CS4041 CHIPSet

- ■ Local Bus
  - VESA Local Bus 2.0 Compatible
  - Full L1 Write Back Cache Support
  - Up to 3 LDEV#s and 3 sets of LREQ# / LGNT# pairs provided directly
  - Read and write bursting from VL Masters supported
- ■ DRAM Controller
  - 8 banks of DRAMs supported (4 double banks SIMMs, etc.)
  - Page mode and page interleave
  - 256K, 1M, 4M, and 16M deep DRAMs supported
  - Direct Drive RAS
  - Direct drive CAS, DWE, and MA for up to 36 DRAM chips
  - Hidden refresh with staggered RAS
  - SMM memory support
  - Variety of timing modes for system optimization
- ■ Cache Controller
  - Direct mapped, external tag, internal comparator
  - 16 byte line size
  - 64K, 128K, 256K, 512K or 1M size
  - Write back or Write through
  - Single bank or dual bank (word interleaved) cache
  - Multiple timing modes supported for cost performance tradeoff
- ■ Power Management
  - SMI support
  - Many power management features can be utilized without SMI
  - Internal Clock switching and stopping
  - Intel, AMD, and Cyrix support
  - Event monitoring
  - I/O restart capability
- ■ Integrated Local Bus IDE
  - Requires only 3 TTL
  - Support for 8 drives (4 connectors)
  - Data port accesses accelerated via local bus accesses
  - Timing modes selectable for each drive
- ■ Keyboard Controller
  - Integrated state machine based keyboard controller
  - Mouse port included
  - Keylock input provided on a multifunction pin

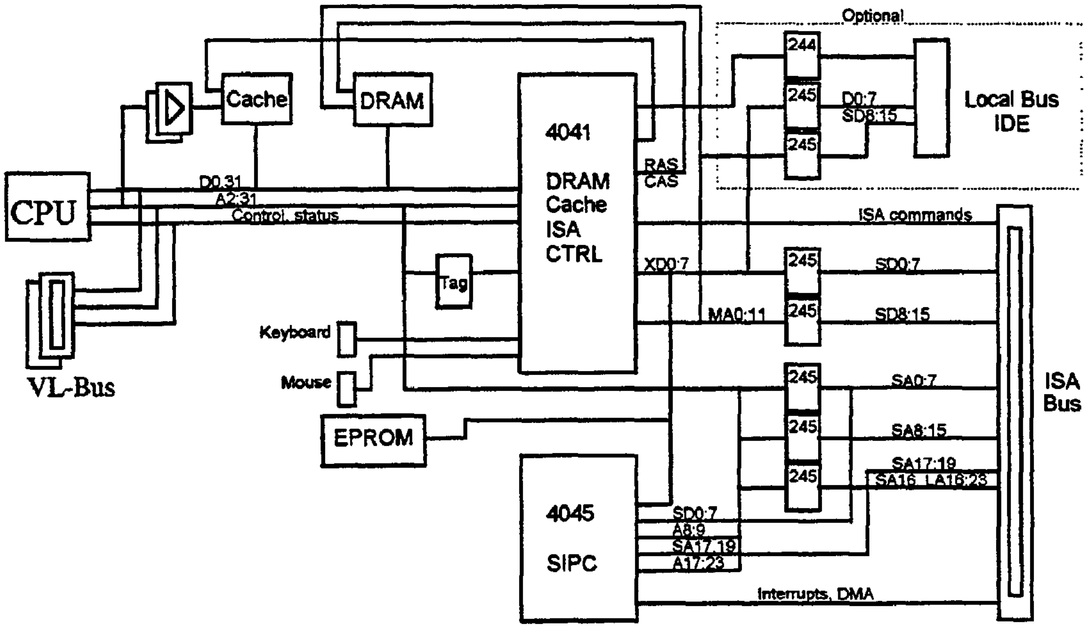


Figure 1: System Block Diagram

Table of Contents

List of Figures

List of Tables

1. Introduction

The CS4041 is the first product in the GreenCHIPS CHIPSet product portfolio of Chips and Technologies, Inc. It provides all of the system logic for implementing a high performance, Energy Star compliant 486 PC/AT design, while maintaining an extremely competitive cost structure. The powerful feature set includes the CHIPS "standard" system blocks and offers a new level of system integration while addressing the ever evolving requirements that the market place demands. It is 100% PC/AT compatible and directly supports the 486DX, 486DX2, 486DX4, 486SX and 486 derivatives that support the CPU write back cache architecture.

