address and data information for Video Memory accesses. The presence of information on this bus is always controlled by the signals described below.

$\overline{VCAS}$

$\overline{\text{Video Column Address Strobe}}$ (Output; Active LOW)

The $\overline{VCAS}$ output is driven LOW to indicate that a valid column address is being output on the VAD_0–VAD_{15} bus. The HIGH-to-LOW transition of $\overline{VCAS}$ is normally used to strobe the column address into dynamic RAMs. The column address cycle follows a row address cycle except during memory refresh operations.

$\overline{VR}$

$\overline{\text{Video Read}}$ (Output; Active LOW)

The $\overline{VR}$ output is driven LOW to indicate that external VAD bus transceivers should drive Video Memory data onto the VAD_0–VAD_{15} bus to be input by the RPC. This signal can be used to avoid bus contention between the Video Memory address being driven out and the memory data being received by the RPC.

$\overline{VRAS}$

$\overline{\text{Video Row Address Strobe}}$ (Output; Active LOW)

The $\overline{VRAS}$ output is driven LOW to indicate that a valid row address is being output on the VAD_0–VAD_{15} bus. The HIGH-to-LOW transition of $\overline{VRAS}$ is normally used to strobe the row address into dynamic RAMs.

$\overline{VRFE}$

$\overline{\text{Video Refresh Enable}}$ (Output; Active LOW)

The $\overline{VRFE}$ output is driven LOW for the duration of a Video Memory Refresh operation. This signal can be used to control the bank select decoder so that all memory chips receive $\overline{RAS}$ on the refresh row address cycle.

$\overline{VWE}$

$\overline{\text{Video Write Enable}}$ (Output; Active LOW)

The $\overline{VWE}$ output is driven LOW to indicate that valid data is being output on the VAD_0–VAD_{15} bus. The HIGH-to-LOW transition of $\overline{VWE}$ is normally used to strobe data into dynamic RAMs.

Printer Interface Signals

LSYNC

Line Sync (Input)

The LOW-to-HIGH transition of the LSYNC input is used to start a scan line sequence that may consist of a decrement of the Y-Margin count or the start of the X-Margin control process prior to the scan out of serial pixel data for a line. LSYNC will only be recognized if the conditions for a PSYNC have been met.

PSYNC

Page Sync (Input)

The LOW-to-HIGH transition of the PSYNC input is used to start the Y-Margin control process if the RPC has been initialized to scan out the serial pixel information of a page on the VDO output. PSYNC will be recognized if the RC bit in the Mode Register is set and the beginning of the first scan line has been loaded in the the RPC shift register. Otherwise PSYNC will be ignored.

VCLK

Video Clock (Input)

The VCLK input determines the data rate for serialization of pixel data on the VDO output during scan out. If the Printer Interface is being operated in Synchronous Mode, the serial data rate equals the VCLK rate. In Asynchronous Mode the data rate equals the VCLK rate divided by four. The mode determines whether LSYNC must be synchronous to VCLK in addition to the VCLK division control. PSYNC has no timing requirement related to VCLK in either mode.

VDO

Video Data Out (Output)

The VDO output provides the serial pixel data that is used by the print engine to scan out the page. When the Printer Interface is inactive or during the X or Y Margin time, the VDO output remains HIGH. VDO may toggle at the pixel rate during the active page area and the output polarity is selected by the VP bit in the Mode Register.

FUNCTIONAL DESCRIPTION

Register Description

The RPC contains 16 user-addressable registers that may be accessed by the CPU Interface. Each register has a unique location as selected by the A_0–A_3 address pins. These registers may be grouped into the Temporary Register, Operation Control Registers, Source Address Registers, Texture Address Registers and Destination Address Registers according to Table 1.

Table 1. RPC Registers

	Port Address	Name
	0	Temporary Register (Temp)
Operation Control Registers:	1	Mode/Status Register (Mode)
	2	Margin Register (XMGR, YMGR)
	3	Page X Size Register (PXSR)
	4	Page Y Size Register (PYSR)
	5	Video Band Boundary Register (VBBR)
	6	Video Memory Refresh Rate Register (VMRR)
	7	Memory Timing Register (MTR)
Source Address Registers:	8	Source Address Register 1 (SAR1)
	9	Source Address Register 2 (SAR2)
	10	Source Address Register 3 (SAR3)
Texture Address Registers:	11	Texture Address Register 1 (TAR1)
	12	Texture Address Register 2 (TAR2)
	13	Texture Address Register 3 (TAR3)
Destination Address Registers:	14	Destination Address Register 1 (DAR1)
	15	Destination Address Register 2 (DAR2)

In order to minimize the number of registers that must be programmed to begin RPC operations, some registers have unrelated bytes or fields packed together. Some of the address registers contain control fields that specify options for the address or the related operations.

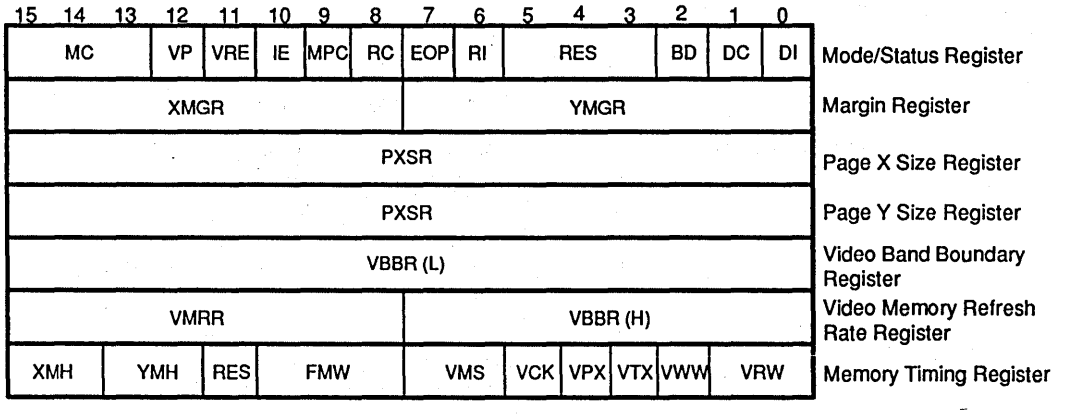
For Dispatch operations, the address registers may be loaded using the automatic sequencing feature provided by the RPC. This allows all words to be written to the Temporary Register at Port 0 in the proper order, and be loaded into the correct location. Repetitive operations may be programmed by the host without complex port address changes using this feature. The Texture Address Registers are automatically skipped if texturing is not selected, so only five words must be loaded for each non-textured Dispatch.

The Temporary Register, at Port 0, serves as a holding register for data in some RPC operations. In Font Load, Font Read and Graphics Load modes, the TEMP holds data for transfer between the host and the Font or Video Memory Interfaces. Since the host and the RPC internal controller must share access, the TEMP may not always be ready for the host, and the RDY pin will be driven inactive until the internal operation is complete.

In Dispatch and ORP Modes, the TEMP location can be used for writing values to the SAR, TAR and DAR words. These words must be written in the proper order to be loaded correctly. When the last word (DAR2) is written to Port 0, the operation is initiated. The next word written to Port 0 (after $\overline{DRQ}$ goes active) will be assumed to be SAR1 for the start of the next operation. These address registers may also be read or written using their explicit port addresses.

Operation Control Registers

The Operation Control Registers are normally initialized once after the power-up and $\overline{\text{RESET}}$ is applied, and with the exception of the Mode Register, are not changed frequently during the operation of the RPC. These registers provide the system configuration information for the printer and memory interfaces. See Figure 1 for the placement of control bits and fields in these registers.

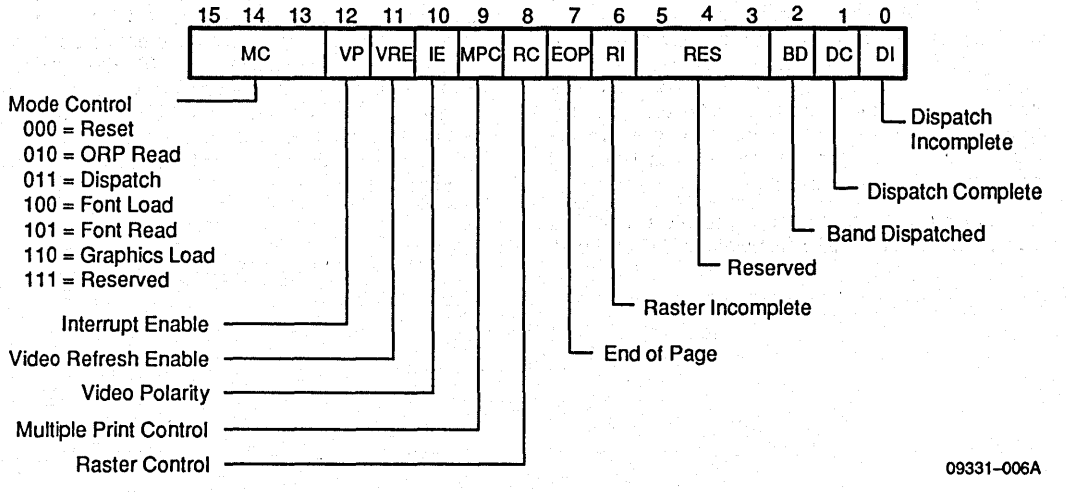


09331-005A

Figure 1. Operation Control Registers

Mode/Status Register

The upper byte of the Mode/Status Register allows the operation type and various options to be selected and the lower byte provides a read-only access to Status flags (see Figure 2). The Status may be read at any time during RPC operations and special provision has been made to optimize the response of the Ready pin for accesses of this register. Flag bits that are active will be cleared after each read access to insure that each read accurately reflects the current Status. Writes to the Mode Register may not be recognized until the RPC has completed the current operation, however, changing certain bits in the Mode may cause unexpected results if care is not taken to allow printing operations to complete.



09331-006A

Figure 2. Mode/Status Register

Bits 13, 14 and 15 of the Mode Register are the Mode Control field and are used to select the operation type according to Table 2.

Table 2. Mode Control Field

Bit	15	14	13	Operation Type
	0	0	0	Software Reset
	0	0	1	ORP Read
	0	1	0	ORP Load
	0	1	1	Dispatch
	1	0	0	Font Load
	1	0	1	Font Read
	1	1	0	Graphics Load
	1	1	1	Reserved

The Software Reset code is executed as a command when it is written to the Mode Register to clear an existing operating mode. This causes any current operation to be terminated (except Dynamic RAM Refresh). Certain internal scratch registers are initialized and the band control logic is cleared to Band 0. A Software Reset is normally issued after each full page is printed. Band-buffer systems may be setup to alternate bands continuously as subsequent pages are printed and then reset is not required.

The RPC maintains current page and reference registers to be used in address calculations. Software Reset, as well as activating the $\overline{\text{RESET}}$ pin, causes these values to be cleared so that the next sequence of Dispatching and Scan-out begins at the top of Video Memory space. Software Reset must be issued after each page is printed to control the limit of physical memory space in a single full-page buffer system. It does not alter the contents of any of the programmable address registers.

Video Polarity is selected by bit 12 of the Mode Register. This bit is reset to zero to specify that a HIGH level on the Video Data Out (VDO) pin indicates a black image and a LOW level indicates a white image. VP is set to a one to specify that a HIGH level indicates white and a LOW indicates black. Regardless of the setting of the VP bit, the VDO pin is driven HIGH when the printer is outside the printable area defined by the page size, or when printing is idle.

Bit 11 of the Mode Register is the Video Refresh Enable bit. It is cleared when the $\overline{\text{RESET}}$ pin is activated or by writing to the Mode Register with a zero in that bit. The Video Refresh Rate Register should be initialized before VRE is set. If VRE is cleared by writing the Mode Register, only a Refresh cycle currently in progress would be completed.

Bit 10 of the Mode Register is the Interrupt Enable bit. In order for the RPC to activate the Interrupt pin, the IE bit must be set. Clearing IE will cause $\overline{\text{INTR}}$ to go inactive if an interrupt was currently pending.

The Multiple Print Control is bit 9 of the Mode Register. This bit is zero in the normal case where each page is printed once. The RPC normally performs read-modify-write cycles on the Video Memory for scanning out the pixel data to be printed. This allows "white" to be written back to each memory word in order to erase each page in preparation for the next. If MPC is set, a memory read cycle is used to allow the same page to be scanned out multiple times. The MPC bit must be cleared before the last printing of a page to clear the buffer. MPC can only be used in systems with a full-page buffer. Writing the Mode Register to set MPC and RC should not be done while Dispatching is in progress. MPC should be set at the beginning of page assembly or some time before RC to ensure that it is recognized the first time the page is printed. When the video bands are configured as two full page buffers, MPC should must be set before the dispatch of the first page or before the dispatch of the second page.

Printing is enabled by setting the Raster Control, bit 8 of the Mode Register. RC should not be set until at least the first band of the page has been Dispatched so that the first words of the first scan line can immediately be fetched for loading into the shift register. Once this pixel data is ready to be serialized onto the VDO pin, the RPC can recognize a PSYNC input. Otherwise, PSYNC and LSYNC are ignored.

RC is cleared by the RPC when the printing of a page is complete. RC is also cleared by a Raster Incomplete error condition. If RC is cleared by writing the Mode or the $\overline{\text{RESET}}$ pin, a printing operation in progress will be terminated and cannot be restarted mid-page. Setting RC by writing the Mode will not take effect if the EOP flag is set. This allows Mode changes during printing without erroneously starting a new page.

The low byte of the Mode Register contains status information that cannot be modified by a write access, unless Software Reset is selected in the Mode Control field. This will clear all status flags and deactivate any interrupt condition that was pending.

End-of-Page is indicated by bit 7 of the Status byte. This occurs when the printing of a page is complete as determined by the X and Y Margin and Size values that were programmed into the RPC. EOP causes the Interrupt pin to be activated and is cleared when the Status Register is read.

Bit 6 of the Status byte indicates a Raster Incomplete error condition has been detected. This occurs when the LSYNC input is activated and the current scan line has not been completely serialized to the printer as programmed in the Page X Size Register. This condition cannot be recovered from and may be due to incorrectly programmed X Size or insufficient VCLK frequency. RI causes an interrupt and clears RC.

Bit 2 of the Status byte is the Band Dispatched flag. This flag is activated when the destination of a Dispatch operation equals or exceeds the Video Band Boundary of the current band. The RPC will not dispatch this character but rather, will interrupt to provide an opportunity to

eration equals or exceeds the Video Band Boundary of the current band. The RPC will not dispatch this character but rather, will interrupt to provide an opportunity to dispatch any partial characters to the next band first. A second Dispatch operation (of the same or different character) into the next band will cause the RPC to complete the dispatch. See the section on Video Bands for additional information on band control.

Bit 1 of the Status byte is the Dispatch Complete flag. This bit indicates that the last Dispatch or ORP Operation completed transferring the entire image block. This condition does not cause an interrupt and is intended for use in systems that poll status after each Dispatch to determine when the next operation may be started. When DC is set the RPC will also activate $\overline{DRQ}$.

Dispatch Incomplete is indicated by bit 0 of the Status byte. This occurs when the image block being dispatched extends beyond the Video Band Boundary of the current band. The RPC stops the Dispatch at that point and generates an interrupt. The SAR, TAR and DAR registers for the remaining partial character may be read from the RPC for use when the next band is being dispatched. See Video Bands for more information on handling sliced characters.

Margin Register

The Margin Register, at Port 2, holds the least significant 8 bits of the X Margin in the high byte and the least significant 8 bits of the Y Margin in the low byte. The Memory Timing Register at port 7 holds the two most significant bits for each of the margins. Together, these registers allow up to 1,023 pixels in the X direction and 1,023 lines in the Y direction for margin area that will always be scanned out before the printable area begins.

Page X Size Register

The Page X Size Register is at Port 3. This specifies the number of pixels in the printable area of a scan line, not including the X Margin. A full 16-bit value is supported.

Page Y Size Register

The Page Y Size Register is at Port 4. This specifies the number of lines in the printable area of a page, not including the Y Margin. A full 16-bit value is supported.

Video Band Boundary Register

The 16-bit word at Port 5 plus the low byte of Port 6 comprise the Video Band Boundary. Bit 7 of the byte at Port 6 is the most-significant bit of this 24-bit address. The Video Band Boundary points to the word address of the beginning of the second band in Video Memory space. The beginning of the first band is always 0 and bands must start on a line boundary, which is internally calculated using the PXSR rounded up to a multiple of 16 pixels. For a full-page buffer system, the Video Band Boundary must be set to a value greater than the size of the page.

Video Memory Refresh Rate Register

The register at Port 6 holds the Video Memory Refresh Rate in the high byte and the most-significant 8 bits of the VBBR in the low byte. The VMRR specifies the number of CLK cycles that will elapse between dynamic memory refresh cycles if refreshing is enabled by the VRE bit in the Mode Register. The value programmed will depend on the refresh interval of the memory chips used and the frequency of the CLK input.

Memory Timing Register

The Memory Timing Register at Port 7 contains control information for the Video and Font Memory interfaces, in addition to the two most-significant bits for the X and Y Margins. This register must be programmed before any memory or printer operations are used. The MTR is not cleared by $\overline{RESET}$. See Figure 3 for the bit position assignments of this register.

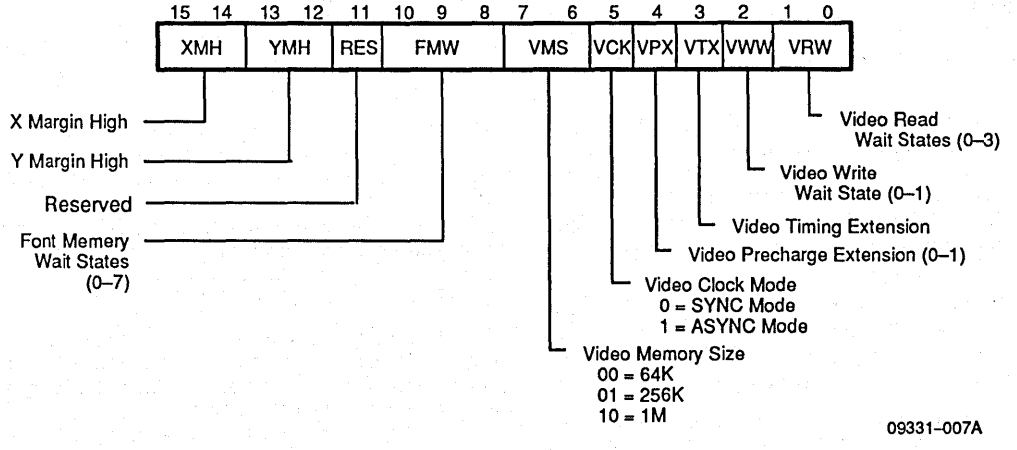


Figure 3. Memory Timing Register

Bits 15 and 14 of the MTR correspond to bits 9 and 8 respectively, of the X Margin value. Bits 13 and 12 correspond to bits 9 and 8 respectively, of the Y Margin value. These bits must be loaded with zeros if the margin values are less than 256.

Bits 10, 9 and 8 form the Font Memory Wait States field of the MTR with bit 10 being the most significant bit. This provides for 0 to 7 wait states to be inserted into every Font Memory cycle. Wait states extend T2 (see Font Memory Timing diagrams) of the cycle to extend $\overline{FRD}$ and $\overline{FWR}$ as required by the static memories. Bits 7 and 6 of the MTR form a 2-bit field that is used to select the Video Memory Size. This is determined by the type of dynamic RAM chips that are used for the Video Memory. The options, 64K, 256K and 1M, result in differences in how the Bank Select, Row address and Column address are output on the VAD bus during address cycles. Table 3 defines the use of the VMS field.

Table 3. VMS Control Field

MTR 7	MTR 6	DRAM	Bank Address	Row/Col Address
0	0	64K	VAD_{15}–VAD_8	VAD_7–VAD_0
0	1	256K	VAD_{15}–VAD_9	VAD_8–VAD_0
1	0	1M	VAD_{15}–VAD_{10}	VAD_9–VAD_0
1	1	Reserved		

Bit 5 of the MTR determines the VCLK mode for printing operations. If this VCK bit is a zero, Synchronous mode is selected. This means that the LSYNC input must change synchronously to the VCLK and meet the setup time specified. In this mode the pixel rate equals the VCLK rate. If the VCK bit is a one, Asynchronous mode is selected and LSYNC does not need to have any timing relationship to VCLK. In Asynchronous mode the VCLK is divided by four to determine the pixel rate.

An important feature of the RPC relates to Asynchronous mode. The internal synchronization of LSYNC can result in some variation in the position of the starting pixel of successive scan lines, relative to VCLK. In order to minimize this variation, the RPC attempts to detect the rising edge of LSYNC during both phases of VCLK and starts the VCLK divider on the phase it is detected. This limits the variation to one eighth of a pixel due to the divide by four of the VCLK rate. A VCLK frequency of up to 32 MHz may be used in this Asynchronous mode.

The Video Precharge Extension option is selected by bit 4 of the MTR. When VPX is a zero, the minimum dynamic RAM precharge time, when $\overline{VRAS}$ and $\overline{VCAS}$ are inactive, is nominally 1.5 CLKs. (See AC Timing Specifications to determine exact values.) If VPX is set to one, an additional CLK cycle is inserted into the minimum precharge time. This may be required for some RAMs.

An optional Video Timing Extension cycle is selected by bit 3 of the MTR. When VTX is set to one, an additional CLK cycle is inserted after the Row address cycle of each memory access. During this cycle the Row address is held valid on the VAD bus and the falling edge of $\overline{VCAS}$ is delayed to the next CLK cycle. This may be used to provide additional time for bank select decoders. The VTX bit does not affect the timing of memory Refresh cycles.

Bit 2 of the MTR is the Video Write Wait State bit. When set, it allows one wait state to be inserted to extend T3 of a Video Memory Write cycle and to extend T5 of a Read-Modify-Write cycle. This extends $\overline{VWE}$ and VAD Data Out by one CLK. This may be required for some RAMs.

Bit 1 and 0 of the MTR form the Video Read Wait States field. The VRW allows 0 to 3 wait states to be inserted to extend T2 of Video Memory Read and Read-Modify-Write cycles. This extends $\overline{VR}$ and delays the requirement for valid Data In on the VAD bus. Bit 1 is the most-significant bit of this field that is encoded to correspond to the number of waits desired.

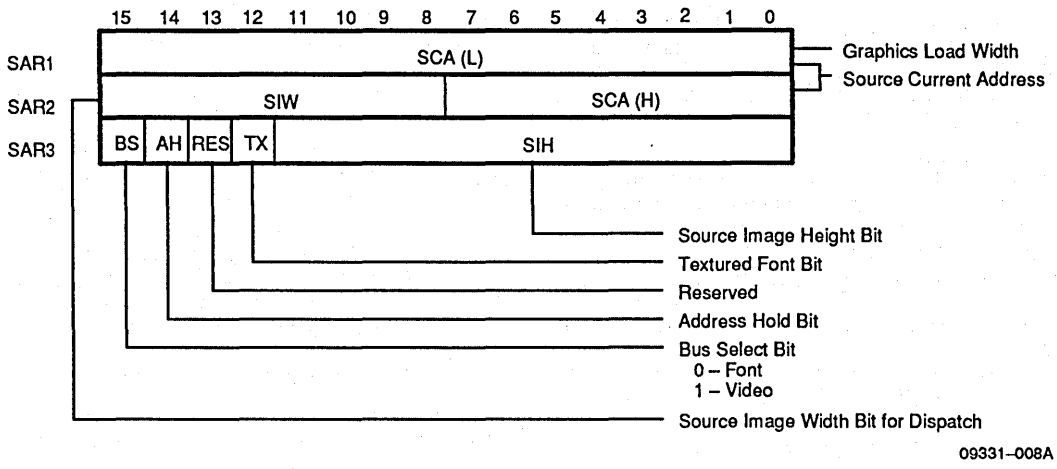


Figure 4. Source Address Register

Source Address Registers

The Source Address Registers provide information for Dispatch operations. This information is assumed to be correct when the Dispatch is initiated by loading DAR2. Additionally, SAR1 is used for the width in a Graphics Load operation.

The first word of the Source Address Registers holds the lower 16 bits of the current Source Address (Figure 4). The upper 8 bits of this 24-bit address are held in the low byte of SAR2. Since the Source Address is updated as an operation progresses, reads of SAR1 or SAR2 may not return the original value. This is essential for handling sliced characters. The high byte of SAR2 holds the value of the Source Image Width in words for a Dispatch. This allows images from 16 to 4,096 pixels wide. SAR1 must be loaded with the width in words for a Graphics Load operation wherein the address is contained in the DAR. The width value does not change during an operation.

The lower 12 bits of SAR3 hold the Source Image Height, which can be from 1 to 4,096 scan lines. The height value is updated like the address during an operation to handle sliced characters. The remaining upper bits of SAR3 hold control information.

Bit 15 of SAR3 is the Bus Select bit for the source of the character in a Dispatch operation. When this bit is a zero, the character is loaded from the Font Memory. When it is a one, the character is loaded from the Video Memory.

Bit 14 of SAR3 is the Address Hold bit. When this bit is set, the source address for a Dispatch operation is not incremented. This allows a single word to be transferred repeatedly to create a block at the destination that is filled with a pattern, or allows a source to be read from a fixed location like the ORP.

Bit 12 of SAR3 is the Textured Font bit (TX). When this bit is set, a Textured Dispatch operation is performed. The Texture Address Registers must be loaded with valid information. The RPC will sequence through the TAR locations when automatically loading the parameters for a Dispatch that are being written to Port 0 if the TX bit is detected in SAR3. During Dispatch, two read cycles will be performed to load the Font word (first) and the Texture word (second), and then a read-modify-write cycle will merge the Font, Texture and background at the Destination Address. The Font and Texture can be in any combination of Font Memory or Video Memory, but the Destination of a Dispatch is always the Video Memory. Due to the additional read cycle, the performance of a Textured Dispatch may be less than non-textured, depending on how the memory cycles are programmed.

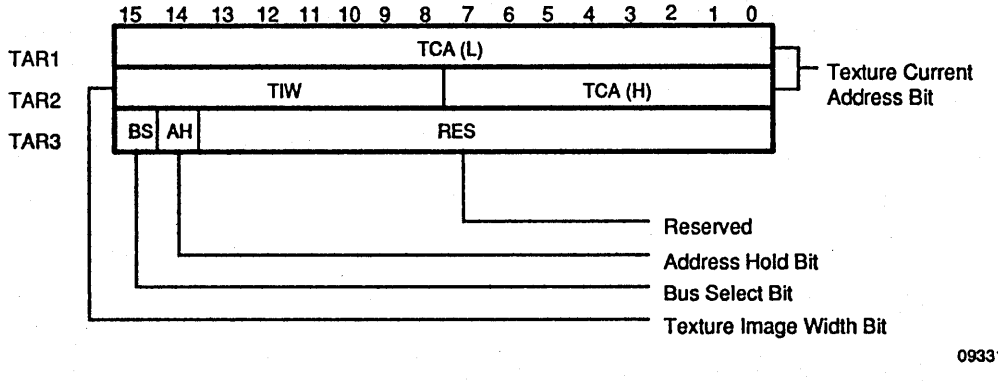


Figure 5. Texture Address Registers

Texture Address Registers

The Texture Address Registers are organized the same as the SAR words, however TAR3 contains less information (Figure 5). TAR1 holds the lower 16 bits of the current Texture Address and the low byte of TAR2 holds the upper 8 bits of this 24-bit address. The high byte of TAR2 holds the Texture Image Width which must be at least equal to the Source Image Width. In TAR3, only the two most significant bits are used.

Bit 15 of TAR3 is the Bus Select bit for the Texture. When this bit is a zero, the Texture is loaded from the Font Memory. When it is a one, the Texture is loaded from the Video Memory.

Bit 14 of TAR3 is the Address Hold bit. When this bit is set, the Texture Address for a Dispatch is not incremented. This allows a one word texture to be repeatedly applied to a character of any size.

The remaining lower bits of TAR3 are not used. A Texture Image Height is not required because the Source Image Height is used for determining the height of the block to be Dispatched. A Texture array must be as high as the largest character it is intended to be applied to. When setting up a Dispatch operation, the DAR must be the last two words programmed regardless of whether they were preceded by the SAR or TAR.

Destination Address Register

The Destination Address Registers are used in all of the RPC operating modes (Figure 6). While the SAR and TAR are predominantly used for only Dispatch, the address information for Font Load, Font Read, Graphics Load and the ORP modes is held in the DAR. It should be noted that the address for a Font Read is actually a "source" address, but the DAR is used to make programming consistent.

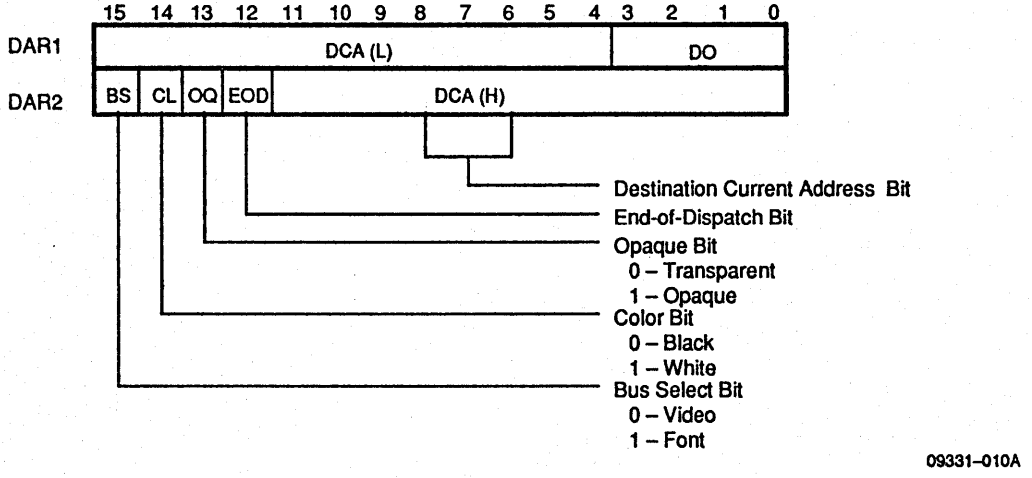


Figure 6. Destination Address Registers

DAR1 holds the lower portion of the Destination Address. In Dispatch and Graphics Load modes, the address is a 28-bit value to provide arbitrary pixel alignment of the words being transferred. The four least significant bits hold the Destination Offset field of DAR1, and provide the 0 to 15 pixel displacement for the starting point of pixel data for each destination word. Consequently, the 16 bits of DAR1 provide the lower 16 bits of the address in these two modes.

In Font Load, Font Read and ORP modes, DAR1 holds the lower 12 bits of the 24-bit address value in bits 15 through 4. Since the transfers in these modes must be word aligned, the four least-significant bits of DAR1 must be loaded with zeros.

DAR2 holds the most significant 12 bits of the Destination Address in the lower 12-bit positions of the word. This is the case for either a 24-bit or 28-bit address. The four most-significant bits of DAR2 hold control information for the current operation. Since the Destination Address is updated as the operation progresses, reads of DAR1 or DAR2 may not return the original value.

Bit 15 of DAR2 is the Bus Select bit. Font Load and Font Read modes may select either Font Memory or Video Memory space. When the BS bit is zero, Video Memory is selected. When it is a one, the Font Memory is selected. For Dispatch, Graphics Load and ORP modes, the BS bit is ignored. The destination for Dispatch and Graphics Load is always Video Memory. The ORP must reside on the Font Memory interface.

Bit 14 of DAR2 is the Color bit (CL). It is used to select whether the active area of a character is printed as black or white. When CL is zero, the character will be black. When CL is one, the character will be white.

Bit 13 of DAR2 is the Opaque bit (OQ). It is used to select whether the inactive area of a character is transparent or opaque. When OQ is zero, the space around the character is transparent and any existing background information will show through. When OQ is a one, the space around the character is opaque and will white-out any existing background. OQ must be a one for Font Load mode.

When both the CL and OQ bits are set, the inactive area around a character will be printed as black. This will black-out any existing background information. See the Video Operations Table for details.

The CL and OQ bits are effective only in Dispatch and Graphics Load modes. These two operations always generate read-modify-write cycles on the destination in Video Memory, so the background information can be preserved if appropriate. They also generate an extra cycle and mask portions of each word according to the Destination Offset. Font Load mode, even to the Video Memory, does a single aligned write cycle, so the offset must be zero and the OQ bit must be a one.

Bit 12 of DAR2 is the End-of-Dispatch bit. The presence of this flag overrides all other information in the SAR, TAR and DAR registers, and triggers an internal operation in the RPC. This sequence indicates that the last Dispatch of a page has been completed. This allows the next Dispatch operation to begin the next page, even if multiple blank bands are required to complete printing of the current page. The RPC will automatically shift out "white" pixels until the correct Page Y Size is reached.

The EOD flag may be set by explicitly writing the DAR to Ports 14 and 15, or by writing a "dummy" Dispatch sequence of the five words of SAR and DAR to Port 0. The RPC automatically initiates an operation after DAR2 is written. In the EOD case, it will be the internal operation that sets up for the next band and adjusts internal registers. DAR1 and DAR2 should both be written to ensure that the DMA Request output is controlled correctly. The next $\overline{DRQ}$ may not occur until printing has progressed to the band marked by the EOD.

Table 4. Video Operations

Case #	OQ Bit	CL Bit	TX Bit	Source Pixel	Texture Pixel	Background Pixel	Result Pixel
1	0	X	X	0	X	0	0
2	0	X	X	0	X	1	1
3	X	0	0	1	X	X	1
4	X	0	1	1	0	X	0
5	X	0	1	1	1	X	1
6	X	1	0	1	X	X	0
7	X	1	1	1	0	X	1
8	X	1	1	1	1	X	0
9	1	0	X	0	X	X	0
10	1	1	X	0	X	X	1

Notes:

Case 1 and 2: *Background shows through transparent inactive pixel*
Case 3: *Normal active pixel*
Case 4 and 5: *Texture overwrites active pixel*
Case 6: *CL reverses active pixel*
Case 7 and 8: *CL reverses texture that overwrites active pixel*
Case 9: *OQ causes inactive pixel to overwrite background*
Case 10: *CL and OQ reverses inactive pixel and overwrites background*

Operating Modes

The RPC performs six operating modes to support the transfer of pixel data among the memory resources of a printing system. These modes may be used to obtain capabilities and performance beyond the range of a general purpose processor executing the same tasks. Working jointly with a host processor, the RPC frees the host from the heavy data transfer burden, and allows it to perform higher level tasks for which it is better suited. Together, the RPC and host CPU can implement the most sophisticated features of an advanced printing system.

The fundamental operation for which the RPC is optimized is the transferring of stored rectangular image blocks to the page buffer. This is Dispatch mode and a number of options are available to augment it. In addition, three modes are provided to allow transfers of data between system memory space and the two memory interfaces controlled by the RPC. Graphics Load mode allows rectangular image blocks to be written to the page buffer with the same overlay options as Dispatch. Font Load and Font Read modes allow data to be transferred for storage and retrieval in a sequential word manner. ORP Load and ORP Read modes allow the RPC to efficiently work with the Am95C76 ORP to rotate image blocks in the process of assembling the page buffer.

Dispatch Mode

Dispatch is selected by writing 011 to the Mode Control field. Then in order to initiate a Dispatch, the CPU must load the Source Address Registers (SAR1-3) and Destination Address Registers (DAR1-2) and optionally the Texture Address Registers (TAR1-3) if texturing is desired. Loading these address registers may be done by writing to the explicit port location for each one or by writing them in sequence to Port 0. The sequence is SAR-DAR (five words) or SAR-TAR-DAR (eight words) for a textured Dispatch. The TAR words must be inserted if the Textured Font bit (TX) in SAR3 is set. The DAR words must be written last to deactivate the $\overline{DRQ}$ pin and trigger the start of the Dispatch operation.

The Source image and texture may be stored in either Font or Video Memory independently, as selected by their respective Bus Select (BS) bits. They will be read from consecutive word locations unless Address Hold (AH) is selected for either. AH is useful for area fill or for one-word textures. The texture addressing will not always be for consecutive memory locations if the Texture Image Width is greater than the Source Image Width. In this case, a new line of texture will be started with each new line of the Source image so that it will be consistently applied to images of varying size. Bit 15 of each image data word is assumed to be the left-most pixel and bit 0 is the right-most as they are read from the storage area.

The destination for Dispatch is always Video Memory. A Dispatch operation does not need to be word aligned, as the DAR provides a 28-bit address where the four least significant bits of DAR1 are an offset to indicate the position of the starting pixel in each word. Read-Modify-Write cycles are performed on the Video Memory words, the new image data is shifted as needed and any partial words outside the sides of an image block are masked so they remain unchanged. Dispatching progresses from left to right and top to bottom. The width must be specified in words and the image block must fit in the available Page X Size rounded up to a multiple of 16 pixels. (The block can't extend beyond the right side of the page.)

The data options for a Dispatch are controlled by the TX bit in SAR3 and the Color (CL) and Opaque (OQ) bits in DAR2. When TX is set, the RPC reads the first word of the Source image followed by the first word of the Texture. Then the first RMW cycle is performed at the Destination. This may modify a full word or partial word depending on the offset. This sequence allows the image, texture and any background information in the Video buffer to be combined. When the BS bits in SAR3 and TAR3 are both zero, indicating Font Memory, the read accesses after the first can occur in parallel with the Video Memory cycles to achieve higher performance. If the Font is stored in Video Memory, accesses can not occur in parallel. If only the Texture is stored in Video Memory, Font accesses can still be in parallel. Without texturing, the highest performance is obtained since one access is eliminated.

When a Dispatch operation completes, the Dispatch Complete (DC) bit in the Status is set and the $\overline{DRQ}$ pin is activated to request the next Dispatch. ($\overline{DRQ}$ can only be activated when the RPC is in Dispatch or ORP modes.) In cases where a Dispatch cannot complete normally, the $\overline{INTR}$ pin is activated. These interrupt conditions may be used for systems where the Video buffer is divided into two bands.

Video Bands

The RPC provides full support for systems having a Video Memory space that is much less than a full page. These features can also be used in systems having two full page buffer areas since the total memory space is limited only by the 28-bit address of the DAR. When the Video buffer is divided into two bands, Dispatching can be in progress in one band while printing is supported from the other band that has already been Dispatched. The bands alternate function in this manner until a page, or multiple pages are complete. This approach allows printer performance that may be limited only by the pixel rate of the print engine.

Since printing can be enabled by setting the Raster Control bit (RC) in the Mode register as soon as the first band is Dispatched, it is essential that all full or partial characters in the first band are completely Dispatched before the second band is begun. Each subsequent band of the page must be fully Dispatched in the time it takes to print the previous band, and Dispatching cannot progress into the next band until the printing of its previous contents is complete. The band control logic of the RPC will ensure that the printer is supplied with a con-

tinuous flow of pixels and that Dispatching is held off if necessary.

The first band will always start at location 000 0000 Hex in Video Memory, which corresponds to the upper left corner. The size of the two bands is determined by the value programmed in the Video Band Boundary (VBB). This value must be the address of the first word of the second band. During Dispatch of the first band, the RPC compares the current destination address to the VBB. If it is greater than or equal to the VBB, the Dispatch is stopped, the Dispatch Incomplete (DI) flag is set and $\overline{INTR}$ is activated.

The DI interrupt must be processed by the host CPU to handle the cases of characters that were sliced by the band boundary. This requires reading the contents of the SAR, TAR (if used) and DAR, and saving this information in a sliced character table in system memory. These partial characters can be Dispatched into the next band after the current band is complete. In the DI case, the $\overline{DRQ}$ pin is activated for the next Dispatch after DAR2 is read. The remaining full or partial characters are Dispatched to the current band and DI is activated whenever a sliced character is detected.

When the starting address for a Dispatch equals or exceeds the current band boundary, the operation is not initiated, the Band Dispatched (BD) flag is set, the current Band for Dispatching is toggled and $\overline{INTR}$ is activated. This provides the opportunity to Dispatch any entries in the sliced character table to the new band. The BD interrupt will be delayed if the previous data in the new band has not been completely scanned out to the printer. After BD is set, the second attempt to Dispatch into the new band will be accepted. If there are no sliced characters pending, the address for the first whole character of the new band should be sent a second time. (Actually only the DAR words need to be rewritten to trigger the Dispatch.)

The RPC calculates the current band boundary from the VBB and automatically converts the DAR provided into the correct physical memory address for the current band. Characters need only be sorted according to the virtual address on the full page. Each horizontal line of a character is maintained in the correct position according to the Image Width and the Page X Size Register.

After the last Dispatch of the band that has the last characters on a page, one final set of DAR words must be written to the RPC with the End-of-Dispatch (EOD) flag set. This may be accomplished by writing to Ports 14 and 15 explicitly, or by sending a dummy Dispatch sequence to Port 0. When EOD is detected, the rest of the registers are ignored and an internal operation is executed. This marks the band as the last of a page and allows the next Dispatch to begin a new page without causing any interrupts. The current page may contain multiple blank bands after the one marked by EOD. The RPC will automatically scan out "white" until the Page Y Size Register indicates that the page is complete. Dispatching of the first band of the next page does not have to be delayed to accomplish this.

In a system with a single full-page buffer, an EOD sequence is not required. Instead, printing can be enabled after the last Dispatch of the page. After printing is complete, as indicated by the End-of-Page interrupt, a Software Reset must be written to the Mode Register. This causes internal registers to be adjusted so that the next page will begin at the top of Video Memory space. Then Dispatch mode can be reselected and the next page can be started.

Graphics Load Mode

Graphics load is selected by writing 110 to the Mode control field. Then SAR1 must be loaded with the correct Image Width in words, and the DAR must be loaded with the 28-bit address of the upper left-hand corner of the destination. A Graphics Load block does not have to be word aligned. The four least significant bits of DAR1 are an offset to indicate the position of the starting pixel in each word. Each memory cycle to the destination in Video Memory is triggered by a CPU write of image data to the Temporary Register at Port 0. The RPC performs Read-Modify-Write cycles so the Graphics block image may be combined with any existing background information, according to the CL and OQ bits of DAR2, in the same manner as Dispatch.

Graphics Load progresses from left to right until the number of words indicated by the Image Width have been written. Then the RPC calculates the start of the next horizontal line of the block according to the Image Width and the Page X Size Register, and continues with the sequence of image words being received from the CPU. Lines will continue from the top down until the CPU stops writing to Port 0. If the DAR offset is not zero, the last word written by the CPU should complete a horizontal line. If a partial line is desired, the last portion of the last word written mid-line will not be transferred to Video Memory unless one more word is written by the CPU. This extra word should be blank.

Graphics Load is effectively terminated when the CPU programs any new operation. There is no inherent Image Height required. The starting DAR is assumed to be the physical memory location desired and no address conversion is performed to adjust for the current Video Band like a Dispatch. Graphics Load can be used among Dispatch operations as long as the image block fits in the current band and the DAR provided is the correct physical memory address. This should always be the case in a full-page buffer system.

Video Data Operations

In Dispatch and Graphics Load the CL and OQ options determine how the image is combined with the background. When CL is zero, the active area of a character is printed as ones, or black. When CL is a one, the character is printed as zeros, or white. When OQ is zero, the inactive pixels of an image block will be transparent and allow any background information to show through. When OQ is a one, the inactive pixels will white-out any background information. If both CL and OQ are set, the normal resultant pixels (i.e., when CL equals zero and OQ equals one) get reversed. The active pixels of a

character will become zeros unless they are to be textured and the texture pixel is a zero, then they will be ones. The inactive pixels of the block will become ones and effectively black-out any background. Table 4 shows the result for all combinations of data and options.

Font Load Mode

Font Load Mode is used for storing information from the CPU to Font Memory or Video Memory. It is selected by writing 100 to the Mode Control field. The DAR must be programmed with a 24-bit address and the Bus Select bit must be zero to indicate Video Memory or one to indicate Font Memory. Since Font Load performs word aligned write cycles to the destination, the OQ bit must be a one and the CL bit and the offset field must be zeros.

Font Memory write cycles are triggered each time the CPU writes to the Temporary Register at Port 0. The Destination Address is incremented after each write cycle. The operation ends when the CPU stops writing data to Port 0, or a new operation is programmed. The DAR may be changed as needed for multiple Font Load operations.

Font Read Mode

Font Read Mode allows the CPU to access stored information in either the Font Memory or the Video Memory. It is selected by writing 101 to the Mode Control field. The DAR must be programmed with a 24-bit address for the first word to be read and the Bus Select bit must be zero to indicate Video Memory or one to indicate Font Memory. The CL, OQ, and offset field are ignored.

As soon as DAR2 is loaded, the RPC reads the first word into the Temporary Register and waits for the CPU to read it out. Each access of Port 0 by the CPU causes the DAR to be incremented and another word to be read from Font or Video Memory. The CPU can read as many words as desired, change the DAR or program a new operation as needed.

ORP Load Mode

ORP Load Mode allows the RPC to load data from the Font Memory to the Am95C76 for character rotation. It is selected by writing 010 to the Mode Control field. SAR1 must be programmed with a word count that is the exact number of words to be loaded into the ORP. The DAR must be loaded with the 24-bit address for the first word to be loaded. When DAR2 is loaded, the RPC executes a continuous stream of Font Memory read cycles until the word count reaches zero. BS, CL, OQ, and the offset in the DAR are ignored. The data can be strobed into the ORP and is of no consequence to the RPC. When the word count in SAR1 has been decremented to zero, the RPC will set the DC bit and activate the $\overline{DRQ}$ pin to request the next operation. The CPU can alternate ORP Load and Dispatch operations in order to transfer rotated characters to the Video buffer.

ORP Read Mode

ORP Read Mode allows the RPC to read rotated data from the ORP to be stored in Font Memory. It is selected by writing 001 to the Mode Control field. SAR1 must programmed with the word count and the DAR must be loaded with the 24-bit address of the destination of the first word. Loading DAR2 triggers a continuous stream of Font Memory write cycles until the word count reaches zero. The RPC does not drive any data onto the FAD_0-FAD_{15} bus during the $\overline{FWR}$ strobe. This is provided by the ORP in the proper sequence. The DC bit and the $\overline{DRQ}$ pin are activated as in ORP Load.

5

Printer Operation

The RPC provides a number of programmable options in order to easily interface to a variety of print engines and accommodate a broad range of page sizes. This information must be programmed into the appropriate Operation Control Registers (OCR) before printing is initiated. Some of the OCR words are required to execute memory operations and it is recommended that the entire OCR be loaded after the power-up and Reset. This is best accomplished by loading the Memory Timing Register (MTR) at Port 7 first, and then continuing in descending order to the Mode at Port 1. This insures that the Refresh Rate will be setup before Refresh is enabled and that all other options are correctly initialized. Page Size and Margin values may be altered as needed for different pages but this must be done before any Dispatching or Graphics Load operation is started.

In most systems an Opaque Dispatch operation should be used to initially clear the entire Video Memory buffer. The SAR can be held pointing to an all zero word. Since the maximum size of a Dispatched block is 4096 x 4096 pixels, several operations may be required. Once this is done after power-up, The RPC will automatically clear the buffer area after each page is printed, unless the Multiple Page option is used.

The Printer Interface may be operated in either Synchronous or Asynchronous Mode, depending on how the print engine timing generates LSYNC relative to VCLK. If the rising edge of LSYNC occurs synchronous to VCLK and meets the setup time required, SYNC Mode can be used and the pixel rate on the VDO pin will equal the VCLK rate. If LSYNC has no fixed timing relationship to VCLK, ASYNC Mode should be used and the pixel rate will be the VCLK rate divided by four. PSYNC needs no fixed timing relationship to VCLK, but the rising edge of PSYNC must precede LSYNC by a minimum of one VCLK so the Y Margin may be evaluated. The printer mode is selected by the VCK bit in the Memory Timing Register at Port 7.

The polarity of the VDO output is selected by the Video Polarity (VP) bit in the Mode Register. When VP is a zero, VDO outputs a HIGH level to indicate a black (active) pixel and a LOW level to indicate white. When VP is a one, VDO outputs a LOW for black and HIGH for white. In any case, VDO is driven HIGH when printing is idle or printing is outside the active page area defined by the page size.

Printing cannot be initiated until the Video buffer has been prepared with the first page or band using Dispatch or Graphics Load. Once this has been done, the Raster Complete (RC) bit in the Mode Register may be set and the RPC will load the internal shift register for the VDO output with the first word from Video Memory location 000 0000 hex. Normally Read-Modify-Write cycles are used to support scan-out, so that each word is written with zeros as scanning progresses in preparation for the next band or page. If the MPC bit is set in the Mode Register, read cycles are used for scan-out so that the same full page can be printed more than once. MPC should be set in advance of RC, or it can be set with RC if there is no memory operation in progress.

The RPC provides two interrupts to simplify the control of printing. The End-of-Page flag in the Mode/Status Register is set and the $\overline{\text{INTR}}$ pin is activated when a number of lines equal to the Page Y Size Register have been shifted out on VDO. The RC bit is automatically cleared at this time. This interrupt can be used to control when the next page is started or when the Multiple Print Control bit (MPC) should be cleared before the last printing of a single page that was printed multiple times. RC cannot be set if EOP is active.

The Raster Incomplete (RI) flag indicates an error condition during printing that cannot be recovered from. RI is set when an LSYNC is detected in the middle of an active scan line. LSYNC cannot be accepted until a number of pixels equal to the programmed Page X Size have been scanned out. RC is cleared if RI is set and the RPC must be programmed to start over on a page after the cause of RI has been determined.

System Interface

The System Interface of the RPC can be operated completely asynchronously to the CLK input. Accesses by the host CPU or external DMA controller may be made at any time and will be internally synchronized by the RPC. The RDY output will be driven inactive if the access cannot be immediately completed.

During Dispatch or ORP modes, the $\overline{DRQ}$ output can be used to control when the RPC is ready for the next operation to be programmed. $\overline{DRQ}$ will be driven active after each operation is completed or when a DI or BD interrupt has been handled. Alternatively, Status may be polled to determine when each Dispatch is complete by testing the DC flag. Status flags are only cleared after Status is read, so DC can stay active through multiple Dispatches. The RDY output may be used to extend CPU accesses until the RPC is able to accept new information, but this may be a long time during a large Dispatch and care must be taken not to modify essential values in the middle of an operation. The Mode/Status Register is always accessible to the CPU but some changes to control bits may not take effect immediately if an operation is in progress. Software Reset and changes to RC are always recognized.

Font Interface

Font Memory cycles may be programmed for automatic insertion of wait states by the FMW field of the MTR. During Dispatch, Font cycles may skip the upper address cycle and FALE2 if it does not need to be updated. This is done to improve performance.

When the Am95C76 ORP is used, it must reside in Font Memory space. The ORP may be programmed or accessed by using Font Load and Font Read modes to the locations reserved for it. ORP Loads and Dispatch operations may be alternated so that rotated characters may be assembled into the page buffer as desired.

Video Interface

The timing for Video Memory cycles may be programmed by several fields in the MTR register at Port 7. The Video Memory Size (VMS) field selects the type of Dynamic RAM that is used in the Video buffer in order to provide the correct combination of Row and Column addresses and bank select bits. The Video Precharge Extension (VPX) bit increases the precharge time when $\overline{VRAS}$ and $\overline{VCAS}$ are inactive by one CLK if needed by the RAMs. The Video Timing Extension (VTX) bit increases the Row address hold time by one CLK to allow for bank decoding if needed by the memory system.

The Video Write Wait State (VWW) bit and the Video Read Wait State (VRW) field allow wait states to be automatically inserted to extend Video Memory cycles to match the performance of the Dynamic RAMs used. For details on wait states see the MTR register description and the Video Memory cycle timing diagrams.

RESET

The $\overline{RESET}$ pin must be activated by the system after power-up. This causes an internal operation that initializes scratch registers that are used for calculation of memory addresses. In addition, the Mode Register is cleared. All other user-accessible registers must be properly initialized before any RPC operations are started. The Video buffer must be cleared by programming appropriate Dispatch operations. $\overline{RESET}$ will always terminate any operations in progress including Refresh of the Video Memory.

A Software Reset operation may be executed by changing the Mode Control field to 000 from any other value. This will terminate an operation in progress and initialize internal scratch registers to begin a new Dispatch in Band 0, and begin printing at the top of Video Memory space. Software Reset will not alter the setting of the VP, VRE or IE bits in the Mode/Status Register and dynamic memory Refresh will not be interrupted.