The high performance CHIPSet consists of the F84041 Systems Controller and F84045 GreenCHIPS IPC. The F84041 System Controller is packaged in a 208-pin PQFP and integrates the major system logic functions. Included in the F84041 is the CHIPS patented Page Interleave DRAM controller, high performance cache controller, VL local bus controller, ISA bus controller, power management module, a local bus IDE controller and fully compatible 8042 keyboard controller with PS/2 mouse support. The companion F84045 is packaged in a 100 pin PQFP and contains the industry standard Integrated Peripheral Controller (IPC) which includes the DMA controllers, timers, interrupt controllers and real time clock.

The enhanced feature set of GreenCHIPS DRAM and cache controllers are perfect for today's High Performance PC/AT designs. The page interleave DRAM controller offers high performance as well as extreme flexibility in supporting 486 memory subsystems. The DRAM controller supports up to eight banks of memory that can be configured with 256K, 1M, 4M or 16M memory devices. Page interleaving, timing modes, memory mix options, direct drive support and block by block parity support can be tuned to meet the most optimum requirements for the system design. In addition, the high performance secondary cache controller provides options that can be optimized for performance, cost or both. The direct mapped cache architecture employs internal comparators with external TAG and data SRAM that can operate in a write-through or write-back mode. Cache sizes from 64K to 1M are supported with flexible single bank or dual bank support that allow flexible timing mode selection based on CPU speed and SRAM speed.

The "Green" in GreenCHIPS comes from the Power management support integrated in the CHIPSet. The CS4041 provides the perfect level of power management support for Energy Star compliant desktops. Included in the power management section is direct support for SMM operation and clock switching for the popular 486 derivatives. Two event timers, programmable I/O pins, I/O restart and programmable event detection provide a wide range of options for power management selection and customization.

The CS4041 provides new levels of integration in system logic CHIPSets by providing a local bus IDE interface and keyboard controller. The robust local bus IDE interface is decoupled from the AT state machine and does not use a VL local bus load. The interface is versatile enough to support up to eight IDE drives allowing each drive to have unique command settings. The result is the best performance for each drive type allowing significant performance gains over the standard ISA interface. This is accomplished without any compromise to the standard VL local bus.

1.1. CPUs Supported

- Intel 486 CPUs
- AMD 486 CPUs
- Cyrix 486 CPUs
- IBM 486 CPUs
- L1 (CPU) write back cache fully supported
- SMI support (both Intel and Cyrix)
- Clock Frequencies:
  25MHz, 33MHz, 40MHz, 50MHz

1.2. External Chips

Qty	TTL Parts	Used For
	Basic System	
3	LS245	A <-> SA & LA
2	LS245	XD0:7 <-> SD0:7 & MA2:9 <-> SD8:15
1	F244	Clock buffer
1	F00	Miscellaneous
	Add for Cache:	
2	F244	Cache address
1	F08	BE# and W/R# combining
	Add for IDE:	
2	LS245	Data bus buffers
1	LS244	Control Signal Buffers
	DRAM Buffers	
0		For 2 banks
4	F244	For full complement of DRAMs. Buffering based on loading.

2. Pinouts

The CS4041 CHIPSet is comprised of two chips, the 84041 and the 84045. Following the 84041 and 84045 pinout diagrams are the pin descriptions.