5

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	–65 to +150°C
Ambient Operating Temperature	–55 to +125°C
Maximum V_{cc} Relative to V_{ss}	–0.3 to +7.0 V
DC Voltage Applied to Any Pin Relative to V_{ss}	–0.5 to Vcc+0.3 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A)	0 to +70°C
Supply Voltage (V_{cc})	+4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Max.	Unit
V_{IL}	Input LOW Voltage		–0.3	+0.8	V
V_{IH}	Input HIGH Voltage	CLK Input only	+2.4	V_{CC}+0.3	V
		All other inputs	+2.0	V_{CC}+0.3	V
V_{OL}	Output LOW Voltage	I_{OL} = 3.2 µA		0.4	V
V_{OH}	Output HIGH Voltage	I_{OH} = 250 µA	+2.4		V
II	Input Leakage Current	$0.4<V_{OUT}<V_{CC}$		±10	µA
I_{OZ}	Output Leakage Current	$0.4<V_{IN}<V_{CC}$		±10	µA
I_{CC}	Power Supply Current			70	mA

PRELIMINARY

CAPACITANCE*

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Max.	Unit
C_{IN}	Input Capacitance			25	pF
$C_{I/O}$	I/O Pin Capacitance			25	pF
C_{OUT}	Output Pin Capacitance			25	pF

* Parameters are not "Tested".

SWITCHING CHARACTERISTICS over operating ranges

Parameter Number	Parameter Symbol	Parameter Description	Min.	Max.	Unit
1	t_{CLK}	CLK Cycle Time	80	1000	ns
2	t_W	CLK HIGH Pulse Width	35		ns
3	t_W	CLK LOW Pulse Width	35		ns
4	t_{PD}	CLK Edge to Output Valid Delay		45	ns
5	t_{PD}	CLK Edge to Output Float Delay (Note 1)		55	ns
CPU INTERFACE					
6	t_S	A_0–A_3 Valid to $\overline{RD}$ or $\overline{WR}$ FE Setup	10		ns
7	t_H	$\overline{RD}$ or $\overline{WR}$ RE to A_0–A_3 Valid Hold	10		ns
8	t_S	$\overline{CS}$ Valid to $\overline{RD}$ or $\overline{WR}$ FE Setup	0		ns
9	t_H	$\overline{RD}$ or $\overline{WR}$ RE to $\overline{CS}$ RE Hold	10		ns
10	t_{PD}	$\overline{RD}$ or $\overline{WR}$ FE to RDY Delay		40	ns
10a	t_{PD}	$\overline{WR}$ FE to $\overline{DRQ}$ RE Delay		40	ns
11	t_{PD}	Data Out Valid to RDY RE Delay	CLK HIGH –10		ns
12	t_H	RDY RE to or $\overline{WR}$ RE Hold	25		ns
13	t_H	$\overline{RD}$ RE to Data Out Valid Hold	0		ns
14	t_D	$\overline{RD}$ RE to Data Out Float Delay (Note 1)		55	ns
15	t_S	Data In Valid to $\overline{WR}$ RE Setup	20		ns
16	t_H	$\overline{WR}$ RE to Data In Valid Hold	20		ns
17	t_W	$\overline{WR}$ LOW Width (Note 2)	60		ns
18	t_W	$\overline{WR}$ Recovery	1 CLK+20		ns
19	t_W	$\overline{RD}$ Recovery	0.5		CLK
FONT INTERFACE					
20	t_H	FALE1 or FALE2 FE to FAD_{0-15} Valid Hold	10		ns
21	t_S	Data In Valid to CLK RE Setup	10		ns
22	t_H	$\overline{FRD}$ RE to Data In Valid Hold	0		ns
23	t_H	$\overline{FWR}$ RE to Data Out Valid Hold	10		ns
23a	t_W	FALE1 or FALE2 HIGH Width	CLK LOW –10		ns
VIDEO INTERFACE					
24	t_S	Data In Valid to CLK RE Setup	20		ns
25	t_H	$\overline{VR}$ RE to Data In Valid Hold	0		ns
PRINTER INTERFACE (Notes 3, 4)					
26	t_{CLK}	VCLK Cycle Time (Synchronous Mode)	50	2000	ns
27	t_W	VCLK HIGH Pulse Width (Synchronous Mode)	20		ns
28	t_W	VCLK LOW Pulse Width (Synchronous Mode)	20		ns
29	t_{CLK}	VCLK Cycle Time (Asynchronous Mode)	31	2000	ns
30	t_W	VCLK HIGH Pulse Width (Asynchronous Mode)	10		ns
31	t_W	VCLK LOW Pulse Width (Asynchronous Mode)	10		ns
32	t_W	PSYNC Pulse Width	1		VCLK
33	t_D	PSYNC RE to LSYNC RE Delay	50		ns
34	t_W	LSYNC Pulse Width	1		VCLK
35	t_S	LSYNC RE to VCLK RE Setup (Synchronous Mode)	15		ns
36	t_D	VCLK RE to VDO Valid Delay (Synchronous Mode)		40	ns
37	t_D	VCLK Edge to VDO Valid Delay (Asynchronous Mode)	1.5 VCLK + 45		ns

PRELIMINARY

Notes:

1) Parameter #5 and #14–Float times indicate that the RPC is no longer driving bus outputs and are measured by determining when outputs that were driven LOW have risen above a level of 0.8 V.

2) Parameter #17–$\overline{WR}$ Width (LOW)–applies only when RDY does not go LOW. When the RPC is not ready to accept a write, specs #10 and #12 will apply and determine the necessary write width.

Notes (continued):

3) PSYNC will not be recognized immediately after the Raster Scan Control bit (RC) is set by the CPU. Two words must be prefetched from the Video Memory before printing of a new page can begin. This requires two Read-Modify-Write cycles plus completion of any memory cycle in progress (approximately 25 CLKs minimum). A PSYNC that occurs before the data for VDO is ready will be ignored.

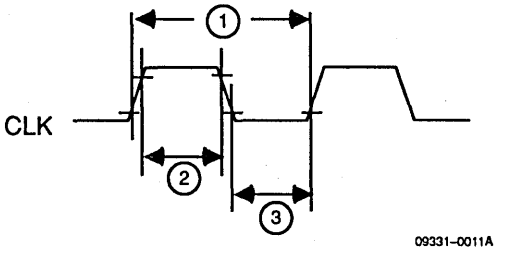
4) LSYNC for a new line cannot occur immediately after the last pixel of the previous line has been clocked out by VCLK. At least 10 VCLKs must occur between the last valid pixel on VDO and the next LSYNC in order for the X Margin and X Size values to be processed correctly.

Output-to-Output Timing Relationships

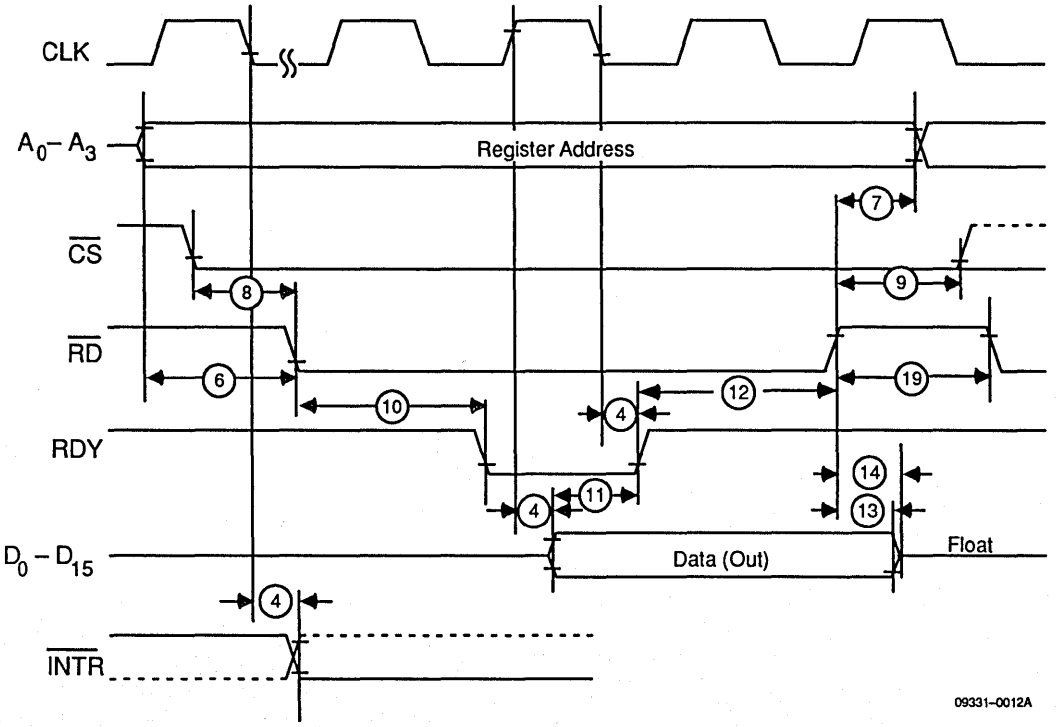
Parameter #4 CLK Edge to Output Valid Delay–can be used for all clocked outputs as indicated in the timing diagrams. All outputs are clocked in the same manner and the delay for any two unrelated outputs will track within 10 ns. Output-to-output relationships can be determined by the number of full or half CLKs between them according to the formula:

TD = (n CLKs x Cycle Time)–10 ns

SWITCHING WAVEFORMS

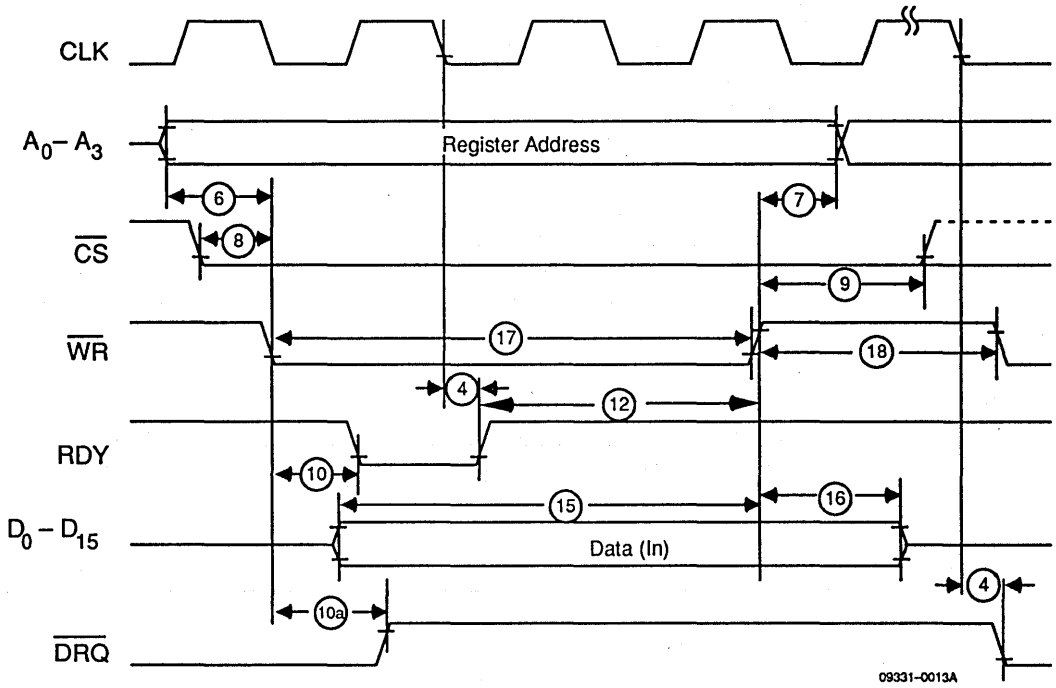


Clock Cycle Timing

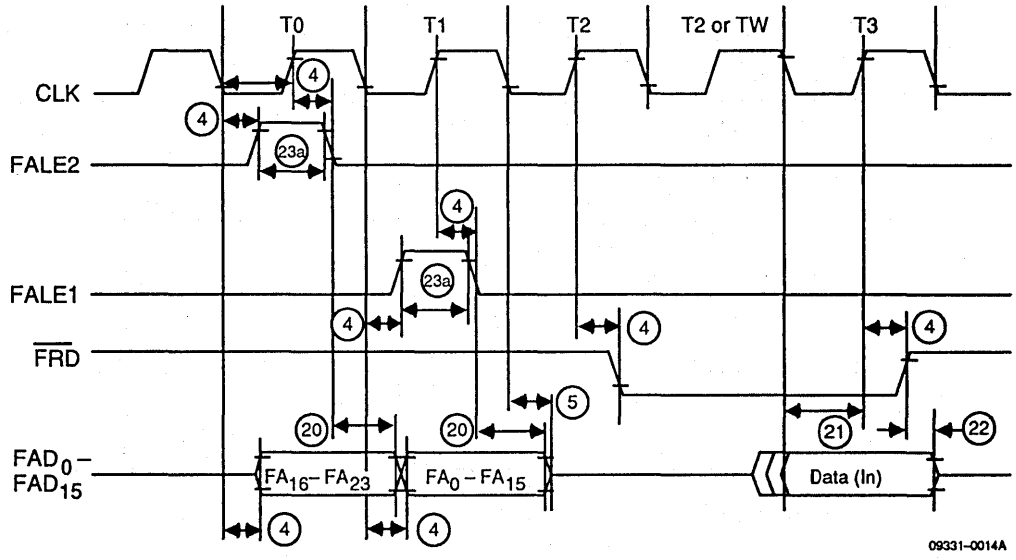


CPU Read Access Timing

SWITCHING WAVEFORMS (continued)

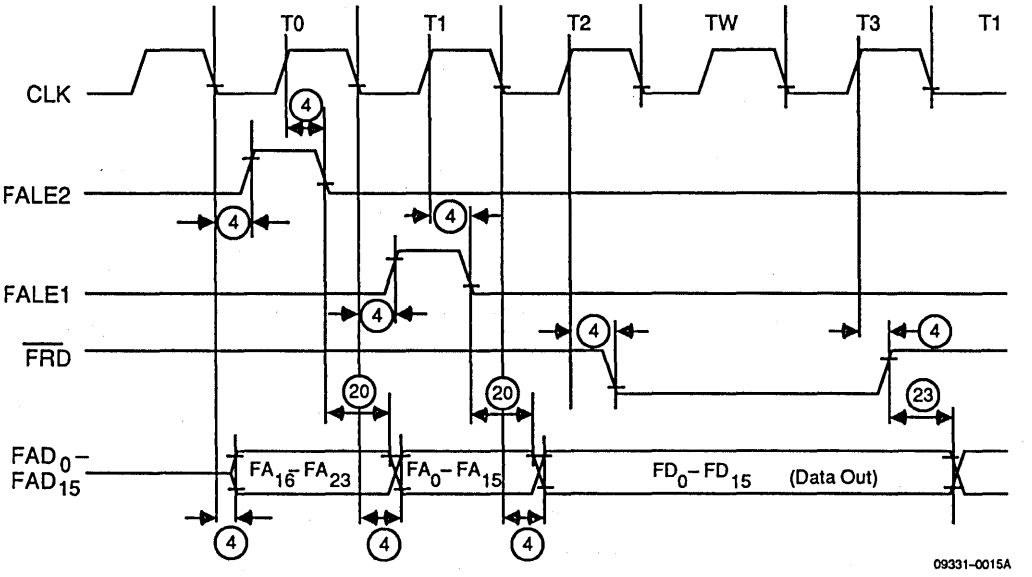


CPU Write Access Timing

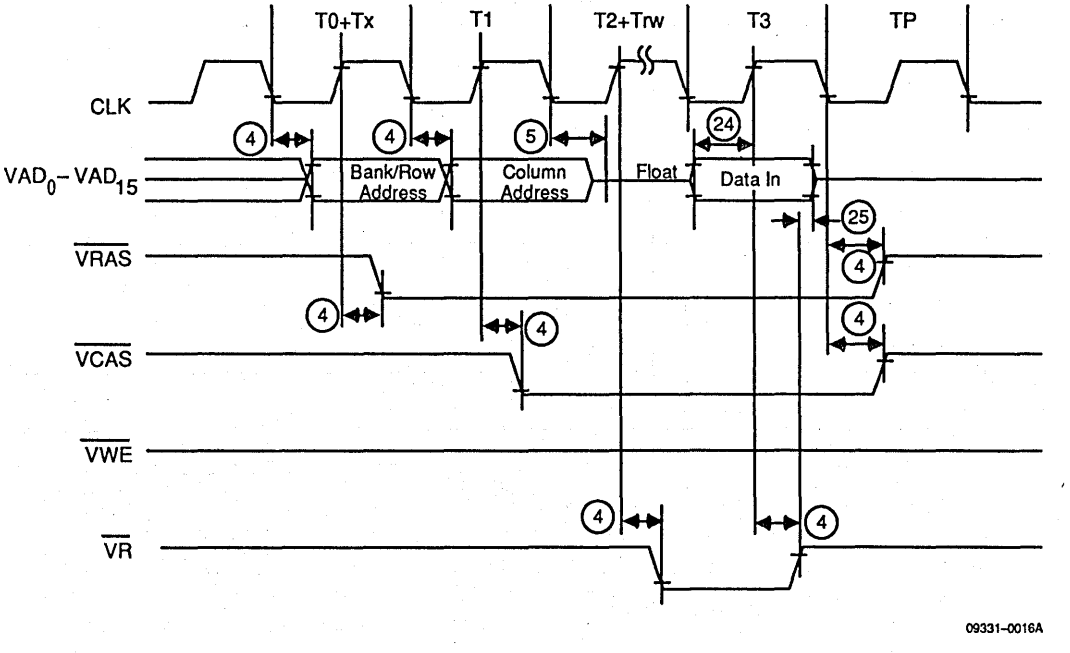


Font Memory Read Access Timing

SWITCHING WAVEFORMS (continued)

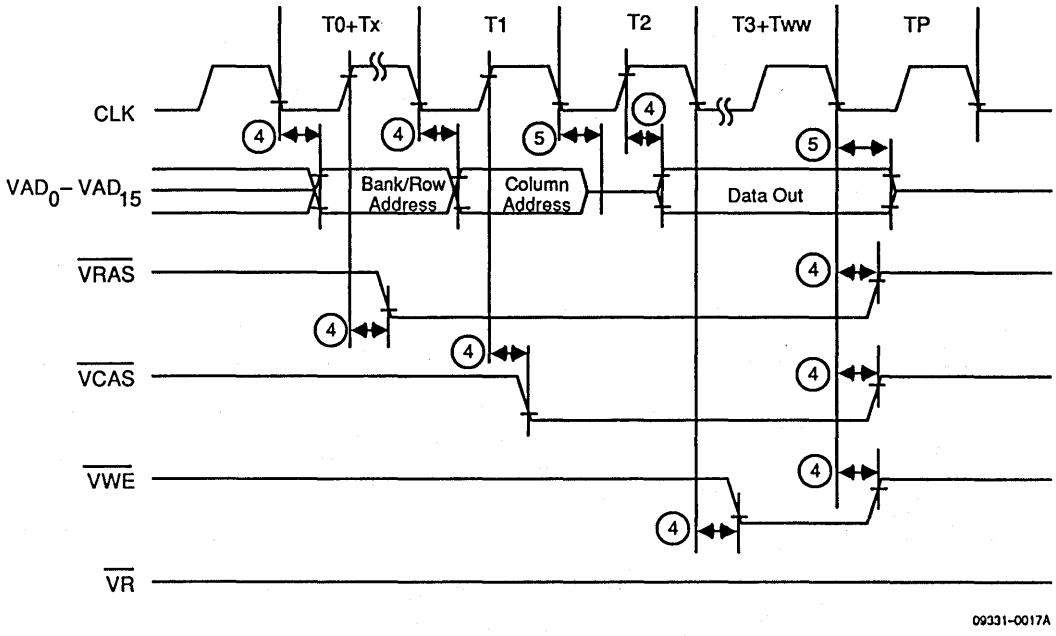


Font Memory Write Access Timing

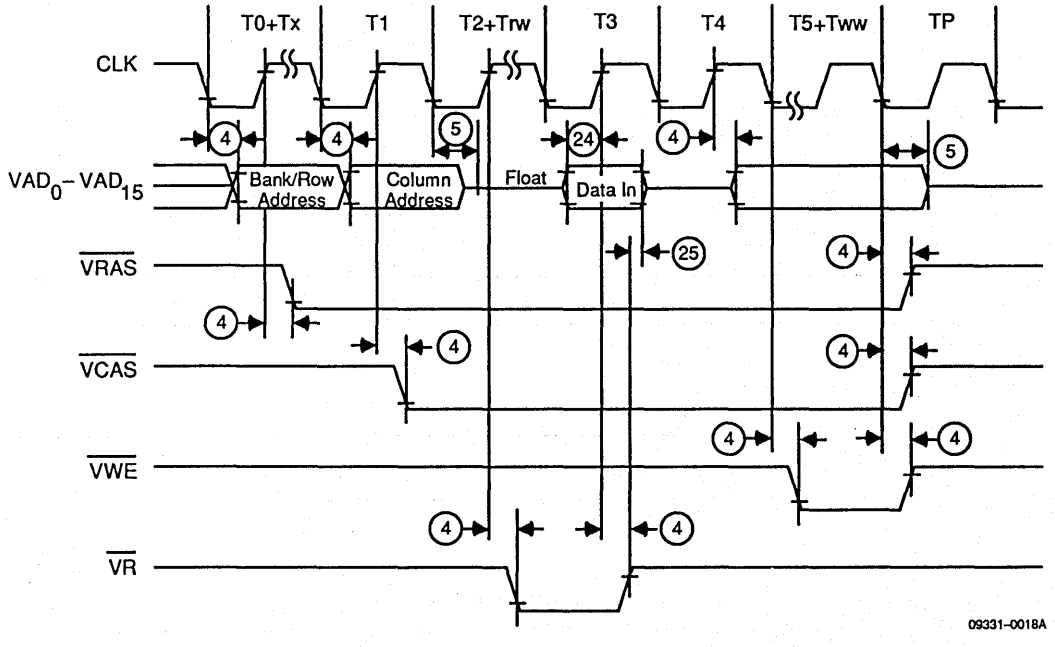


Video I/F Read Access Timing

SWITCHING WAVEFORMS (continued)

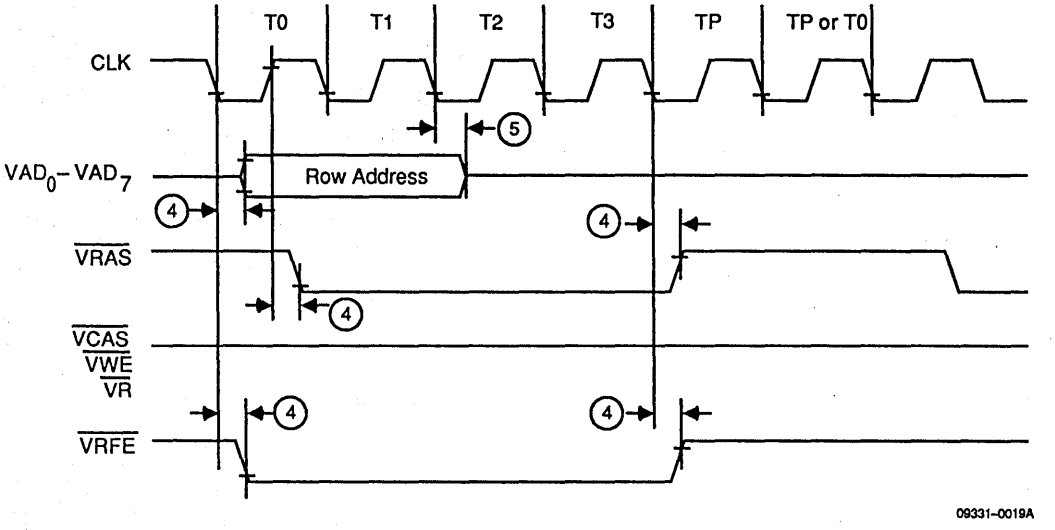


Video I/F Write Access Timing

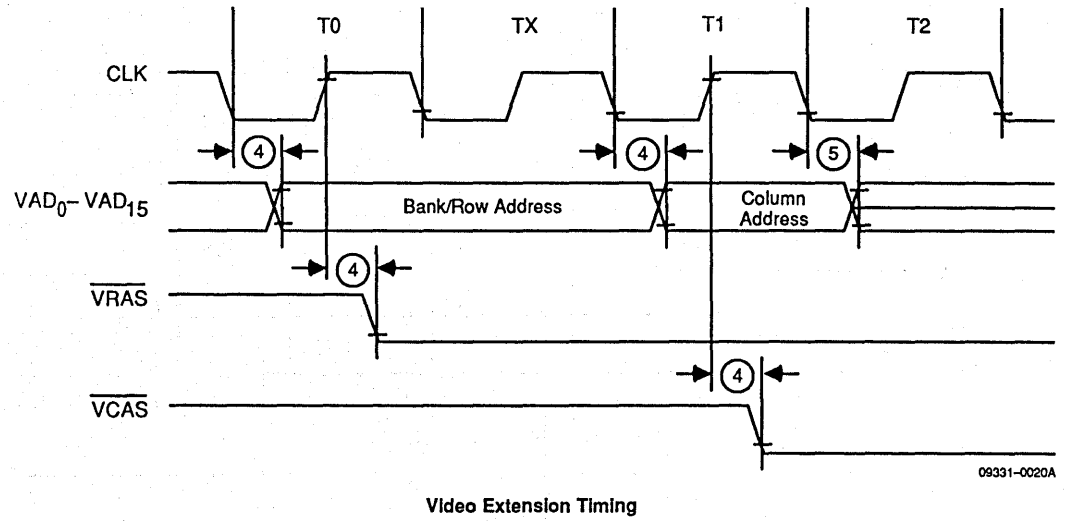


Video I/F Read-Modify-Write Access Timing

SWITCHING WAVEFORMS (continued)

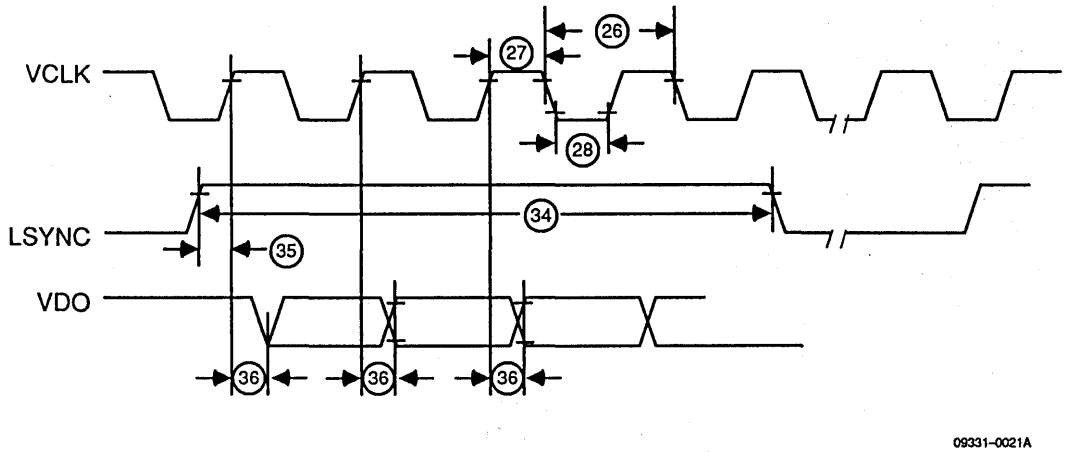


Video I/F Refresh Timing



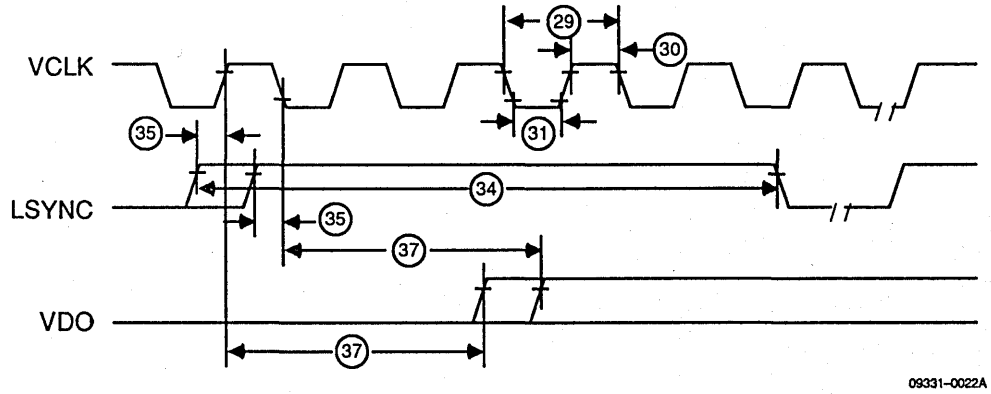
Video Extension Timing

SWITCHING WAVEFORMS (continued)



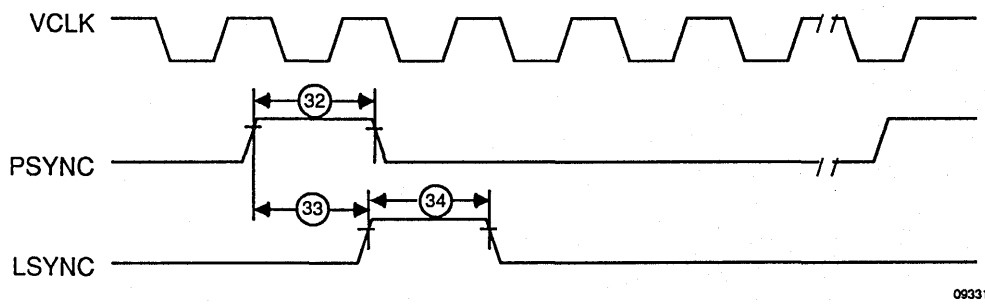
09331-0021A

LBP Timing (SYNC Mode)



09331-0022A

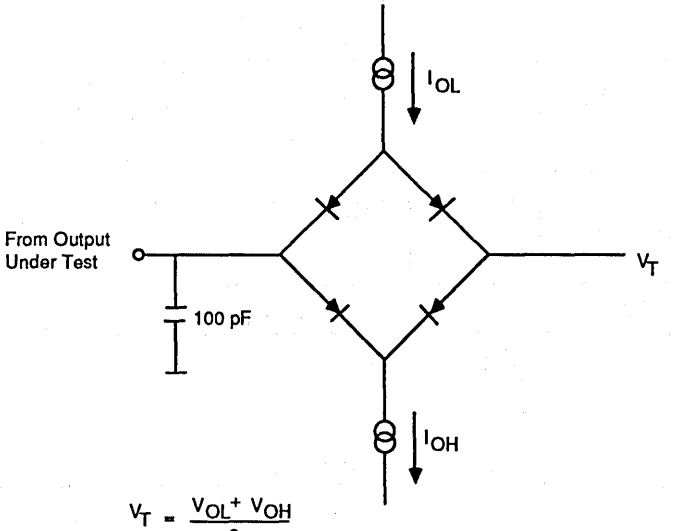
LBP Timing (ASYNC Mode)



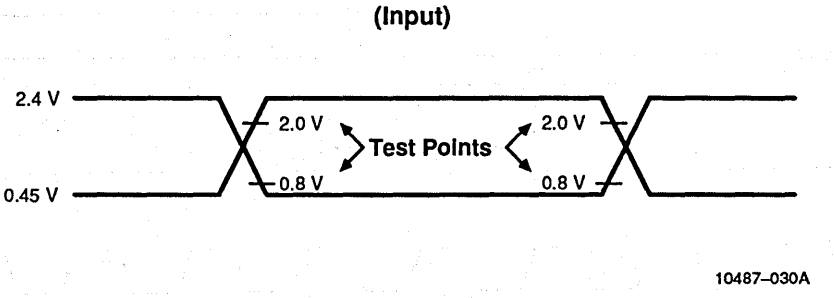
09331-0023A

PSYNC Timing

SWITCHING TEST CIRCUIT



SWITCHING TEST WAVEFORM



Am95C76

Orthogonal Rotation Processor (ORP)

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- Performs high-speed orthogonal rotation of font characters or blocks of bit-mapped data (0, 90, 180, and 270 degrees)
- Interfaces easily to the Am95C75 Raster Printer Controller (RPC) to allow high-speed printing with minimum font memory
- Can operate in standalone mode
- 64 x 64-bit internal memory is large enough to accommodate a standard-size font character at 400 dots-per-inch resolution
- Cascadable in all directions to handle large point-size fonts or bit-mapped data blocks
- Cascades automatically — interface signals make the ORP array function as one large ORP without additional user actions

GENERAL DESCRIPTION

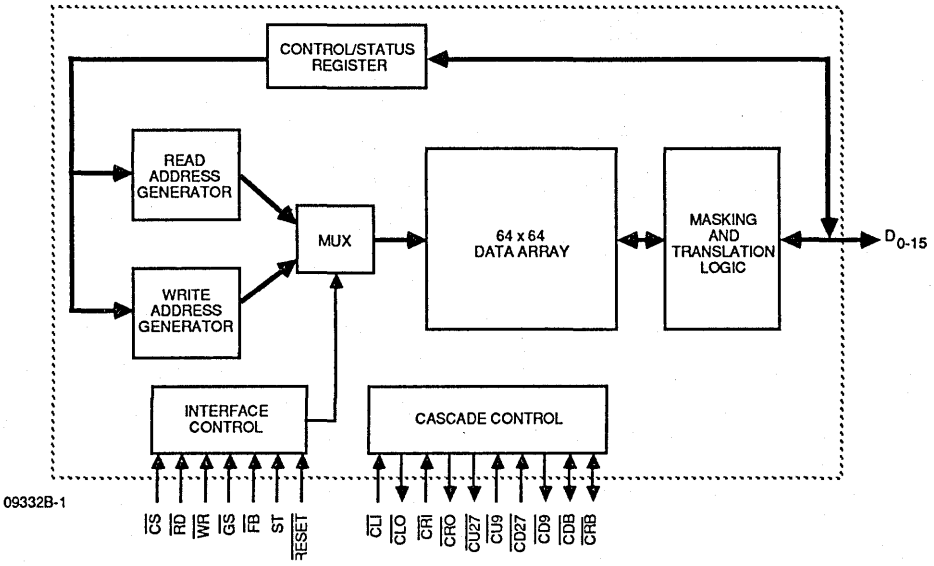
The Orthogonal Rotation Processor (ORP) is a high-performance CMOS device designed to provide orthogonal (0, 90, 180, and 270 degrees) rotation of font characters or blocks of bit-mapped data. In graphics systems, users may print data in portrait mode — where each line is printed across the narrow part of the page, or in landscape mode — where text is printed across the wide part of the page. To accomplish this, each font is usually stored in both portrait and landscape modes. Additionally, if two-sided (duplex) printing is desired, fonts must usually be stored with 180- and 270-degree rotations to print portrait and landscape modes on the back of a page. By using the ORP, 0-, 90-, 180-, and 270-degree rotations are possible, eliminating three-quarters of font memory conventionally used.

The ORP rotates an image block (a font character or other bit-mapped data) in two steps: first, the image block is loaded into the ORP where character size is user-programmed and data is transferred on 16-bit word boundaries; then the data is read from the ORP to the video buffer memory, where the page is assembled with the user-selected rotation.

The only user-accessible register in the ORP is the Control Status Register (CSR), which must be set up with valid parameters prior to each image block transfer. These parameters include image block dimensions, rotation angle, and an enable bit.

The 64 x 64-bit ORP memory may be increased by cascading additional ORPs; interface signals pass information between ORPs to allow the ORP array to function as one large ORP without additional control from the user.

BLOCK DIAGRAM



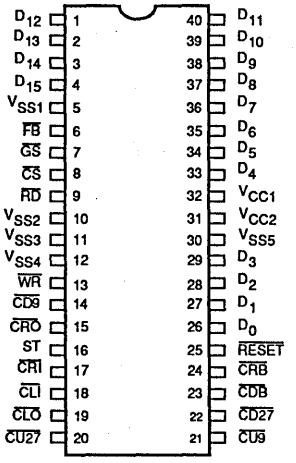
Publication #	Rev.	Amendment
09332	B	/0

Issue Date: September 1987

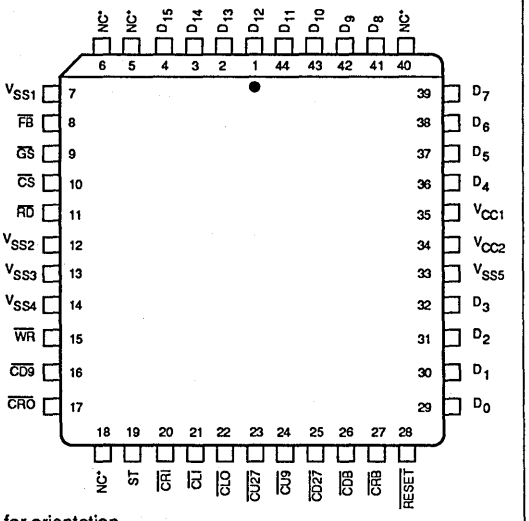
CONNECTION DIAGRAMS

Top View

DIP



PLCC

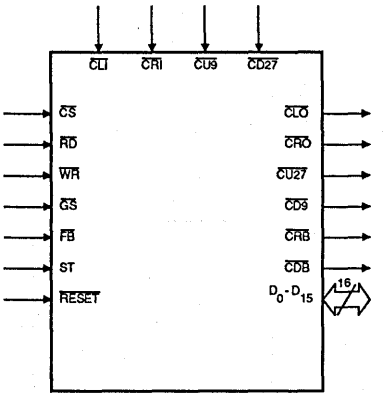


Note: Pin 1 marked for orientation.

09332A-2

09332A-3

LOGIC SYMBOL



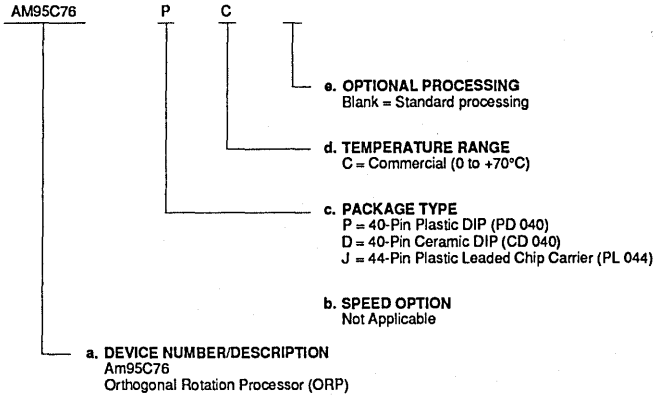
09332A-4

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations

Valid Combinations	
AM95C76	PC DC JC

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

5

PIN DESCRIPTION

$\overline{CD9}$ Cascade Down 90 (Output; Active LOW)
The $\overline{CD9}$ output signal controls the vertical direction token passing in an ORP cascade. The $\overline{CD9}$ output pin of an ORP must be connected to the $\overline{CU9}$ input pin of the ORP below it in the cascade. The $\overline{CD9}$ pin is asserted LOW on **one** of the following conditions: 1) the ORP is in 90-degree Read mode and the current data array read access results in a complete vertical row access (as defined by the XAR field of CSR); and 2) the ORP is in Write mode or 0-degree Read mode, the $\overline{CLI}$ input is asserted LOW, and the entire font section assigned to the ORP has been written to/read from. $\overline{CD9}$ output assertion of an ORP causes the cascade token to be passed to the ORP below it in the cascade.

$\overline{CD27}$ Cascade Down 270 (Input; Active LOW)
The $\overline{CD27}$ input signal controls the vertical direction token passing in an ORP cascade. The $\overline{CD27}$ input pin of an ORP must be connected to the $\overline{CU27}$ output pin of the ORP below it in the cascade. The cascade token is passed through this input pin in 270-degree Read mode and 180-degree Read mode.

$\overline{CDB}$ Cascade Down Bypass (Input/Output; Open Drain, Active LOW)
The $\overline{CDB}$ signal controls the vertical direction token passing in an ORP cascade. This pin is an output when the ORP is in 90-degree Read mode. $\overline{CDB}$ is driven LOW if the ORP is disabled (bit 15 of the CSR is 0) and the $\overline{CU9}$ input pin is LOW. The $\overline{CDB}$ pins of all vertically cascaded ORPs should be connected together and tied to +5 V with an external pull-up resistor. $\overline{CDB}$ pin is an input if the ORP is in 270-degree Read mode; if the ORP is disabled (bit 15 of the CSR = 0) and the $\overline{CDB}$ input is asserted LOW, the $\overline{CU27}$ output pin will be driven LOW immediately.

$\overline{CLI}$ Cascade Left In (Input; Active LOW)
The $\overline{CLI}$ input signal controls the horizontal direction token passing in an ORP cascade. The $\overline{CLI}$ input pin of an ORP must be connected to the $\overline{CRO}$ output pin of the ORP to its right in the cascade. The cascade token is passed through this input pin in Write mode, 0-degree Read mode, and 270-degree Read mode.

$\overline{CLO}$ Cascade Left Out (Output; Active LOW)
The $\overline{CLO}$ output signal controls the horizontal direction token passing in an ORP cascade. The $\overline{CLO}$ output pin of an ORP must be connected to the $\overline{CRI}$ input pin of the ORP to its left in the cascade. The $\overline{CLO}$ pin is asserted LOW on **one** of the following conditions: 1) the ORP is in 90-degree Read mode, the $\overline{CU9}$ input is LOW, and the entire font section assigned to the ORP has been read; 2) the ORP is in 180-degree Read mode and the current data array read access results in a complete horizontal row access (as defined by the XAR field of the CSR); and 3) the ORP is in 180-degree Read mode, bit 15 of CSR is 0 (disabled), and $\overline{CRB}$ is LOW. $\overline{CLO}$ output assertion of an ORP causes the cascade token to be passed to the ORP to its left in the cascade.

$\overline{CRB}$ Cascade Right Bypass (Input/Output; Open Drain, Active LOW)
The $\overline{CRB}$ signal controls the horizontal direction token passing in an ORP cascade. This pin is an output when the ORP is in Write mode or 0-degree Read mode. In this case, $\overline{CRB}$ is driven LOW if the ORP is disabled (bit 15 of the CSR is 0) and the $\overline{CLI}$ input pin is asserted LOW. The $\overline{CRB}$ pins of all horizontally cascaded ORPs should be connected together and tied to +5 V with an external pull-up resistor. $\overline{CRB}$ pin is an input if the ORP is in 180-degree Read mode; if the ORP is disabled (bit 15 of the CSR = 0) and the $\overline{CRB}$ input is asserted LOW, the $\overline{CLO}$ output pin will be driven LOW immediately.

$\overline{CRI}$ Cascade Right In (Input; Active LOW)
The $\overline{CRI}$ input signal controls the horizontal direction token passing in an ORP cascade. The $\overline{CRI}$ input pin of an ORP must be connected to the $\overline{CLO}$ output pin of the ORP to its right in the cascade. The cascade token is passed through this input pin in 90-degree Read mode and 180-degree Read mode.

$\overline{CRO}$ Cascade Right Out (Output; Active LOW)
The $\overline{CRO}$ output signal controls the horizontal direction token passing in an ORP cascade. The $\overline{CRO}$ output pin of an ORP must be connected to the $\overline{CLI}$ input pin of the ORP to its right in the cascade. The $\overline{CRO}$ pin is asserted LOW on **one** of the following conditions: 1) the ORP is in 270-degree Read mode, the $\overline{CD27}$ input is LOW, and the entire font section assigned to the ORP has been read; and 2) the ORP is in Write mode or 0-degree Read mode and the current data array write/read access results in a complete horizontal row access (as defined by the XAR field of the CSR). $\overline{CRO}$ output assertion of an ORP causes the cascade token to be passed to the ORP to its right in the cascade.

$\overline{CS}$ Control/Status Select (Input; Active LOW)
The $\overline{CS}$ input is an active-LOW signal used by the host processor to access the Control Status Register (CSR) of the ORP. $\overline{CS}$ pins of cascaded ORPs must not be connected together.

$\overline{CU9}$ Cascade Up 90 (Input; Active LOW)
The $\overline{CU9}$ input signal controls the vertical direction token passing in an ORP cascade. The $\overline{CU9}$ input pin of an ORP must be connected to the $\overline{CD9}$ output pin of the ORP above it in the cascade. The cascade token is passed through this input pin in Write mode, 0-degree Read mode, and 90-degree Read mode.

$\overline{CU27}$ Cascade Up 270 (Output; Active LOW)
The $\overline{CU27}$ output signal controls the vertical direction token passing in an ORP cascade. The $\overline{CU27}$ output pin of an ORP must be connected to the $\overline{CD27}$ input pin of the ORP above it in the cascade. The $\overline{CU27}$ pin is asserted LOW on **one** of the following conditions: 1) the ORP is in 270-degree Read mode and the current data array read access results in a complete vertical row access (as defined by the XAR field of CSR); 2) the ORP is in 180-

degree Read mode, the $\overline{CRI}$ input is LOW, and the entire font section assigned to the ORP has been read; and 3) the ORP is in 270-degree Read mode, bit 15 of the CSR is 0 (disabled), and the $\overline{CDB}$ input is LOW. $\overline{CU27}$ output assertion of an ORP causes the cascade token to be passed to the ORP above it in the cascade.

D_0–D_{15} Data Bus (Input/Output; Three State)
The 16-bit Data Bus is used for data transfer between the ORP and the host processor or Font Memory.

$\overline{FB}$ $\overline{Fly\text{-}By}$ (Input; Active LOW)
The $\overline{FB}$ input is an active-LOW signal used by the host processor to access the data array of the ORP during a Fly-By operation. An $\overline{RD}$ strobe assertion in conjuction with an $\overline{FB}$ assertion causes a write access to the ORP data array if the R/$\overline{W}$ bit in the Control Status Register is 0. A $\overline{WR}$ assertion in conjunction with an $\overline{FB}$ assertion causes a read access to the ORP data array if the R/$\overline{W}$ bit in the CSR is 1. $\overline{FB}$ pins of cascaded ORPs should be connected together.

$\overline{GS}$ $\overline{Group\ Select}$ (Input; Active LOW)
The $\overline{GS}$ input is an active-LOW signal used by the host processor to access the data array of the ORP during a Normal Read/Write operation. $\overline{GS}$ pins of cascaded ORPs should be connected together.

$\overline{RD}$ $\overline{Read}$ (Input; Active LOW)
The $\overline{RD}$ input is an active-LOW signal used by the host processor to read the CSR when $\overline{CS}$ input is asserted, to read from the ORP data array when the $\overline{GS}$ input is asserted, and to write into the ORP data array when $\overline{FB}$ input is asserted. On either a CSR read access or a data array read access, the ORP will drive D_0–D_{15} with data while $\overline{RD}$ input is LOW. In Fly-By mode the $\overline{RD}$ input is used as a write strobe; the ORP uses the rising edge of $\overline{RD}$ to write the data on pins D_0–D_{15} into the internal data array. $\overline{RD}$ pins on cascaded ORPs should be connected together.

$\overline{RESET}$ $\overline{Reset}$ (Input; Active LOW)
$\overline{RESET}$ is an asynchronous active-LOW input which initializes the ORP. The effect of $\overline{RESET}$ is to clear all 16 bits of the Control Status Register, force D_0–D_{15} pins to high-impedance state, and force all cascade output pins inactive (HIGH).

ST Start (Input; Active HIGH)
The ST input is an active-HIGH signal used by the host processor to initiate a read/write access to the ORP. The ST inputs of all cascaded ORPs should be connected together. When connecting the ORP to the RPC (Am95C75), the ST input of the ORP should be connected to the FALE1 pin of the RPC.

V_{CC1}, V_{CC2} +5-V Power Supply

V_{SS1}, V_{SS2}, V_{SS3}, V_{SS4}, V_{SS5} Ground

$\overline{WR}$ $\overline{Write}$ (Input; Active LOW)
The $\overline{WR}$ input is an active-LOW signal used by the host processor to write into the CSR when $\overline{CS}$ input is asserted, to write into the ORP data array when $\overline{GS}$ input is asserted, and to read from the ORP data array when $\overline{FB}$ input is asserted. On either a CSR write access or a Normal data array write access, D_0–D_{15} must be driven with valid data prior to the rising edge of $\overline{WR}$. In Fly-By mode, the $\overline{WR}$ input is used as a read strobe; the ORP will drive D_0–D_{15} with data from the internal data array while $\overline{WR}$ is LOW. $\overline{WR}$ pins of cascaded ORPs should be connected together.

FUNCTIONAL DESCRIPTION

The block diagram for the Am95C76 is shown on the front cover. Communication with the external host processor takes place over the 16-bit Data Bus, D_0–D_{15}. Transfers over the Data Bus are controlled by the $\overline{CS}$, $\overline{GS}$, $\overline{FB}$, ST, $\overline{RD}$, and $\overline{WR}$ input lines. Communication between cascaded ORPs occur over the cascade control signals $\overline{CLO}$, $\overline{CRO}$, $\overline{CLI}$, $\overline{CRI}$, $\overline{CD9}$, $\overline{CU27}$, $\overline{CU9}$, $\overline{CD27}$, $\overline{CRB}$, and $\overline{CDB}$.

There are six major functional blocks in the ORP: 1) 64 x 64 data array, 2) cascade control logic, 3) host processor interface control logic, 4) control status register (CSR), 5) data array read/write address generation logic, and 6) data array read mask translation logic. The ORP is controlled by the 16-bit control status register (CSR). The CSR specifies the font horizontal size, font vertical size, Read/Write mode, angle of rotation, font clipping status, chip enable, and cascade token.

Control Status Register (CSR)

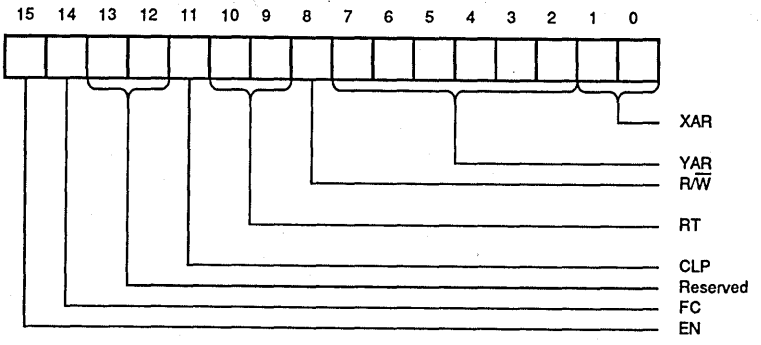
This 16-bit register is used for issuing commands and receiving status information. Figure 1 shows the bit assignment of the CSR. The $\overline{CS}$ input pin acts as the read/write enable for CSR. When the $\overline{RD}$ input is LOW, the ORP status information is driven onto the D_0–D_{15} lines; when the $\overline{WR}$ input is LOW, the data on D_0–D_{15} lines is loaded into the CSR. The CSR is cleared when the $\overline{RESET}$ input pin is LOW. There are two reserved bits in the CSR; these bits must be programmed to 0.

Chip Enable (EN)

The EN bit controls the state of the ORP. When this bit is set to 1, the ORP is enabled and data transfer between the internal data array and the D_0–D_{15} pins is allowed. When this bit is cleared to 0, the ORP is disabled and no data transfers between the internal data array and the D_0–D_{15} pins can occur. In a multiple ORP cascade, the unused ORPs should be disabled by programming their EN bits to 0.

First Chip (FC)

The FC bit is the cascade token. When this bit is set to 1, the ORP is enabled (provided EN is set to 1) and data transfer between the internal data array and the D_0–D_{15} pins is allowed. When this bit is programmed to 0, the ORP is disabled and data transfer between the internal data array and the D_0–D_{15} pins is inhibited. While initializing ORPs only one ORP should have its FC bit set to 1. The location of the ORP whose FC bit is to be initialized to 1 is a function of the rotation angle while performing a font or bit-map data array read; see Figure 2 for details. Before a font or bit-map data array write, the FC bit of the **top-left** ORP should be set to 1. **For a single ORP application, the FC bit should be set to 1 to enable data array read/write.** The FC bit is propagated between ORPs in the cascade configuration through the cascade interface control pins, without intervention by the host processor. Since the host processor interface signals (ST, $\overline{GS}$, $\overline{FB}$, $\overline{RD}$, $\overline{WR}$, D_0–D_{15}) of cascaded ORPs are inter-connected, the FC bit ensures that the internal data array of only **one** of the cascaded ORPs is connected to the D_0–D_{15} lines at any time.