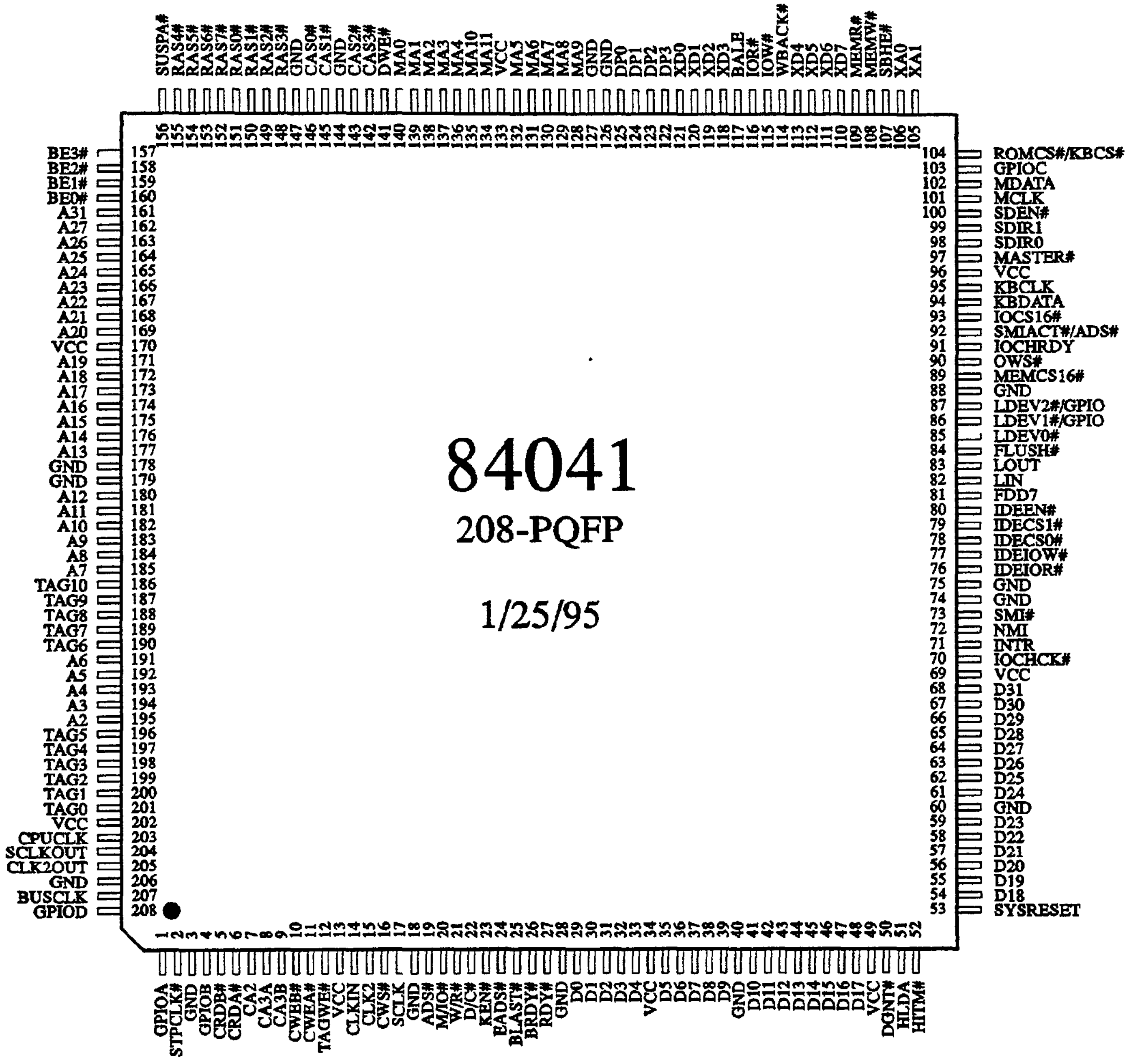


Figure 2.1: 84041 Pinout

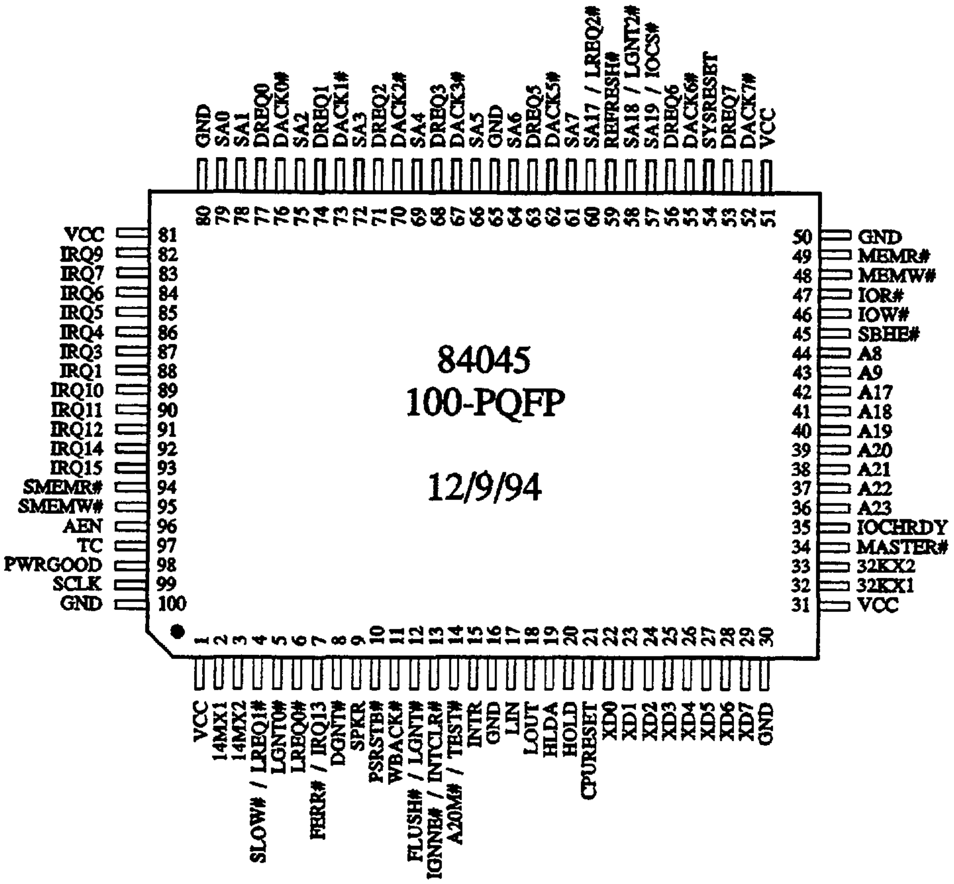


Figure 2.2: 84045 Pinout

2.1. Pin Descriptions

2.2. 84041 Pin Overview

The 84041 table below lists the pins by signal names. The detailed pin functional descriptions for the 84041 follow the pin list.

Table 2.1: 84041 Pin List

Signal	Qty	In/out	Driver	I_{OL} (mA)	I_{OH} (mA)	Ext. Load C_L (pF)	External Pull-up (Ω)	Comments
CLKIN	1	IN	-	-	-	-	-	From oscillator
CLK2OUT	1	OUT	TS	8	-2	30	-	4041 CLK2
SCLKOUT	1	OUT	TS	8	-2	85	-	4041, 4045, VL (usually buffered)
CPUCLK	1	OUT	TS	8	-2	50	-	CPU (usually buffered
STPCLK#	1	OUT	TS	4	-1	30	-	CPU
SUSPA#	1	IN	-	-	-	-	-	Cyrix CPU
CLK2	1	IN	-	-	-	-	-	Oscillator circuit
SCLK	1	IN	-	-	-	-	-	Oscillator circuit
CWS#	1	IN	-	-	-	-	-	Unbuffered clock
BUSCLK	1	OUT	TS	4	-1	50	-	Buffer
SYSRESET	1	IN	-	-	-	-	-	IPC
HLDA	1	IN	-	-	-	-	-	CPU
DGNT#	1	IN	-	-	-	-	-	IPC
MASTER#	1	IN	-	-	-	-	300	ISA Bus
SMIACT# / SMIADS#	1	IN	-	-	-	-	10K	CPU
ADS#	1	I/O	TS	4	-1	75	10K	CPU, loc. bus dev.
W / R#	1	I/O	TS	4	-1	75	-	CPU, loc. bus dev.
D / C#	1	I/O	TS	4	-1	75	-	CPU, loc. bus dev.
M / IO#	1	I/O	TS	4	-1	75	-	CPU, loc. bus dev.
RDY#	1	I/O	TS	4	-1	75	10K	CPU, loc. bus dev.