Key: XAR = X image size in 16-bit words
YAR = Y pixel size
R/W = 1 (Read only)
0 (Write only)
RT = 00 (0° Rotation)
01 (90° Rotation)
10 (180° Rotation)
11 (270° Rotation)
CLP = 1 (No translation)
0 (Translates logical to physical address)
FC = 1 (Enable token)
0 (Disable token)
EN = 1 (Enable)
0 (Disable)

09332B-5

Figure 1. Control Status Register Format

Clipping (CLP)

This bit is used to generate the starting address for the internal data array in conjunction with the YAR and XAR fields of the CSR. The CLP bit should be programmed to 0 when the XAR and YAR fields of the CSR specify the unprocessed font size (e.g., ORP initialization for a **complete** font write/read from the internal data array). The ORP translates the logical addresses to physical addresses for the internal data array if the CLP bit is programmed to 0.

The CLP bit is set to 1 by the ORP after an internal data array access has occurred. The CLP bit significantly reduces host processor overhead in resuming ORP data array writes/reads on a partially processed font. Partially processed fonts occur when the ORP is used in conjunction with image blocks larger than the video memory size. Typically the video memory is configured to hold an integral number of scan lines of the image block. Fonts get "clipped" at the video memory overflow boundary; therefore, the ORP is designed to rotate partially processed fonts. The ORP provides the host processor with the next physical Y address and the font X size on a CSR read after clipping. The host processor then stores the CSR value in memory and copies the stored value into CSR to resume processing on a clipped font. Since the CLP bit would be set by the ORP to 1 after a data array access, the XAR and YAR fields are interpreted as physical addresses for the internal data array. The CLP bit should not be cleared to 0 by the host processor on resuming clipped font processing.

Rotation Mode (RT)

These bits specify the angle of rotation while reading data out of the ORP internal data array. The ORP allows for 0-, 90-, 180-, and 270-degree rotations. The following table shows the definition of this field:

Bit 1	Bit 0	Rotation Angle (Counter Clockwise)
0	0	0 degree
0	1	90 degree
1	0	180 degree
1	1	270 degree

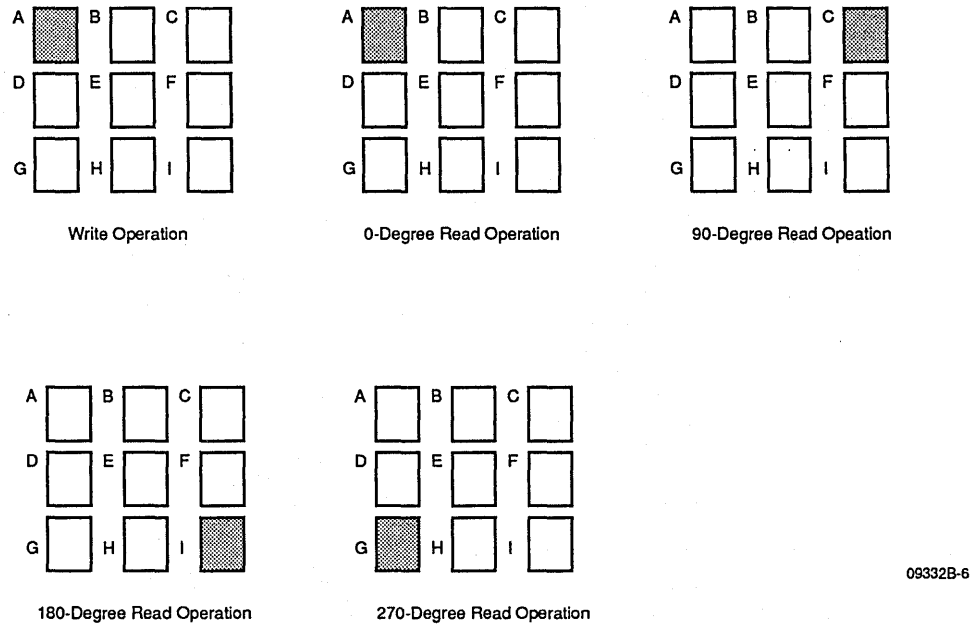


Figure 2. FC Bit Position in Cascade Configuration
(While processing 3 x 3 fonts/images > 128 pixels in both X & Y dimensions)

Reset Mode

The ORP enters into Reset mode when the $\overline{RESET}$ input pin is asserted. The Control Status Register is cleared; since the EN bit and FC bits are cleared, all data transfers through the Data Bus D_0–D_{15}, except for write to CSR, are ignored by the ORP.

Normal Read Mode

The ORP is in Normal Read mode when the R/$\overline{W}$ bit in CSR is 1 and the $\overline{GS}$ and $\overline{RD}$ input pins are asserted LOW. If the EN and FC bits in CSR are both set to 1, a data array read transaction occurs; if either bit is 0, the data array read transaction is inhibited and D_0–D_{15} output pins float.

Normal Write Mode

The ORP is in Normal Write mode when the R/$\overline{W}$ bit in CSR is 0 and the $\overline{GS}$ and $\overline{WR}$ input pins are asserted LOW. If the EN and FC bits in CSR are both set to 1, a data array write transaction occurs; if either bit is 0, the data array write transaction is inhibited and D_0–D_{15} input pins are ignored.

Fly-By Read Mode

The ORP is in Fly-By Read mode when the R/$\overline{W}$ bit in CSR is 1 and the $\overline{FB}$ and $\overline{WR}$ input pins are asserted LOW. The Fly-By Read mode allows the host processor to simultaneously read from the ORP data array and write into the Font Memory (e.g., while building a Font Memory of rotated fonts using the ORP to rotate font data). If the EN and FC bits in CSR are both set to 1, a data array read transaction occurs; if either bit is 0, the data array read transaction is inhibited and D_0–D_{15} pins float.

Fly-By Write Mode

The ORP is in Fly-By Write mode when the R/$\overline{W}$ bit in CSR is 0 and the $\overline{FB}$ and $\overline{RD}$ input pins are asserted LOW. The Fly-By Write mode allows the host processor to simultaneously read from the Font Memory and write into the ORP (e.g., while down-loading a font into the ORP for rotation). If the EN and FC bits in CSR are both set to 1, a data-array write transaction occurs; if either bit is 0, the data-array write transaction is inhibited and D_0–D_{15} input pins are ignored.

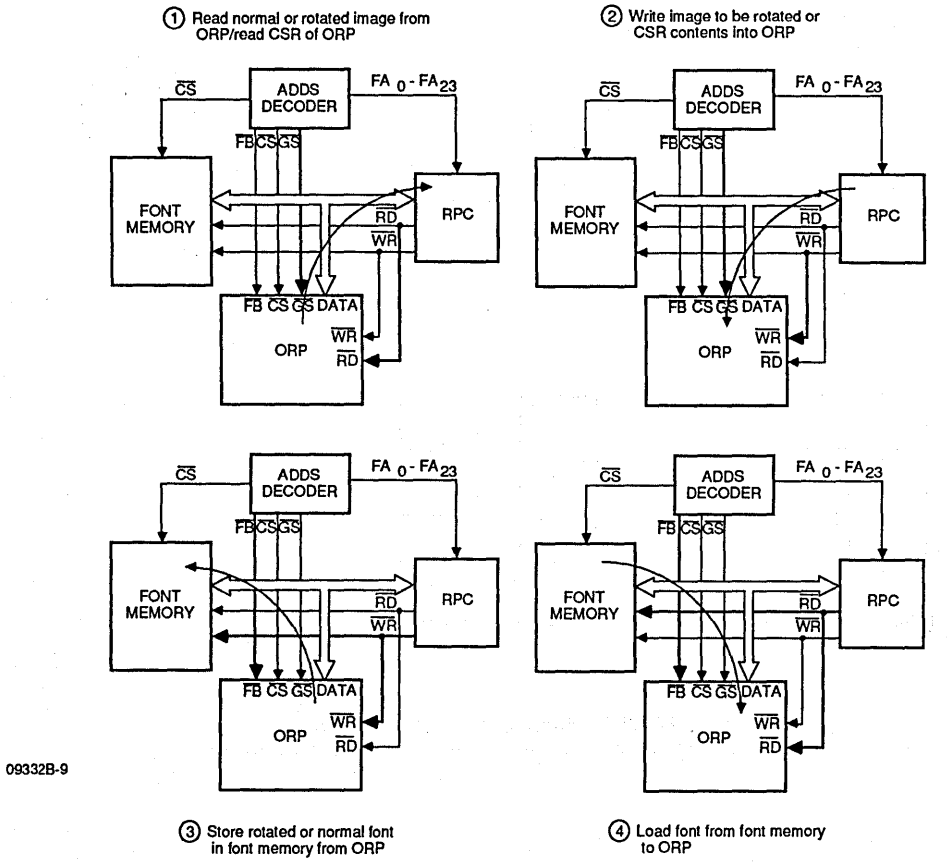


Figure 4. ORP Operation Modes

Figure 3-1 shows the output image of each rotation mode. Note that the RT field is ignored when data is written into the ORP data array (R/$\overline{W}$ = 0).

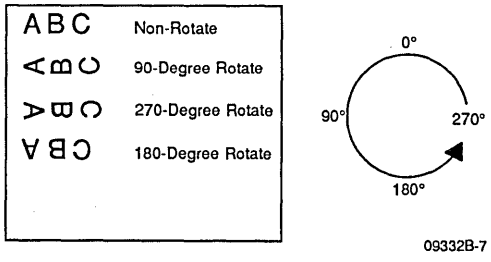


Figure 3-1. Output Format of Each Rotation Mode

Read/$\overline{\text{Write}}$ (R/$\overline{W}$)

The R/$\overline{W}$ bit controls the ORP operation mode. If this bit is programmed to 1, the ORP data array is configured as read only (i.e., Normal Read or Fly-By Read mode). If this bit is 0, the ORP data array is configured as write only (i.e., Normal Write or Fly-By Write mode).

Y Address Register (YAR)

The 6-bit YAR field specifies the Y size of the image block in pixels. The actual value loaded into YAR is the required number of pixels minus one.

In 90- and 180-degree rotation modes, the content of YAR represents the physical start address for the data array in the Y direction.

In 0- and 270-degree rotation modes, the initialized content of YAR represents the physical end address for the data array in the Y direction. If the CLP bit is programmed to 0, the physical start address is assumed to be 0 in the Y direction, whereas if the CLP bit is programmed to 1, the value in the YAR field is interpreted as the physical start address in the Y direction for processing a clipped font.

X Address Register (XAR)

The 2-bit XAR field specifies the X size of the image block in words. The value loaded into XAR is the required number of words minus one.

Under the 180- and 270-degree rotation modes, the content of XAR represents the physical start address for the data array in the X direction.

Under the 0- and 90-degree rotation modes, the content of XAR represents the physical end address of the data array in the X direction.

Operation Modes

The ORP has seven operation modes: 1) CSR Read, 2) CSR Write, 3) Reset, 4) Normal Read, 5) Normal Write, 6) Fly-By Read, and 7) Fly-By Write (see Figure 4). Any combination of inputs other than those defined below are invalid operating modes and are ignored by the ORP.

CSR Read Mode

When $\overline{CS}$ and $\overline{RD}$ are asserted LOW, the D_0–D_{15} lines are driven with CSR contents.

CSR Write Mode

When $\overline{CS}$ and $\overline{WR}$ are asserted LOW, the data on D_0–D_{15} is latched into the CSR on the trailing edge of $\overline{WR}$.

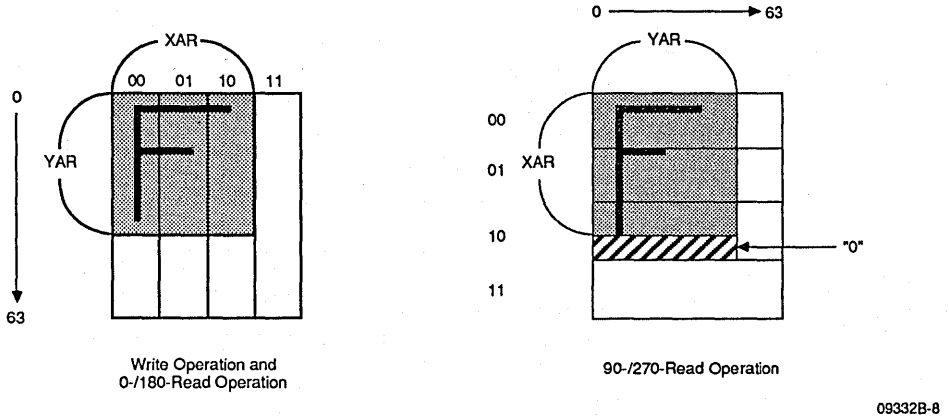


Figure 3-2. Relationship Between XAR Value and YAR Value

Data Handling and Formats

All data transferred on the host processor interface to the ORP are 16 bits wide; this is true for CSR transactions as well as data-array transactions. The internal data-array addressing and the bit assignment are different for the write access and the four read rotation modes. Figure 5 shows the MSB/LSB position in each operation.

In a data-array write operation, 0-degree read operation, and 90-degree read operation, the bit assignment of the data array corresponds to the actual image block; the top-left position is the MSB bit. In a 180- or 270-degree data-array read operation, the MSB/LSB is in reverse order; the bit reversal is handled within the ORP (see Figure 5).

Cascade Control

Multiple ORPs can be cascaded in any number, both horizontally and vertically, with respect to font scan direction. In a cascade configuration, any size font or bit-map image can be accommodated. The ten cascade signals (described in the pin description section) control the cascade token (FC) propagation through the multi-ORP cascade. $\overline{CLI}$, $\overline{CLO}$, $\overline{CRI}$, $\overline{CRO}$, and $\overline{CRB}$ control horizontal direction cascade token passing. $\overline{CU9}$, $\overline{CU27}$, $\overline{CD9}$, $\overline{CD27}$, and $\overline{CDB}$ control vertical direction cascade token passing. If the EN bit of CSR of a cascaded ORP is 0, then the cascade token will be passed through it to the next enabled ORP. $\overline{CRB}$ and $\overline{CDB}$ are bidirectional, open-drain signals and facilitate token pass-through on disabled ORPs.

Figure 6 shows the input/output definition of the cascade control signals in the data array write mode and the four read rotation modes.

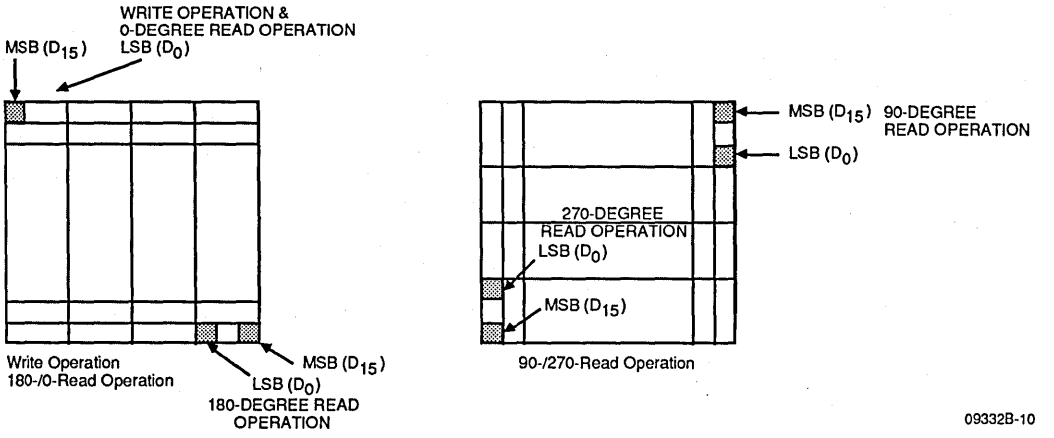


Figure 5. ORP Data Array Bit Position

	Write	0-Degree Read	90-Degree Read	180-Degree Read	270-Degree Read
$\overline{CLI}$	INPUT	INPUT			INPUT
$\overline{CLO}$			OUTPUT	OUTPUT	
$\overline{CRI}$			INPUT	INPUT	
$\overline{CRO}$	OUTPUT	OUTPUT			OUTPUT
$\overline{CU27}$				OUTPUT	OUTPUT
$\overline{CD9}$	OUTPUT	OUTPUT	OUTPUT		
$\overline{CD27}$				INPUT	INPUT
$\overline{CU9}$	INPUT	INPUT	INPUT		
$\overline{CRB}$	OUTPUT	OUTPUT		INPUT	
$\overline{CDB}$			OUTPUT		INPUT

09332B-11

Figure 6. Cascade Signal Input/Output Definition

Multi-ORP Cascade Example

Figure 7 shows an example of a multi-ORP cascade using four ORPs supporting processing of an image block up to 128 pixels x 128 pixels.

Write Mode

$\overline{CRO}$ must be connected to $\overline{CLI}$ of the right-side ORP. Each right-most ORP must have its $\overline{CRO}$ fed back to the $\overline{CLI}$ of the corresponding left-most ORP. The $\overline{CRB}$ in each ORP must be connected to the $\overline{CLI}$ at the left-most ORP. Because the $\overline{CRB}$ pin is an open-drain output, a pull-up resistor must be connected to this pin. The FC bit in the first ORP will be set to 1 by the CPU. The other ORPs' FC bits must be 0. After (XAR + 1) words have been loaded, the $\overline{CRO}$ (ORP #1) will be driven LOW and the second ORP's FC bit will then be set to 1. This process is repeated until the entire memory area is loaded with data. When the entire font section assigned to an ORP is loaded, $\overline{CLI}$ is connected internally to $\overline{CD9}$. When the final data for the second ORP is loaded, $\overline{CLI}$ of the first ORP will be driven LOW; thus $\overline{CU9}$ of the third ORP will be driven LOW and its FC bit will be set to 1. This propagation of the FC bit will continue between the next two ORPs.

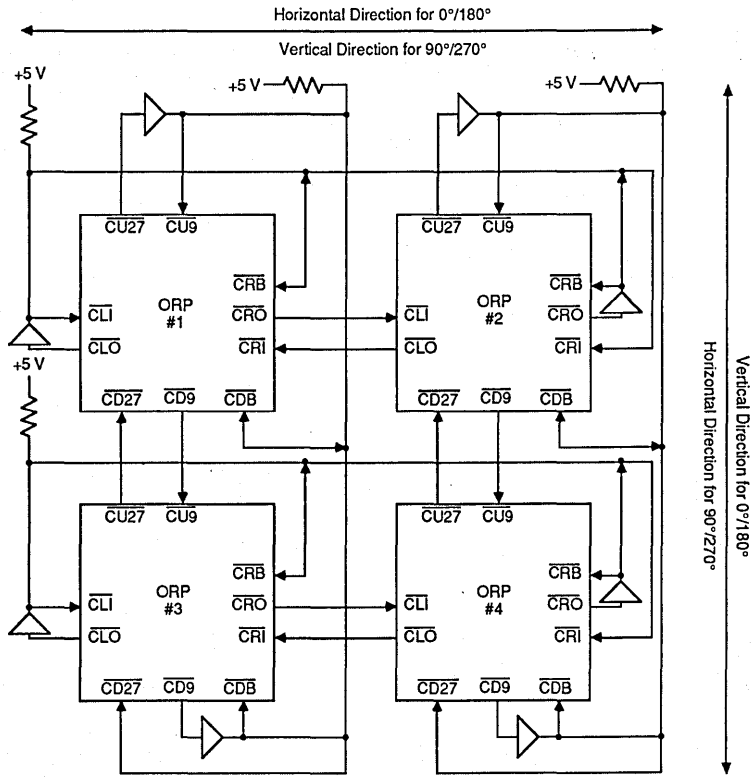
If the size of the image block is smaller than the cascade configuration — for instance 64 x 128 (in a 128 x 128 system) — the second and fourth ORPs will have to be programmed to the desired state. The EN bit of each should be set to 0. Under this condition $\overline{CLI}$ will be internally connected to $\overline{CRB}$ for the second and fourth ORP, preventing their FC bits from being set.

0-Degree Read Operation

In the 0-degree read operation, the FC bit propagation is the same as that of the write operation.

90-Degree Read Operation

During a 90-degree read operation, the $\overline{CLO}$ and $\overline{CRI}$ are used to propagate the FC bit in the vertical direction. These signals correspond to $\overline{CD9}$ and $\overline{CU9}$ in the write operation. $\overline{CDB}$ is now defined as an output. In the horizontal direction, $\overline{CD9}$ is driven LOW after all (XAR + 1) words are read out from the chip. $\overline{CD9}$ is connected to the lower ORP's $\overline{CU9}$. The lowest ORP's $\overline{CD9}$ must be fed back to the $\overline{CU9}$ at the top ORP through the open-drain buffer. $\overline{CDB}$ in each ORP must be connected together



09332B-12

Figure 7. Cascade Connection

to $\overline{CU9}$ at the top ORP. $\overline{CDB}$ is an open-drain output and needs a pull-up resistor. The FC bit in the second ORP will be set to 1 by the CPU. The other ORP's FC bit must be set to 0. The read operation will begin with ORP #2. The horizontal direction FC bit propagation will be done between the second and fourth ORP. When the entire font section assigned to the ORP is read out, $\overline{CU9}$ is internally connected to $\overline{CLO}$. When the final data in ORP #4 is read out, $\overline{CU9}$ of the second ORP will be driven LOW, causing ORP #2 to drive $\overline{CLO}$ LOW, resulting in the FC bit of the first ORP to be set to 1. Then the horizontal direction FC bit propagation will continue between the first and the third ORPs.

180-Degree Read Operation

In a 180-degree read operation, $\overline{CU27}$ and $\overline{CD27}$ are used to propagate the FC bit in the vertical direction. $\overline{CRB}$ is defined as an input. $\overline{CRI}$, $\overline{CLO}$, and $\overline{CRB}$ are used to propagate the FC bit in the horizontal direction. In this operation, the bottom-right ORP (ORP #4) needs to be programmed with the FC bit equal to 1. The other ORP's FC bit must be set to 0. The horizontal direction FC bit propagation will be done by $\overline{CLO}$ and $\overline{CRI}$ according to the XAR bits. The left-most ORP (ORP #3) must have $\overline{CLO}$ fed back to ORP #4's $\overline{CRI}$ and $\overline{CRB}$ through an open-drain buffer. The $\overline{CRB}$ input is effective only when that chip is in disable state (EN = 0). The ORP whose EN bit is set to 1 uses the $\overline{CRI}$ as a cascade input.

When the entire font section assigned to the ORP is read out, the $\overline{CRI}$ will be connected to $\overline{CU27}$ internally. When final data is read from ORP #3, ORP #4 will drive $\overline{CU27}$ LOW according to $\overline{CRI}$. The FC bit is then transferred to ORP #2. Then the horizontal direction FC bit propagation will be done between ORP #2 and ORP #1.

If the Y size of the image block is smaller than the cascade configuration, ORPs #3 and #4 will have to be programmed to a disabled state. The EN bit of these ORPs has to be set to 0. Under this condition, $\overline{CRB}$ will be connected directly to $\overline{CLO}$ internally.

270-Degree Read Operation

During a 270-degree read operation, the $\overline{CRO}$ and $\overline{CLI}$ pins are used to propagate the FC bit in the vertical direction. $\overline{CDB}$ is defined as an input. $\overline{CU27}$, $\overline{CD27}$, and $\overline{CDB}$ are used to propagate the FC bit in the horizontal direction. In this operation, the bottom-left ORP (ORP #3) needs to be programmed with the FC bit equal to 1. The other ORP's FC bit must be set to 0. The horizontal direction FC bit propagation will be done by $\overline{CU27}$ according to the XAR bits. The top ORP (ORP #1) must have $\overline{CU27}$ fed back to ORP #3's $\overline{CD27}$ and $\overline{CDB}$ through an open-drain buffer. The $\overline{CDB}$ input is effective only when that chip is in the disable state (EN = 0). The ORP whose EN bit is set to 1 uses the $\overline{CD27}$ pin as a cascade input.

When the entire font section assigned to the ORP is read out, the $\overline{CD27}$ input will be connected to the $\overline{CRO}$ output internally. When the final data is read from ORP #1, ORP #3 will drive $\overline{CRO}$ LOW according to $\overline{CD27}$. The FC bit is then transferred to ORP #4. The horizontal direction FC bit propagation will then be done between ORP #4 and ORP #2.