BRDY#	1	I/O	TS	4	-1	75	10K	CPU, loc. bus dev.
KEN#	1	OUT	TS	4	-1	30	-	CPU
FLUSH#	1	OUT	TS	4	-1	30	-	CPU
BLAST#	1	I/O	TS	4	-1	75	-	CPU, loc. bus dev.
EADS#	1	I/O	TS	4	-1	75	10K	CPU, loc. bus dev.
HITM#	1	IN	-	-	-	-	10K	CPU
WBACK#	1	OUT	TS	4	-1	100	-	To 4045 & VL slots
BALE	1	OUT	TS	24	-4	240	-	ISA Bus
MEMR#	1	I/O	TS	24	-3	240	10K	ISA Bus
MEMW#	1	I/O	TS	24	-3	240	10K	ISA Bus
IOR#	1	I/O	TS	24	-3	240	10K	ISA Bus
IOW#	1	I/O	TS	24	-3	240	10K	ISA Bus
IOCHRDY	1	I/O	OC	24	-	240	1K	ISA Bus
0WS#	1	IN	-	-	-	-	300	ISA Bus
MEMCS16#	1	I/O	OC	24	-	240	300	ISA Bus
IOCS16#	1	IN	-	-	-	-	300	ISA Bus
BE0-3#	4	I/O	TS	4	-1	75	-	CPU, loc. bus dev.
A2-9	8	IN	-	-	-	-	-	CPU, loc. bus dev., AT buffers
A10:16	7	I/O	TS	4	-1	75	-	CPU, loc. bus dev., AT buffers
A17:23	7	IN	-	-	-	-	-	CPU, loc. bus dev., AT buffers
A24:27, A31	5	I/O	TS	4	-1	75	-	CPU, loc. bus dev., AT buffers
SBHE#	1	I/O	TS	24	-4	240	-	ISA Bus
XA0-1	2	I/O	TS	4	-1	50	-	F245
ROMCS#	1	OUT	TS	4	-1	50	-	ROM chip, 8042CS
LDEV0:2#	3	IN	-	-	-	-	10K	From VL-Bus slots
CA2	1	OUT	TS	8	-2	75	-	8 cache RAMs
CA3A,B	2	OUT	TS	4	-1	50	-	4 cache RAMs
CRDA,B	2	OUT	TS	4	-1	50	-	4 cache RAMs
CWEA#,B	2	OUT	TS	4	-1	50	-	4 cache RAMs
TAGWE#	1	OUT	TS	4	-1	50	-	3 tag RAMs
TAG0:10	11	I/O	TS	4	-1	30	-	1 tag RAM