If the size of the image block is smaller than the cascade configuration, ORPs #3 and #4 will have to be programmed to a disabled state. The EN of these ORPs has to be set to 0. Under this condition, $\overline{CDB}$ will be connected directly to $\overline{CU27}$ internally.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65 to +150°C
Maximum V_{CC} Relative to V_{SS} ..... –0.3 to +7.0 V
DC Voltage Applied to Any
Pin Relative to V_{SS} ..... –0.5 to V_{CC} + 0.3 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IL}	Input LOW Voltage		–0.5	+0.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + 0.5	V
V_{OL}	Output LOW Voltage	I_{OL} = 3.2 mA		+0.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = –400 µA	2.4		V
I_{OZ}	Output Leakage Current	.45 < V_{OUT} < V_{CC}		±10	µA
I_I	Input Current	0 < V_{IN} < V_{CC}		±10	µA
I_{CC}	Power Supply Current			50	mA

PRELIMINARY

CAPACITANCE*

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
C_{IN}	Input Capacitance	f = 1 MHz		15	pF
$C_{I/O}$	Bidirectional Pin Capacitance	f = 1 MHz		20	pF
C_{OUT}	Output Pin Capacitance	f = 1 MHz		20	pF
C_L	Output Load Capacitance			100	pF

* Parameters are not tested.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
HOST INTERFACE TIMING					
1	t_{ARC}	$\overline{RD}/\overline{WR}$ ↑ to ST ↑, Access Recovery Time	30		ns
2	t_{STW}	ST ↑ to ST ↓, ST Pulse Width	30		ns
3	t_{GSFBS}	$\overline{GS}/\overline{FB}$ ↓ to $\overline{RD}/\overline{WR}$ ↓, $\overline{GS}/\overline{FB}$ Setup Time	10		ns
4	t_{GSFBH}	$\overline{RD}/\overline{WR}$ ↑ to $\overline{GS}/\overline{FB}$ ↑, $\overline{GS}/\overline{FB}$ Hold Time	0		ns
5	t_{CSS}	$\overline{CS}$ ↓ to $\overline{RD}/\overline{WR}$ ↓, $\overline{CS}$ Setup Time	10		ns
6	t_{CSH}	$\overline{RD}/\overline{WR}$ ↑ to $\overline{CS}$ ↑, $\overline{CS}$ Hold Time	20		ns
7	t_{WACCS}	ST ↓ to $\overline{WR}$ ↓, Write Access Time	40		ns
7A	t_{FWACCS}	ST ↓ to $\overline{RD}$ ↓, Fly-By Write Access Time	40		ns
8	t_{WDIS}	Data-In Setup Time to $\overline{WR}$ ↑	20		ns
8A	t_{FWDIS}	Fly-By Data-In Setup Time to $\overline{RD}$ ↑	20		ns
9	t_{WDIH}	Data-In Hold Time After $\overline{WR}$ ↑	10		ns
9A	t_{FWDIH}	Fly-By Data-In Hold Time After $\overline{RD}$ ↑	10		ns
10	t_{RACCS}	ST ↓ to $\overline{RD}$ ↓, Read Access Time	40		ns
10A	t_{FRACCS}	ST ↓ to $\overline{WR}$ ↓, Fly-By Read Access Time	40		ns
11	t_{RDOV}	$\overline{RD}$ ↓ to Data-Out Valid Delay		35	ns
11A	t_{FRDOV}	$\overline{WR}$ ↓ to Fly-By Data-Out Valid Delay		35	ns
12	t_{RDOZ}	$\overline{RD}$ ↑ to Data-Out Invalid Delay	0	30	ns
12A	t_{FRDOZ}	$\overline{WR}$ ↑ to Fly-By Data-Out Invalid Delay	0	30	ns
13	t_{RD}	$\overline{RD}$ Signal Pulse Width	50		ns
14	t_{WR}	$\overline{WR}$ Signal Pulse Width	50		ns
23	t_{RESET}	$\overline{RESET}$ Pulse Width	200		ns
ORP CASCADE INTERFACE TIMING					
15	t_{CAV}	$\overline{RD}/\overline{WR}$ ↓ to $\overline{CLO}/\overline{CRO}/\overline{CU27}/\overline{CD9}$ ↓ Cascade Output Valid Delay		55	ns
16	t_{CAIV}	ST ↓ to $\overline{CLO}/\overline{CRO}/\overline{CU27}/\overline{CD9}$ ↑ Cascade Output Invalid Delay		30	ns
17	t_{SCA}	$\overline{CLI}/\overline{CRI}/\overline{CU9}/\overline{CD27}/\overline{CDB}/\overline{CRB}$ ↓ to ST ↓ Cascade Input Setup Time	10		ns
18	t_{HCA}	ST ↓ to $\overline{CLI}/\overline{CRI}/\overline{CU9}/\overline{CD27}/\overline{CDB}/\overline{CRB}$ ↑ Cascade Input Hold Time	10		ns

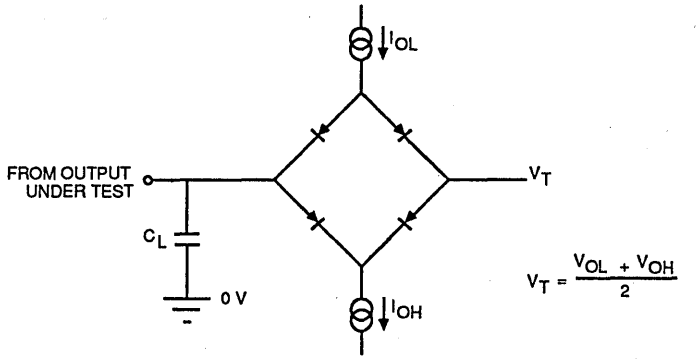
PRELIMINARY

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
ORP CASCADE INTERFACE TIMING (Cont'd.)					
19	t_{BCAL}	$\overline{CRB}\downarrow$ to $\overline{CLO}\downarrow$ (180° Read), $\overline{CDB}\downarrow$ to $\overline{CU27}\downarrow$ (270° Read)		25	ns
19A	t_{ODBCAL}	$\overline{CU9}\downarrow$ to $\overline{CDB}\downarrow$ (90° Read), $\overline{CLI}\downarrow$ to $\overline{CRB}\downarrow$ (Write, 0° Read) O.D. ORP Disabled Bypass Cascade Valid Delay		25	ns
20	t_{BCAH}	$\overline{CRB}\uparrow$ to $\overline{CLI}\uparrow$ (180° Read), $\overline{CDB}\uparrow$ to $\overline{CU27}\uparrow$ (270° Read) ORP Disabled Bypass Cascade Invalid Delay		25	ns
20A	t_{ODBCAH}	$\overline{CU9}\uparrow$ to $\overline{CDB}\uparrow$ (90° Read), $\overline{CLI}\uparrow$ to $\overline{CRB}\uparrow$ (Write, 0° Read) O.D. ORP Disabled Bypass Cascade Invalid Delay		25	ns
21	t_{FLBCAL}	$\overline{CLI}\downarrow$ to $\overline{CD9}\downarrow$, $\overline{CU9}\downarrow$ to $\overline{CLO}\downarrow$, $\overline{CRI}\downarrow$ to $\overline{CU27}\downarrow$, $\overline{CD27}\downarrow$ to $\overline{CRO}\downarrow$ ORP Font Section Complete Bypass Cascade Valid Delay		25	ns
22	t_{FLBCAH}	$\overline{CLI}\uparrow$ to $\overline{CD9}\uparrow$, $\overline{CU9}\uparrow$ to $\overline{CLO}\uparrow$, $\overline{CRI}\uparrow$ to $\overline{CU27}\uparrow$, $\overline{CD27}\uparrow$ to $\overline{CRO}\uparrow$ ORP Font Section Complete Bypass Cascade Invalid Delay		25	ns

PRELIMINARY

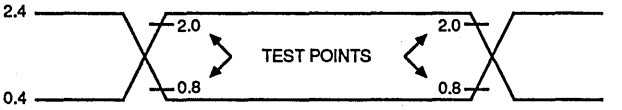
5

SWITCHING TEST CIRCUIT
(Standard Load)



09332B-13

SWITCHING TEST WAVEFORM
(Input)



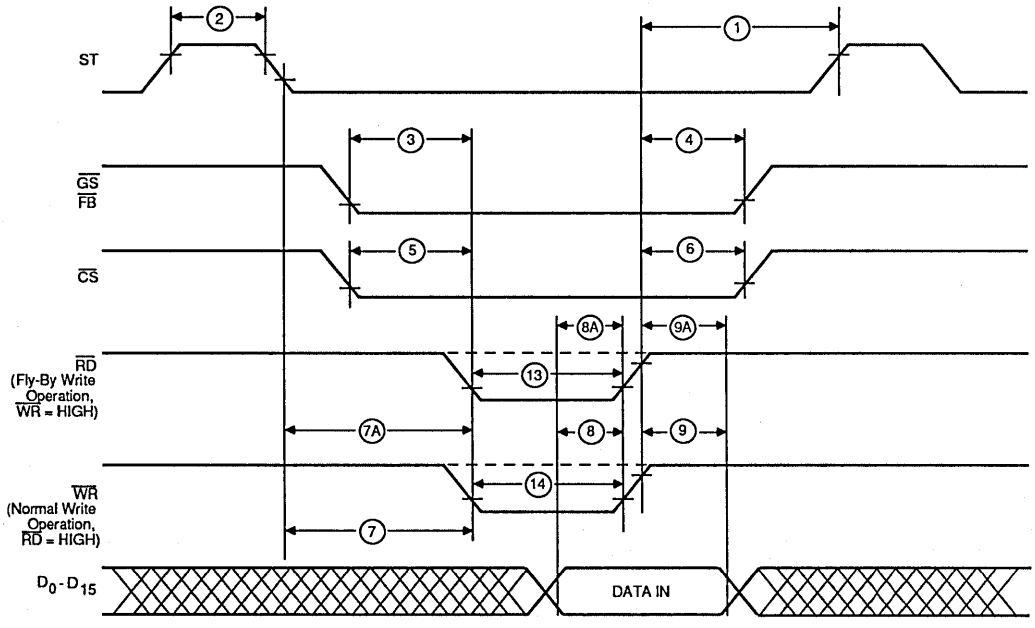
09332B-14

SWITCHING WAVEFORMS
KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE, ANY CHANGE PERMITTED	CHANGING, STATE UNKNOWN

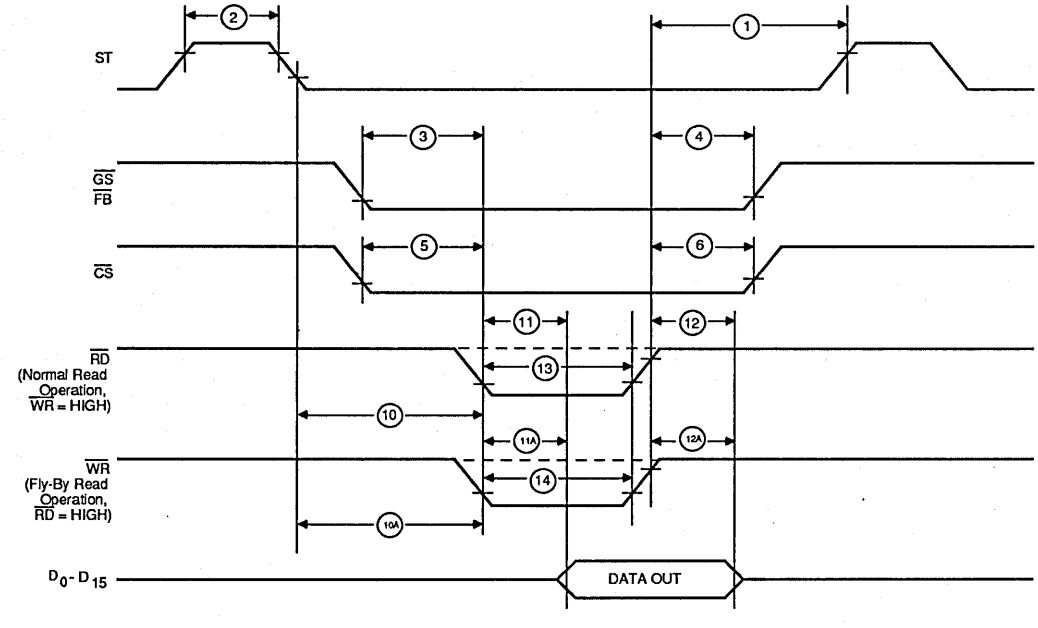
KS000010

SWITCHING WAVEFORMS (Cont'd.)



09332B-15

ORP Write Operation Timing

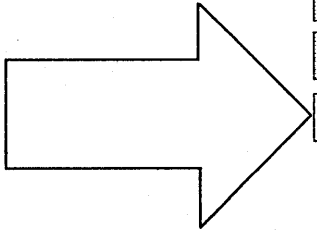


09332B-16

ORP Read Operation Timing

5

Advanced
Micro
Devices



CHAPTER 6
GENERAL INFORMATION

THERMAL RESISTANCE OF AMD PRODUCTS

Device	Package	Process	Gates[1] (Equiv.)	Area (KSq Mils)	Power[2] (mW)	θ_{JA}[3] (C/W)	θ_{JC}[3] (C/W)
80286	CA2068	NMOS	40.0K	98.0	3000	31	8
	CGX068	NMOS	40.0K	98.0	3000	37	2
80L286	PL 068	NMOS	40.0K	98.0	2250	34	
8086	CD 040	NMOS	9.6K	52.9	1700	32	7
	PD 040	NMOS	9.6K	52.9	1700	44	
	PL 044	NMOS	9.6K	52.9	1700	47	
8088	CD 040	NMOS	9.6K	36.1	1700	34	7
	PD 040	NMOS	9.6K	36.1	1700	44	
	PL 044	NMOS	9.6K	36.1	1700	47	
8237A	CD 040	NMOS	2.2K	30.6	750	34	7
	PD 040	NMOS	2.2K	30.6	750	44	
Z8530H	CD 040	NMOS	5.0K	48.8	1250	34	7
	PD 040	NMOS	5.0K	48.8	1250	44	
	PL 044	NMOS	5.0K	48.8	1250	47	
9513A	CD 040	NMOS	4.5K	39.3	1375	34	7
	PD 040	NMOS	4.5K	39.3	1375	44	
	CL 044	NMOS	4.5K	39.3	1375	47	7
	PL 044	NMOS	4.5K	39.3	1375	47	
9516A	SD 048	NMOS	3.7K	52.4	1750	35	10
	PD 048	NMOS	3.7K	52.4	1750	50	
Am9517A	CD 040	NMOS	2.2K	30.6	750	34	7
	PD 040	NMOS	2.2K	30.6	750	44	
	PL 044	NMOS	2.2K	30.6	750	42	
9519A	CD 028	NMOS	1.5K	39.3	725	37	9
	PD 028	NMOS	1.5K	39.3	725	47	
Am9580A	CA2068	NMOS	150.0K	126.2	3500	31	8
Am95C85	PL 044	CMOS	53.3K	98.6	1000		
Am7971A	PL 068	NMOS	40.0K	114.7	3000	38	
	CA2068	NMOS	40.0K	114.7		35	6
	CGX068	NMOS	40.0K	114.7		23	
Am95C71	PL 068	CMOS	31.3K	100.1	750	35	
Am95C75	PL 084	CMOS	9.4K	59.0	350	35	
Am95C76	CD 040	CMOS	12.0K	57.0	250	47	4
	PD 040	CMOS	12.0K	57.0	250	47	
	PL 044	CMOS	12.0K	57.0	250	47	

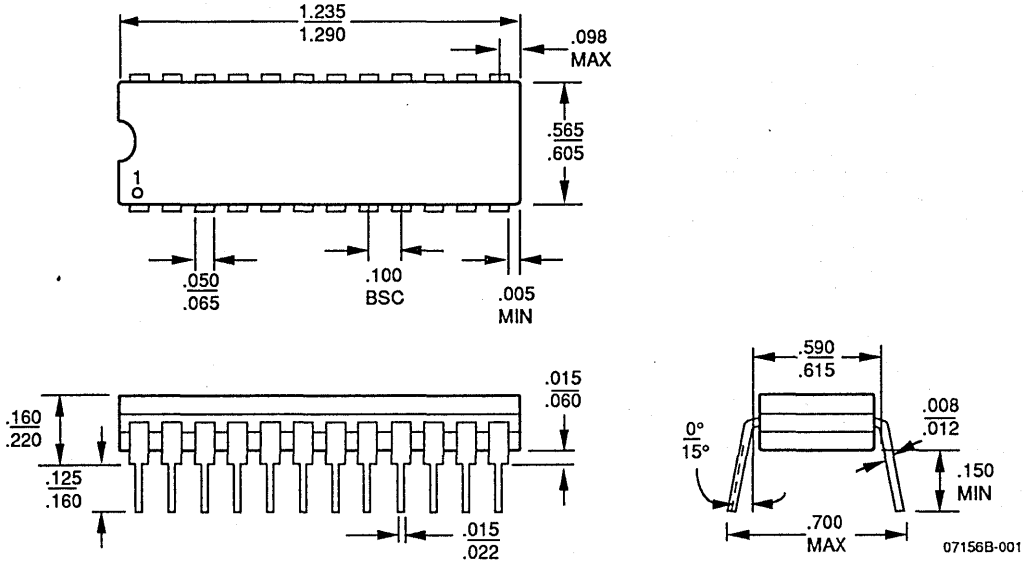
THERMAL RESISTANCE (continued)

Device	Package	Process	Gates[1] (Equiv.)	Area (KSq Mils)	Power[2] (mW)	θ_{JA}[3] (C/W)	θ_{JC}[3] (C/W)
Am95C94	PL 068	CMOS	30.0K	112.7	275	35	
Am95C95	PL 068	CMOS	35K	140	470	35	
Am95C96	PL 084	CMOS	35K	140	470	35	
Am33C93A	PD 040	CMOS	12.5K	38.2	500	49	
	PL 044	CMOS					
Z8530H	PD 040	NMOS	27K	51	1250	44	
	PL 044	NMOS	27K	51	1250	47	
Z85C30	PD 044	CMOS	27K	53	110	32	
	PL 044	CMOS	27K	53	110	30	
	CD 040	CMOS	27K	53	110	30	4
	CL 044	CMOS	27K	53	110	33	6
Am29C325	CG145	CMOS	43K	128	1470	23	5
Am29C327	CG169	CMOS	230K	250	2650	20	4
Am5380	CD 040	NMOS	1.3K	54	725	35	
	PD 040	NMOS	1.3K	54	725	45	7
	PL 044	NMOS	1.3K	54	725	30	
Am53C80N	PD 040	CMOS	780	27	300	41	7
	PL 044	CMOS	780	27	300	44	
Am53C80	PD 048	CMOS	780	22	300	35	7
	PL 044	CMOS	780	22	300	44	

Notes:
1. The gate counts are only an approximation (total devices / 3).
2. Power is the highest ICC over the temperature times the nominal power supply voltage.
3. θ_{JA} and θ_{JC} values were estimated by extrapolating measured data. Values are to be considered as "typical" for the device.
4. There is no accepted industry definition for QJC for molded plastic packages at this time.
5. CMOS data is not available.

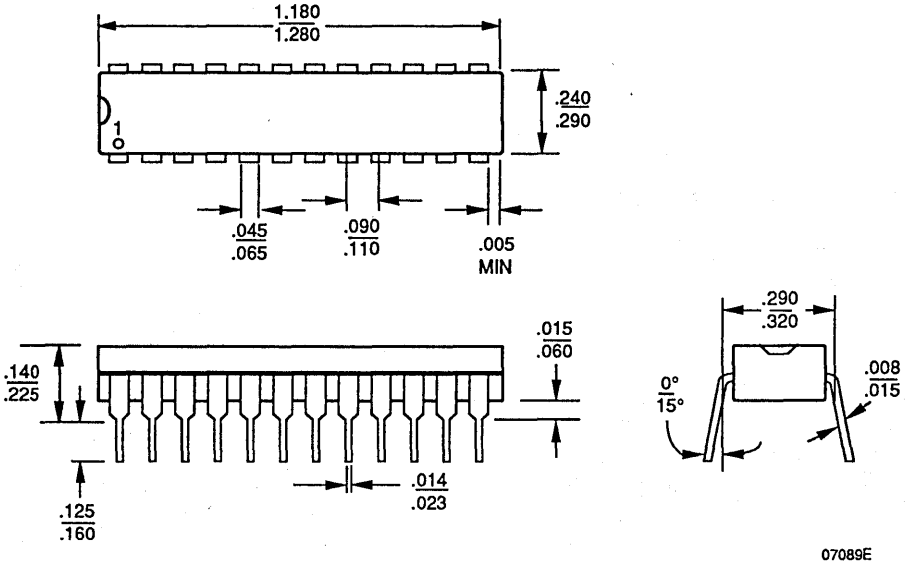
PHYSICAL DIMENSIONS

CD 024

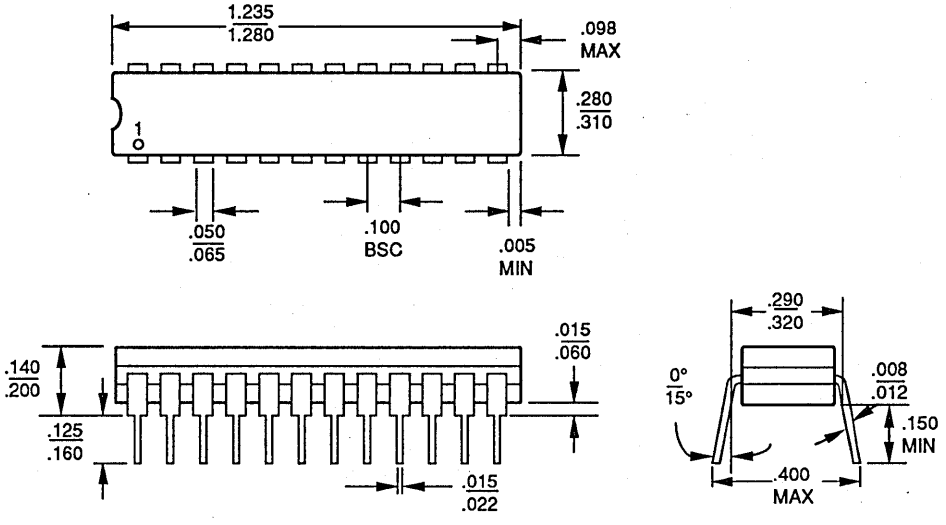


6

PD 3024


1.180
1.280
.240
.290
1
.045
.065
.090
.110
.005
MIN
.140
.225
.015
.060
.125
.160
.014
.023
.290
.320
0°
15°
.008
.015
07089E

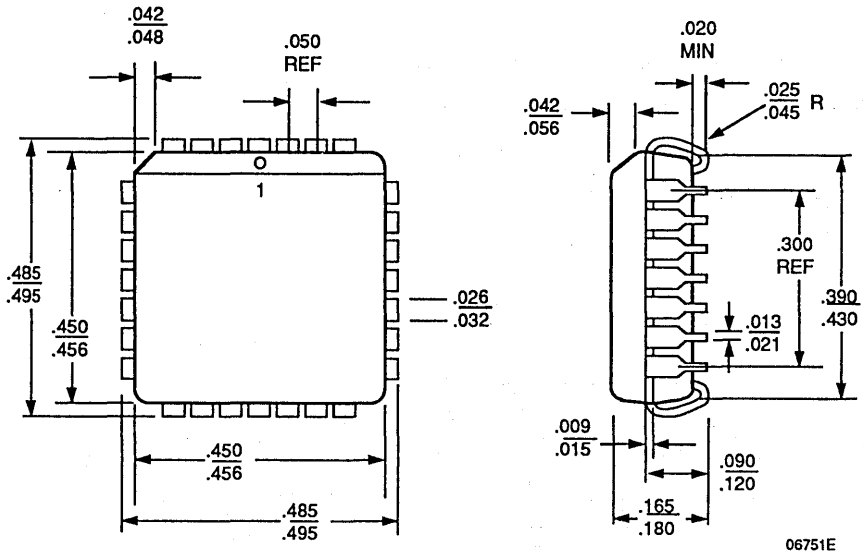
CD 3024



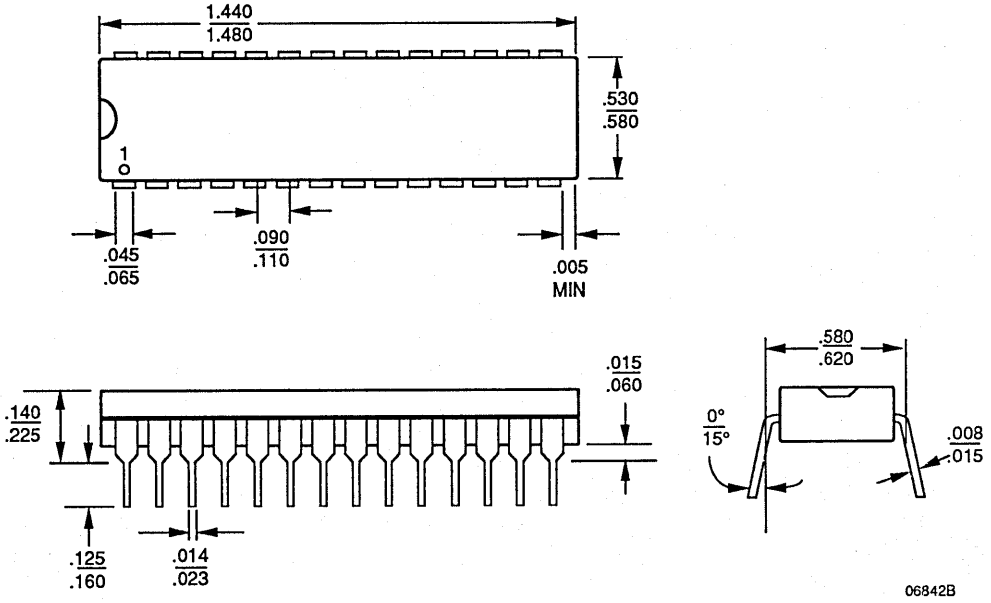
06850C

6

PL 028

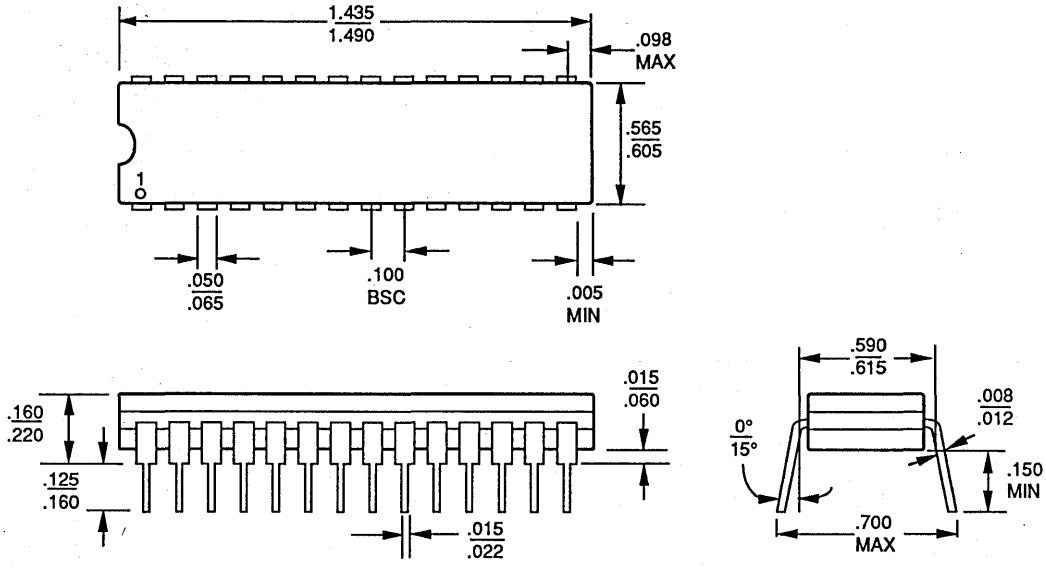

.042
.048
.050
REF
1
.485
.495
.450
.456
.026
.032
.450
.456
.485
.495
.020
MIN
.042
.056
.025
.045
R
.300
REF
.390
.430
.013
.021
.009
.015
.090
.120
.165
.180
06751E

PD 028

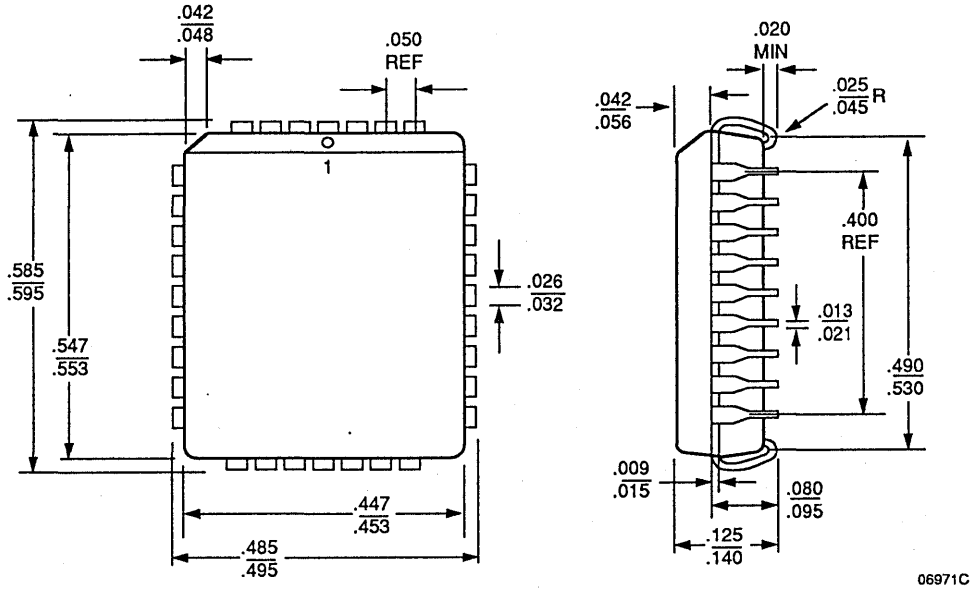

1.440
1.480
.530
.580
1
.045
.065
.090
.110
.005
MIN
.015
.060
.140
.225
.125
.160
.014
.023
.580
.620
0°
15°
.008
.015
06842B

6

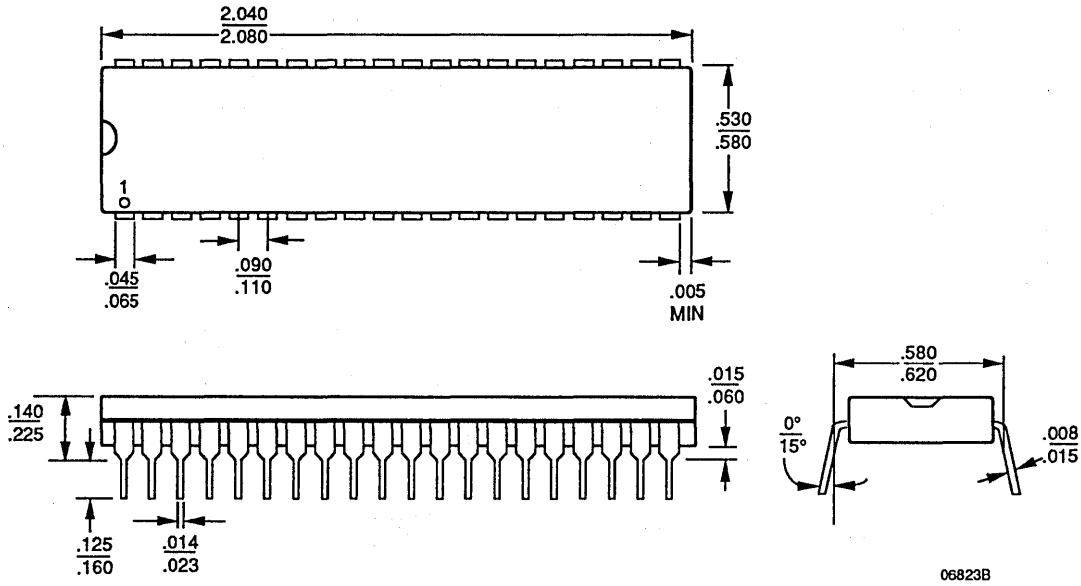
CD 028


1.435
1.490
.098
MAX
.565
.605
1
.050
.065
.100
BSC
.005
MIN
.160
.220
.125
.160
.015
.060
.015
.022
.590
.615
.008
.012
0°
15°
.150
MIN
.700
MAX

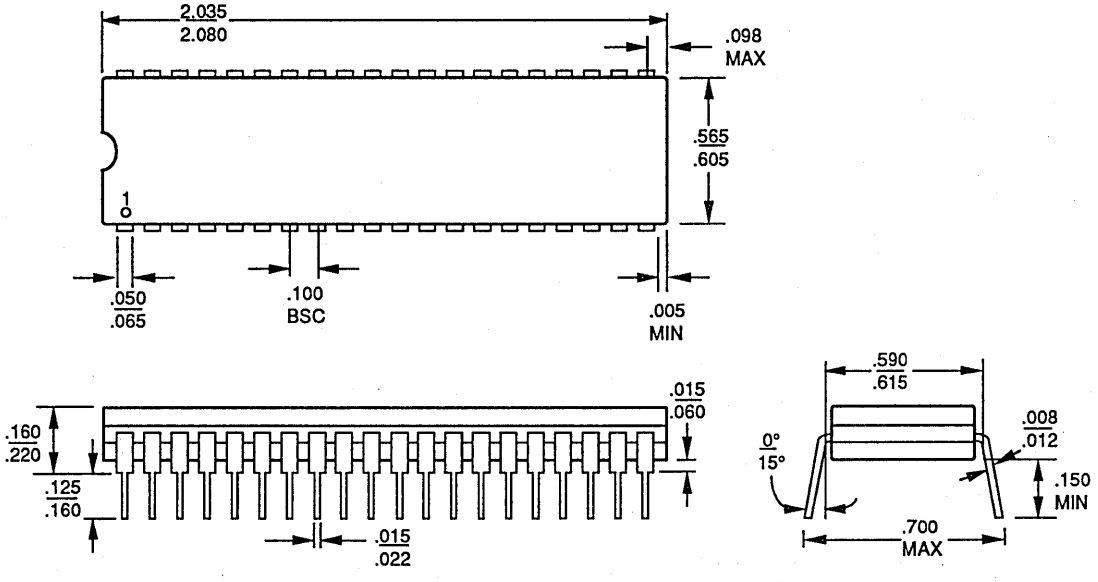
PL 032


.042
.048
.050
REF
1
.585
.595
.547
.553
.026
.032
.447
.453
.485
.495
.020
MIN
.042
.056
.025
.045
R
.400
REF
.013
.021
.490
.530
.009
.015
.080
.095
.125
.140
06971C

PD 040

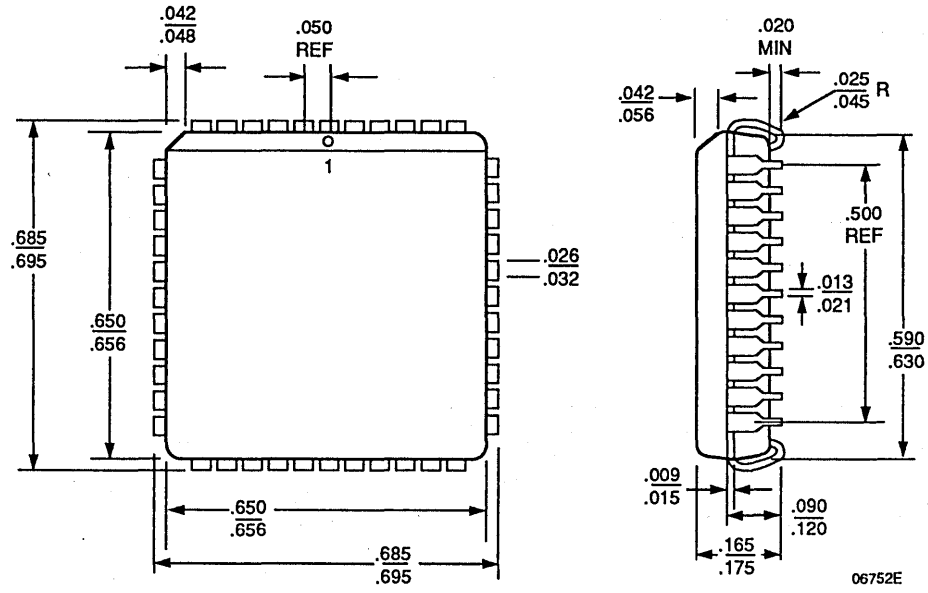

2.040
2.080
.530
.580
1
.045
.065
.090
.110
.005
MIN
.015
.060
.140
.225
.125
.160
.014
.023
.580
.620
0°
15°
.008
.015
06823B

CD 040

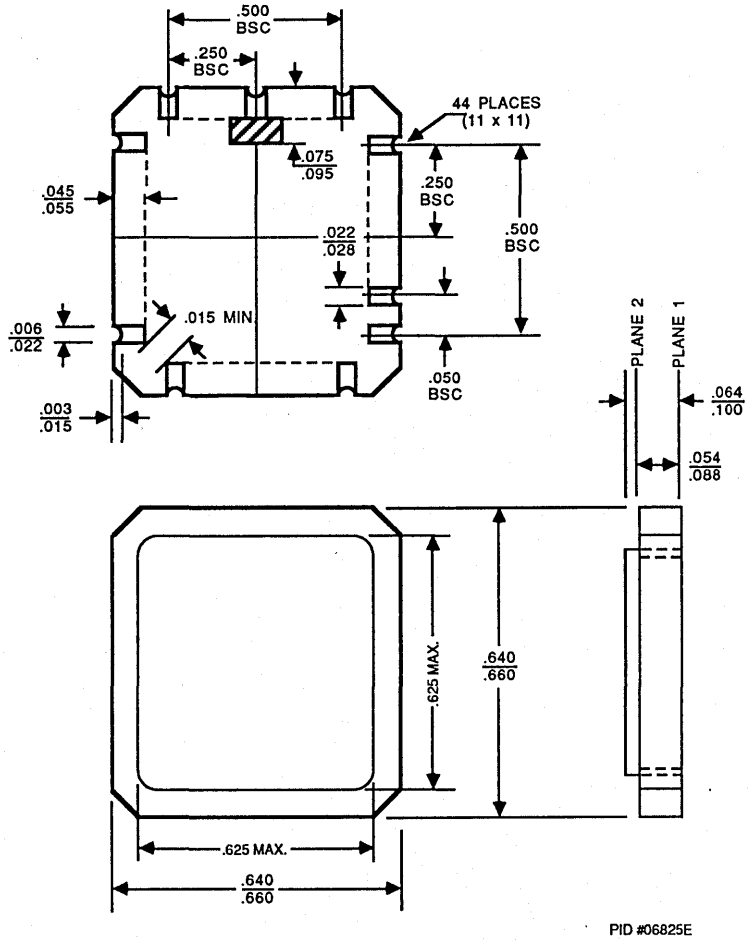

2.035
2.080
.098
MAX
.565
.605
1
.050
.065
.100
BSC
.005
MIN
.015
.060
.160
.220
.125
.160
.015
.022
.590
.615
.008
.012
0°
15°
.150
MIN
.700
MAX
06824C

6

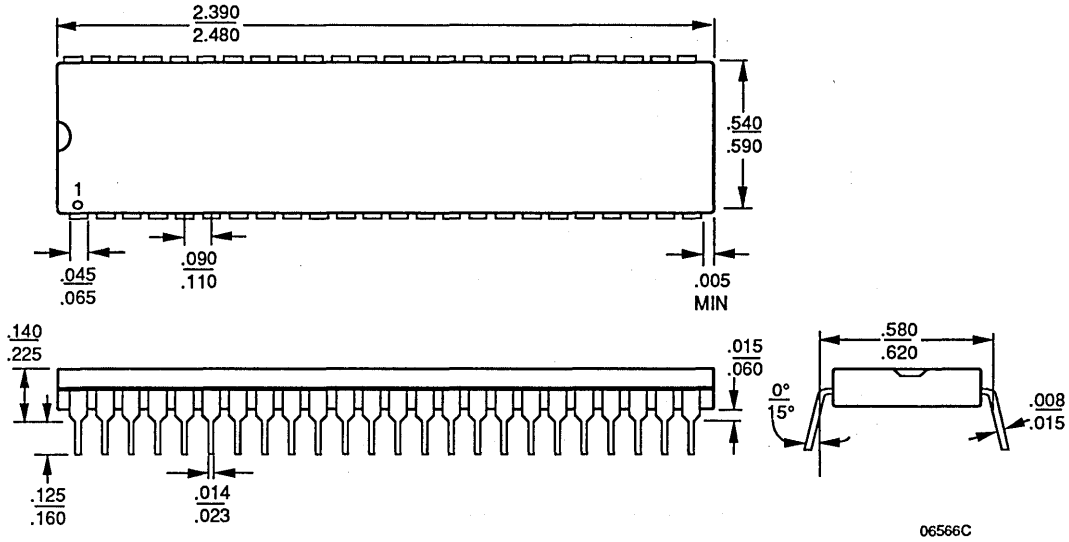
PL 044


.042
.048
.050
REF
1
.685
.695
.650
.656
.026
.032
.650
.656
.685
.695
.020
MIN
.042
.056
.025
.045
R
.500
REF
.013
.021
.590
.630
.009
.015
.090
.120
.165
.175
06752E

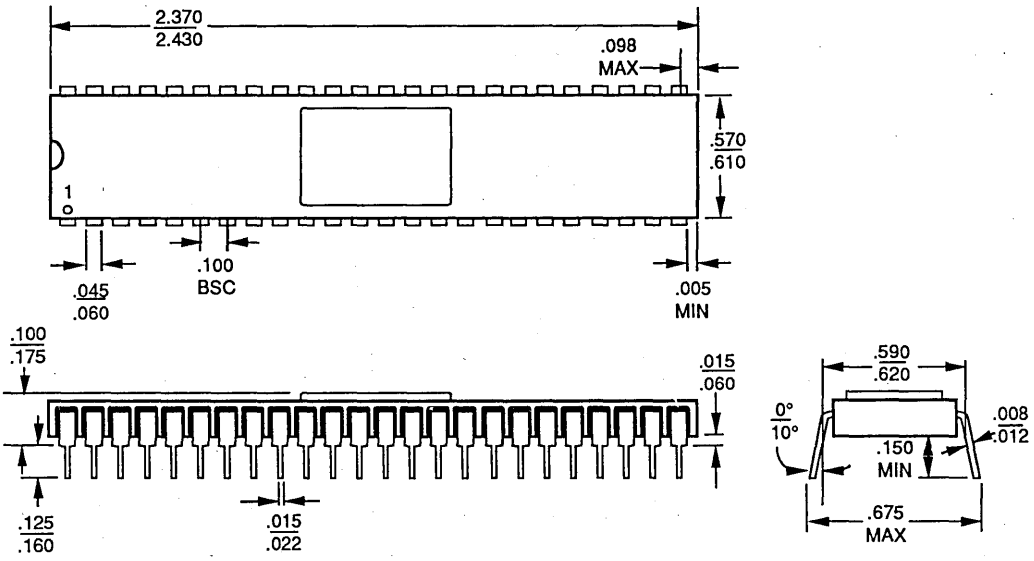
CL 044


.500
BSC
.250
BSC
44 PLACES
(11 x 11)
.075
.095
.250
BSC
.500
BSC
.045
.055
.022
.028
.015 MIN.
.006
.022
.050
BSC
.003
.015
PLANE 2
PLANE 1
.064
.100
.054
.088
.625 MAX.
.640
.660
.625 MAX.
.640
.660
PID #06825E

PD 048


2.390
2.480
.540
.590
1
.045
.065
.090
.110
.005
MIN
.140
.225
.015
.060
.125
.160
.014
.023
.580
.620
0°
15°
.008
.015
06566C

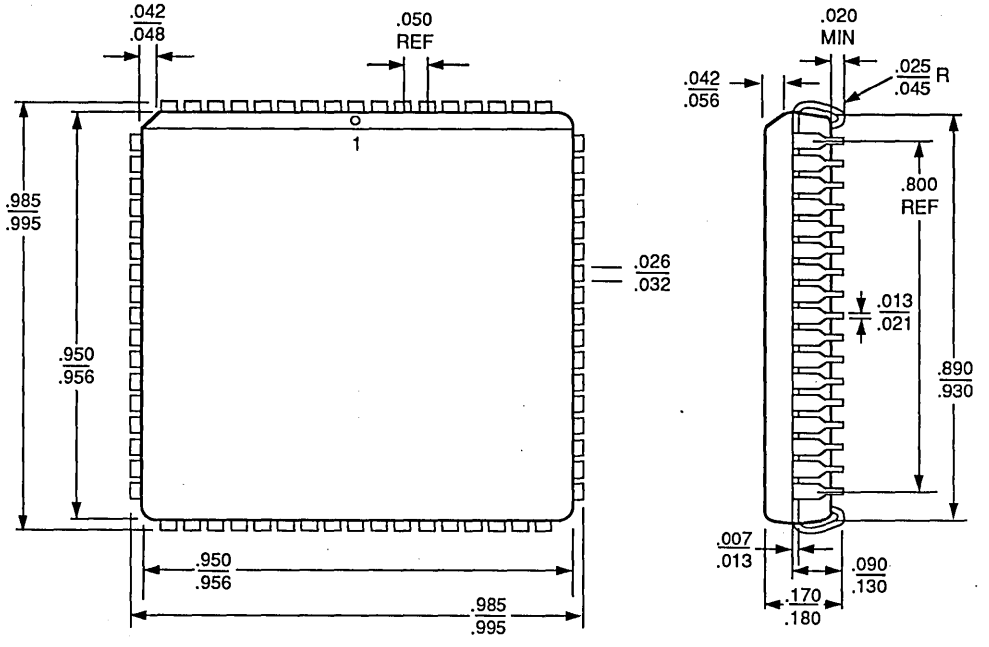
SD 048



07644B

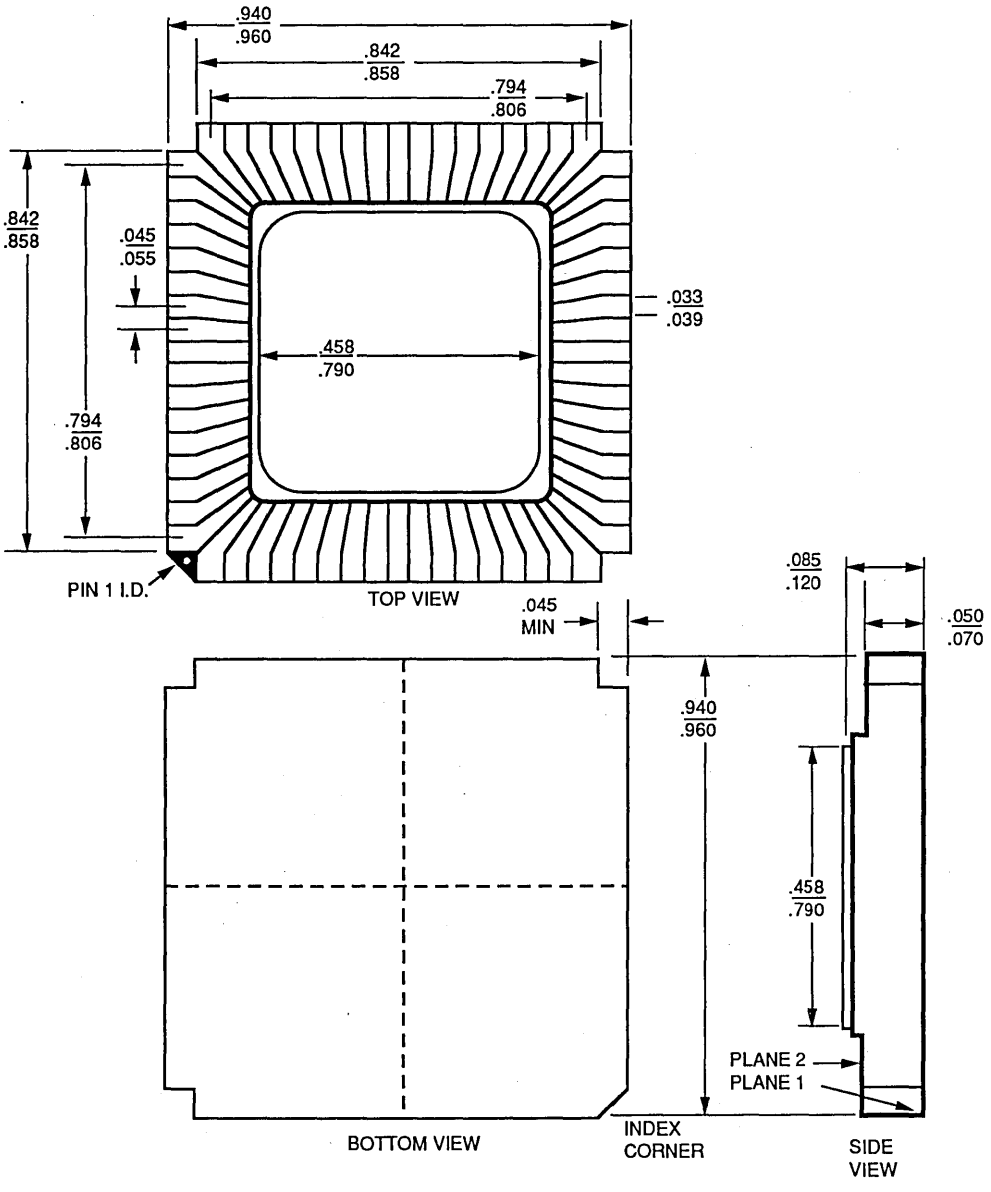
6

PL 068

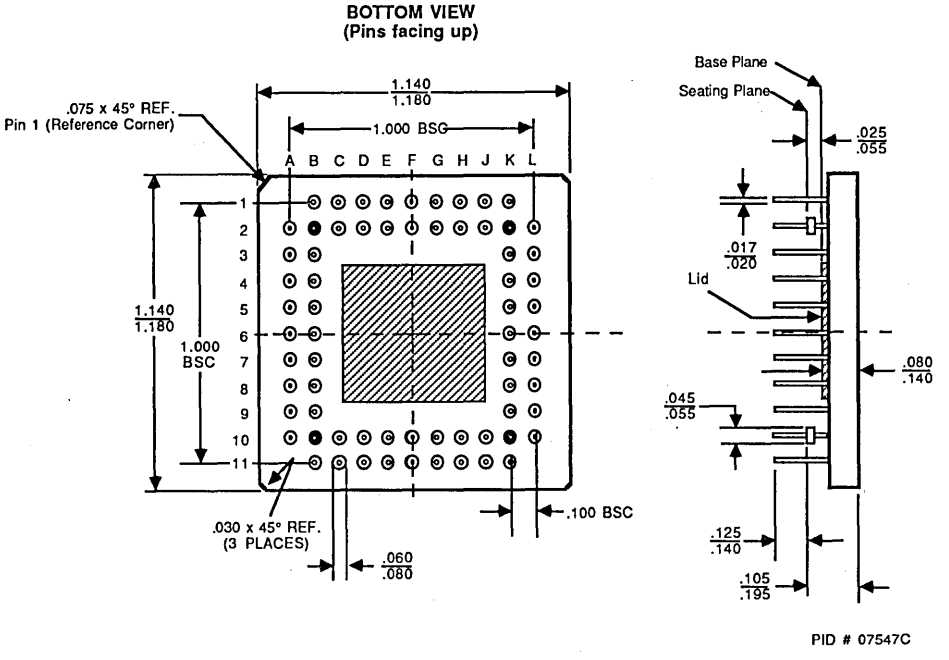


06753J

CA 2068

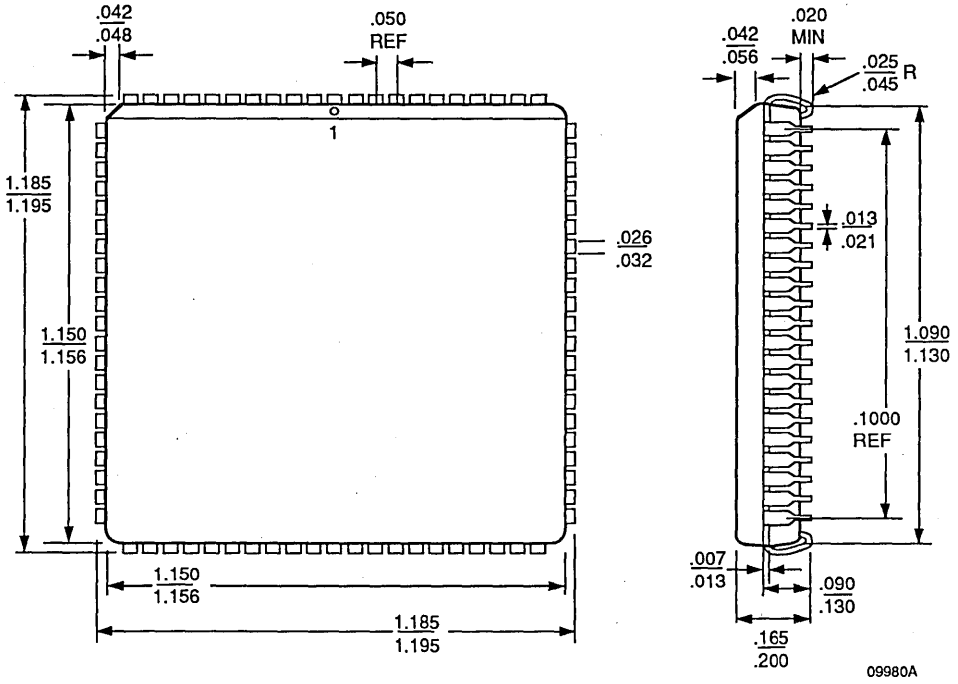

.940
.960
.842
.858
.794
.806
.842
.858
.045
.055
.033
.039
.458
.790
.794
.806
PIN 1 I.D.
TOP VIEW
.045
MIN
.085
.120
.050
.070
.940
.960
.458
.790
PLANE 2
PLANE 1
BOTTOM VIEW
INDEX
CORNER
SIDE
VIEW

CGX 068

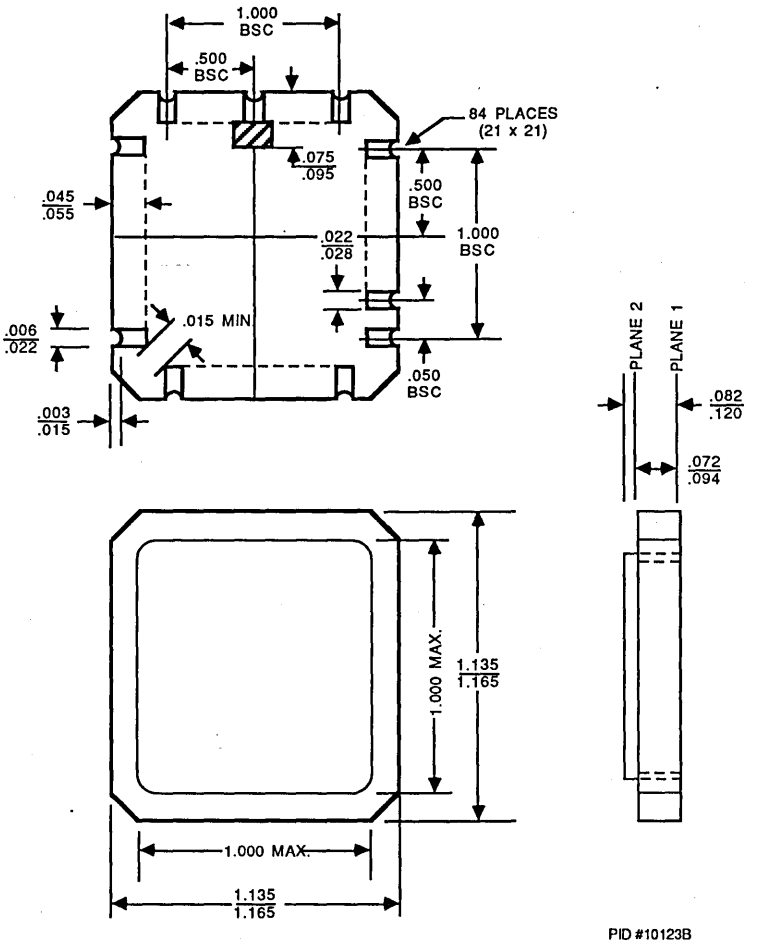


PID # 07547C

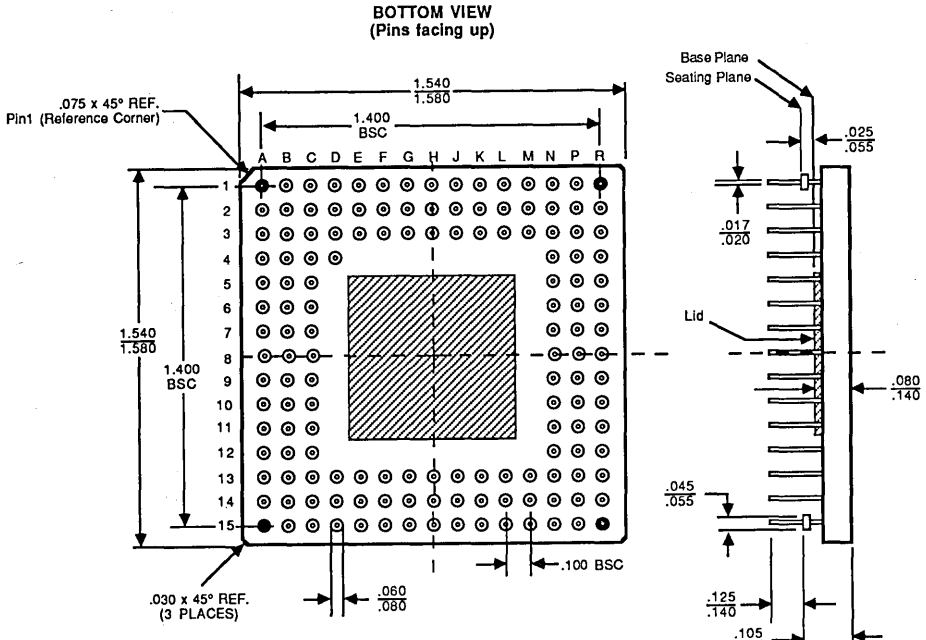
PL 084


.042
.048
.050
REF
1
1.185
1.195
.026
.032
1.150
1.156
1.150
1.156
1.185
1.195
.020
MIN
.042
.056
.025
.045
R
.013
.021
1.090
1.130
.1000
REF
.007
.013
.090
.130
.165
.200
09980A

CL 084

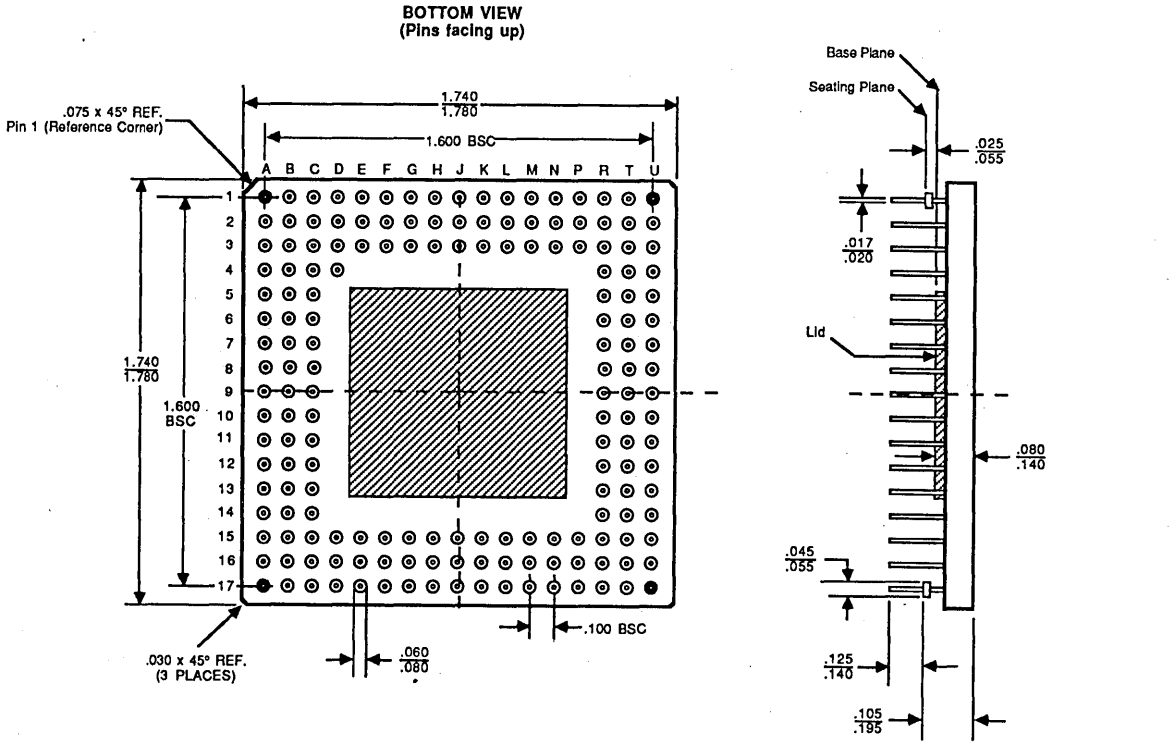

1.000
BSC
.500
BSC
84 PLACES
(21 x 21)
.075
.095
.500
BSC
.045
.055
.022
.028
1.000
BSC
.015 MIN
.006
.022
.050
BSC
.003
.015
PLANE 2
PLANE 1
.082
.120
.072
.094
1.000 MAX.
1.135
1.165
1.000 MAX.
1.135
1.165
PID #10123B

CGX 145



PID # 09691B

CGX 169

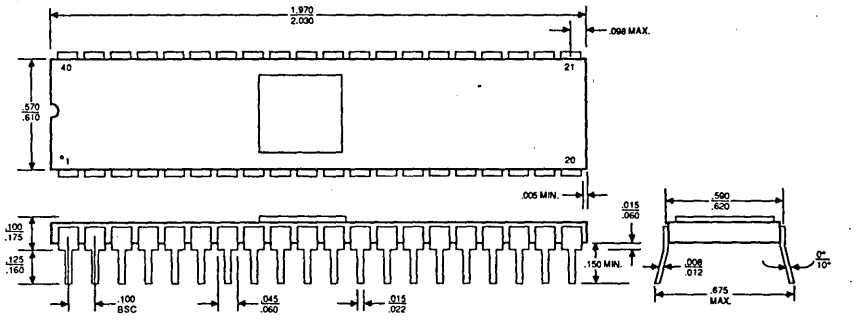


PID # 07322C

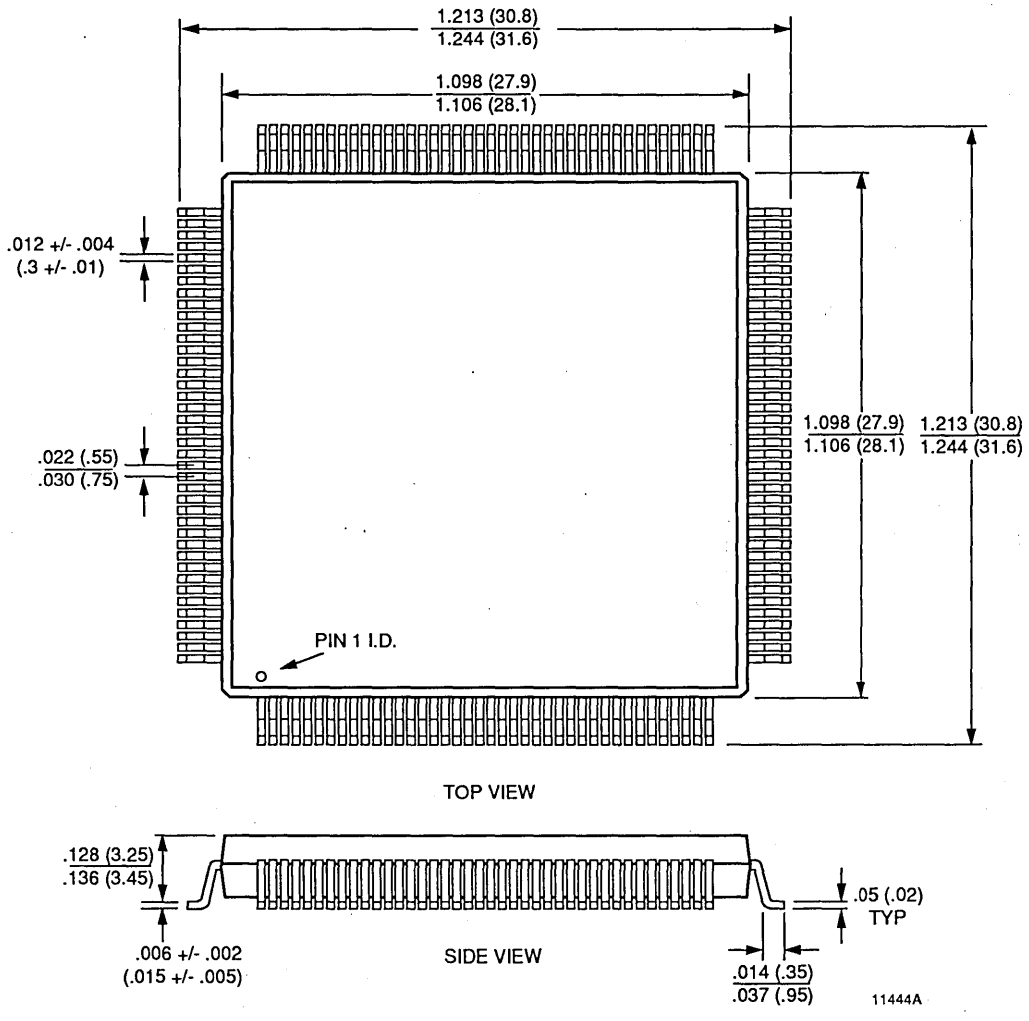
CGX 084*

* Package in development. Contact your local AMD Sales Representative for information.

SD 040


1.970
2.030
.098 MAX.
40
21
.570
.610
1
20
.005 MIN.
.015
.060
.590
.620
.100
.175
.125
.160
.150 MIN.
.008
.012
0°
10°
.675
MAX.
.100
BSC
.045
.060
.015
.022
PID # 07570B

PQJ 160


1.213 (30.8)
1.244 (31.6)
1.098 (27.9)
1.106 (28.1)
.012 +/- .004
(.3 +/- .01)
.022 (.55)
.030 (.75)
1.098 (27.9)
1.106 (28.1)
1.213 (30.8)
1.244 (31.6)
PIN 1 I.D.
TOP VIEW
.128 (3.25)
.136 (3.45)
.006 +/- .002
(.015 +/- .005)
SIDE VIEW
.05 (.02)
TYP
.014 (.35)
.037 (.95)
11444A

SUPPORT LITERATURE

The following is a list of AMD product literature which, in addition to the data sheets contained in this book, can be ordered from your local AMD Sales Representative or the Literature Distribution Center at (800) 538-8450, Extension 5000; inside California, call (408) 749-5000.

Key:
- AN = Application Note
- AR = Article Reprint
- HB = Handbook
- PD = Product Description
- TM = Technical Manual
- UM = User's Manual

Device	Order No.	Pub. Date	Title
Am29C325	09746A	(7/88)	Am29C325 CMOS 32-Bit Floating-Point Processor (PD)
	07171B	(9/88)	Am29C300 Brochure
	09856B	(10/88)	Am29C300 (AN)
	08191A	(9/86)	Am29C325 Floating-Point Processor—A Compendium of Technical Articles
Am29C327	07171B	(9/88)	Am29C300 Grid Brochure
	10028A	(9/88)	Double-Precision Floating-Point Processor (UM)
	10386A	(11/88)	Am29C327 Double-Precision Floating-Point Processor (PD)
	09856B	(10/88)	Am29C300 (AN)
	11601A	(2/89)	Am29C327/68020 (AN)
	10977A	(6/88)	"AMD's Floating-Point Chip Goes Beyond IEEE Specs," *Electronics* (AR)
	10887A	(6/88)	Am29C327 Reference Guyide
5380/C80N	09871A	(7/87)	"Build a High-Performance SCSI Bridge Controller," *Electronic Design* (AR)
7971A	04086C	(11/88)	Am7971A/Am95C71 VCEP (PD)
	09009A	(10/86)	"Compressing Data Conserves Memory in Bit-Mapped Display" *EDN* (AR)
	10343A	(11/87)	"Data-Compression Chip Eases Document-Processing Design," Computer Design (AR)
80C286	09729A	(5/89)	80C286 CMOS High-Speed Microprocessor (PD)
	11813A	(4/89)	80286 Article Reprints Volume 1 Brochure
	12591A	(9/89)	80286 Article Reprints Volume 2 Brochure
80286	10048A	(11/87)	Thermal Design Considerations for the 80286–16 (AN)
	11334A	(11/87)	Advanced Personal Computer Design Brochure
	11578A	(10/88)	"Death of 286 May Be Exaggerated," *MicroDesign Resources* (AR)
	11616A	(11/88)	"Plenty of Life in Old Chip," *New York Times* (AR)
	11813A	(4/89)	80286 Article Reprints Volume 1 Brochure
	12591A	(9/89)	80286 Article Reprints Volume 2 Brochure
	11127A	(11/88)	Personal Computer Decisions Brochure
	12173A	(5/89)	80286 OS/2 Benchmark Analysis Brochure

SUPPORT LITERATURE (continued)

Device	Order No.	Pub. Date	Title
80286	11623A	(11/88)	"16-bit Micros Fortify Their Positions Against 32-bit Intruders," *Computer Design* (AR)
	09920A	(4/88)	16MHz 80286 DRAM Interface for Fast Processors Without Caches Technical Paper
80C287	12190A/0	(3/89)	Comparing AMD's CMOS 80C287 and 80EC287 Math Coprocessors to the Intel NMOS 80287 and CMOS 80C287A (AN)
80EC287	12190A/0	(3/89)	Comparing AMD's CMOS 80C287 and 80EC287 Math Coprocessors to the Intel NMOS 80287 and CMOS 80C287A (AN)
8052	03893C	(3/88)	Advanced Bit-Mapped and Alphanumeric Display Products Brochure
	06178A	(4/85)	Am8052 Bus Interface Guide (AN)
81C451/458	03893C	(3/88)	Advanced Bit-Mapped and Alphanumeric Display Products Brochure
8151A	03893C	(3/88)	Advanced Bit-Mapped and Alphanumeric Display Products Brochure
	07837A	(5/86)	8151A (AN)
8152A/52B	03893C	(3/88)	Advanced Bit-Mapped and Alphanumeric Display Products Brochure
8172	03893C	(3/88)	Advanced Bit-Mapped and Alphanumeric Display Products Brochure
8177	03893C	(3/88)	Advanced Bit-Mapped and Alphanumeric Display Products Brochure
	07837A	(5/86)	A 1024 × 1280 Pixel Bit-Mapped Display with 16 Million Color Choices
Z8530H	03335A	(4/84)	Z8030/Z8530 (PD)
Z85C30	12035B	(5/89)	ISDN Solutions Brochure
	07513C	(4/89)	Z85C30 Serial Communications Controller (TM)
9513A	03402D	(12/84)	Am9513A/Am9513 System Timing Controller (TM)
9516A	04910A	(10/84)	Direct Memory Access Controller Am9516/ AmZ8016 (TM)
	03334B	(9/85)	Am9516A Universal DMA Controller (UDC) (PD)
	07737A	(2/86)	"High Performance DMA for VME," *Digital Design* (AR)
9517A/8237	00092C	1985	Multimode Direct Memory Access Controller—Am9517A (TM)

SUPPORT LITERATURE (continued)

Device	Order No.	Pub. Date	Title
9519A	05186B	(5/89)	Am9519A Universal Interrupt Controller (TM)
95C60	03893C	(3/88)	Advanced Bit-Mapped and Alphanumeric Display Products Brochure
	08395A	(3/88)	"Super Chips Bring Unprecedented Power to Graphics Display Control," *Computer Design* (AR)
	09848A		"Interface Helps Controller Boost Graphics Performance," *EDN* (AR)
	11268A	(7/88)	"Tapping the QPDM," *Computer Graphics World* (AR)
	09682A	(3/88)	Am95C60 Quad Pixel Dataflow Manager Applications Handbook
	07785C		Am95C60 Quad Pixel Dataflow Manager (TM)
	07777A	(7/87)	Am95C60 Quad Pixel Dataflow Manager (QPDM) (PD)
95C71	04086C	(11/88)	Am7971A/Am95C71 VCEP (PD)
95C75	09387A	(4/87)	Am95C75/Am95C76 Raster Printer Controller Orthogonal Rotation Processor (PD)
	12639A	(5/89)	29C75/76 Chip Set (AR)
95C76	09387A	(4/87)	Am95C75/Am95C76 Raster Printer Controller Orthogonal Rotation Processor (PD)
	12639A	(5/89)	29C75/76 Chip Set (AR)
9580A/90	09480A		Hard Disk Controller (TM)
	09871A	(8/87)	"Build a High-Performance SCSI Bridge Controller," *Electronic Design* (AR)
95C85	09676B	(7/88)	Am95C85 Content Addressable Data Manager (PD)
	08456A	(7/86)	"Hardware Sorting Chip Steps Up Software Pace," *Electronic Design* (AR)
	09491A	(5/87)	"Chip Accelerates Sorting, Searching," *EDN News* (AR)
	10650A	(2/88)	"Database Accelerator Targets Micro Channel Bus," *ESD* (AR)
	08035A	(4/88)	Am95C85 Content Addressable Data Manager (TM)
	11133B	(6/88)	"Intelligent Memory Architectures Attack Real-World Computation," *Computer Design* (AR)
	09848A	(4/87)	"Interface Helps Controller Boost Graphics Performance," *EDN* (AR)

Sales Offices

North American

ALABAMA		(205) 882-9122
ARIZONA		(602) 242-4400
CALIFORNIA,		
Culver City		(213) 645-1524
Newport Beach		(714) 752-6262
Roseville		(916) 786-6700
San Diego		(619) 560-7030
San Jose		(408) 452-0500
Woodland Hills		(818) 992-4155
CANADA, Ontario,		
Kanata		(613) 592-0060
Willowdale		(416) 224-5193
COLORADO		(303) 741-2900
CONNECTICUT		(203) 264-7800
FLORIDA,		
Clearwater		(813) 530-9971
Ft. Lauderdale		(305) 776-2001
Orlando (Casselberry)		(407) 830-8100
GEORGIA		(404) 449-7920
ILLINOIS,		
Chicago (Itasca)		(312) 773-4422
Naperville		(312) 505-9517
KANSAS		(913) 451-3115
MARYLAND		(301) 796-9310
MASSACHUSETTS		(617) 273-3970
MICHIGAN		(313) 347-1522
MINNESOTA		(612) 938-0001
NEW JERSEY,		
Cherry Hill		(609) 662-2900
Parsippany		(201) 299-0002
NEW YORK,		
Liverpool		(315) 457-5400
Poughkeepsie		(914) 471-8180
Rochester		(716) 272-9020
NORTH CAROLINA		(919) 878-8111
OHIO,		
Columbus (Westerville)		(614) 891-6455
Dayton		(513) 439-0470
OREGON		(503) 245-0080
PENNSYLVANIA		(215) 398-8006
SOUTH CAROLINA		(803) 772-6760
TEXAS,		
Austin		(512) 346-7830
Dallas		(214) 934-9099
Houston		(713) 785-9001

International

BELGIUM, Bruxelles	TEL	(02) 771-91-42
	FAX	(02) 762-37-12
	TLX	846-61028
FRANCE, Paris	TEL	(1) 49-75-10-10
	FAX	(1) 49-75-10-13
	TLX	263282F
WEST GERMANY,		
Hannover area	TEL	(0511) 736085
	FAX	(0511) 721254
	TLX	922850
München	TEL	(089) 4114-0
	FAX	(089) 406490
	TLX	523883
Stuttgart	TEL	(0711) 62 33 77
	FAX	(0711) 625187
	TLX	721882
HONG KONG,	TEL	852-5-8654525
Wanchai	FAX	852-5-8654335
	TLX	67955AMDAPHX
ITALY, Milan	TEL	(02) 3390541
		(02) 3533241
	FAX	(02) 3498000
	TLX	843-315286
JAPAN,		
Kanagawa	TEL	462-47-2911
	FAX	462-47-1729
Tokyo	TEL	(03) 345-8241
	FAX	(03) 342-5196
	TLX	J24064AMDTKOJ
Osaka	TEL	06-243-3250
	FAX	06-243-3253

International (Continued)

KOREA, Seoul	TEL	822-784-0030
	FAX	822-784-8014
LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
	TLX	5109554261 AMDFTL
NORWAY, Hovik	TEL	(03) 010156
	FAX	(02) 591959
	TLX	79079HBCN
SINGAPORE	TEL	65-3481188
	FAX	65-3480161
	TLX	55650 AMDMMI
SWEDEN,		
Stockholm	TEL	(08) 733 03 50
(Sundbyberg)	FAX	(08) 733 22 85
	TLX	11602
TAIWAN	TEL	886-2-7213393
	FAX	886-2-7723422
	TLX	886-2-7122066
UNITED KINGDOM,		
Manchester area	TEL	(0925) 828008
(Warrington)	FAX	(0925) 827693
	TLX	851-628524
London area	TEL	(0483) 740440
(Woking)	FAX	(0483) 756196
	TLX	851-859103

North American Representatives

CANADA

Burnaby, B.C.
DAVETEK MARKETING (604) 430-3680

Calgary, Alberta
DAVETEK MARKETING (403) 291-4984

Kanata, Ontario
VITEL ELECTRONICS (613) 592-0060

Mississauga, Ontario
VITEL ELECTRONICS (416) 676-9720

Lachine, Quebec
VITEL ELECTRONICS (514) 636-5951

IDAHO
INTERMOUNTAIN TECH MKTG, INC (208) 888-6071

ILLINOIS
HEARTLAND TECH MKTG, INC (312) 577-9222

INDIANA
Huntington - ELECTRONIC MARKETING CONSULTANTS, INC (317) 921-3450
Indianapolis - ELECTRONIC MARKETING CONSULTANTS, INC (317) 921-3450

IOWA
LORENZ SALES (319) 377-4666

KANSAS
Merriam – LORENZ SALES (913) 384-6556
Wichita – LORENZ SALES (316) 721-0500

KENTUCKY
ELECTRONIC MARKETING CONSULTANTS, INC (317) 921-3452

MICHIGAN
Birmingham - MIKE RAICK ASSOCIATES (313) 644-5040
Holland – COM-TEK SALES, INC (616) 399-7273
Novi – COM-TEK SALES, INC (313) 344-1409

MISSOURI
LORENZ SALES (314) 997-4558

NEBRASKA
LORENZ SALES (402) 475-4660

NEW MEXICO
THORSON DESERT STATES (505) 293-8555

NEW YORK
East Syracuse – NYCOM, INC (315) 437-8343
Woodbury – COMPONENT CONSULTANTS, INC (516) 364-8020

OHIO
Centerville – DOLFUSS ROOT & CO (513) 433-6776
Columbus – DOLFUSS ROOT & CO (614) 885-4844
Strongsville – DOLFUSS ROOT & CO (216) 238-0300

PENNSYLVANIA
DOLFUSS ROOT & CO (412) 221-4420

PUERTO RICO
COMP REP ASSOC, INC (809) 746-6550

UTAH, R^2 MARKETING (801) 595-0631

WASHINGTON
ELECTRA TECHNICAL SALES (206) 821-7442

WISCONSIN
HEARTLAND TECH MKTG, INC (414) 792-0920

Advanced Micro Devices reserves the right to make changes in its product without notice in order to improve design or performance characteristics. The performance characteristics listed in this document are guaranteed by specific tests, guard banding, design and other practices common to the industry. For specific testing details, contact your local AMD sales representative. The company assumes no responsibility for the use of any circuits described herein.

Advanced Micro Devices, Inc. 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA
Tel: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450
APPLICATIONS HOTLINE TOLL FREE: (800) 222-9323 • (408) 749-5703

© 1989 Advanced Micro Devices, Inc.
8/9/89
Printed in USA

ADVANCED MICRO DEVICES, INC.
901 Thompson Place
P.O. Box 3453
Sunnyvale,
California 94088-3453
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306
TOLL-FREE
(800) 538-8450

APPLICATIONS HOTLINE
(800) 222-9323
(408) 749-5703

Printed in USA
11339A