Table 2.1: 84041 Pin List (continued)

Signal	Qty	In/out	driver	I_{OL} (mA)	I_{OH} (mA)	Ext. Load C_L (pF)	External Pull-up (Ω)	Comments
RAS0-7#	8	OUT	TS	16	-3	120	-	12 DRAMs
CAS0-3#	4	OUT	TS	16	-3	65	-	6 DRAMs
DWE#	1	OUT	TS	16	-3	240	-	24 DRAMs
MA0-11	12	I/O	TS	16	-3	200	-	24 DRAMs
D0:31	32	I/O	TS	4	-1	75	-	CPU, DRAMs, local bus
DP0:3	4	I/O	TS	4	-1	65	-	CPU, DRAMs
XD0:7	8	I/O	TS	4	-1	65	-	8042, ROM, LS245, 4045
SDIR0:1	2	OUT	TS	4	-1	40	-	F245
SDEN#	1	OUT	TS	4	-1	40	-	F245, inverter
IOCHCK#	1	IN	-				4.7K	ISA Bus
NMI	1	OUT	TS	4	-1	30	-	CPU
SMI#	1	I/O	TS	4	-1	30	-	CPU
INTR	1	IN	-	-	-	-	-	4045
IDEIOR#	1	OUT	TS	4	-1	40	-	Buffer
IOEIOW#	1	OUT	TS	4	-1	40	-	Buffer
IDECS0,1#	2	OUT	TS	4	-1	40	-	Buffer
IDEEN#	1	OUT	TS	4	-1	40	-	Buffer enables
FDD7	1	IN	-	-	-	-	-	SD7
LIN	1	IN	-	-	-	-	-	IPC
LOUT	1	OUT	TS	4	-1	30	-	IPC
KBCLK, KBDATA	2	I/O	OC	4	-	240	4.7K	Keyboard connector, etc.
MCLK, MDATA	2	I/O	OC	4	-	240	4.7K	Mouse connector
GPIOA	1	IN	-	-	-	-	-	
GPIOB:D	3	I/O	TS	4	-1	50	-	
VCC	8							
GND	15							
Total	208							

Driver types: TS = tri-state, OC = open collector.

2.3. 84045 Pin Overview

The 84045 table below lists the pins by signal names. The detailed pin functional descriptions for the 84045 follows the pin list.

Table 2.2: 84045 Pin List

Signal	Qty	In/out	driver	I_{OL} (mA)	I_{OH} (mA)	Ext. Load C_L (pF)	External Pull-up (Ω)	Comments
14MX1	1	IN	-	-	-	-	-	14.31818MHz crystal input
14MX2	1	OUT	TP	2	-1	50	-	14.31818MHz crystal output
SCLK	1	IN	-	-	-	-	-	Oscillator circuit
PWRGOOD	1	IN	-	-	-	-	-	Power supply or PUC circuit
CPURESET	1	OUT	TP	4	-1	50	-	CPU(s)
SYSRESET	1	OUT	TP	8	-2	50	-	Coproc, 4025, buffer to ISA bus
HOLD	1	OUT	TP	4	-1	30	-	CPU(s)
HLDA	1	IN	-	-	-	-	-	CPU
DGNT#	1	OUT	TP	4	-1	40	-	4041, one buffer direction
WBACK#	1	IN	-	-	-	-	-	4041
LREQ0#	1	IN	-	-	-	-	10K	Local Master. Pull up if not used
LGNT0#	1	OUT	TP	4	-1	50	-	Local Master
MASTER#	1	IN	-	-	-	-	300	ISA Bus
REFRESH#	1	I/O	OC	24	-	240	300	ISA Bus
AEN	1	OUT	TP	24	-3	240	-	ISA Bus
TC	1	OUT	TP	4	-1	240	-	ISA Bus
DREQ0:3, 5:7	7	IN	-	-	-	-	10K	ISA Bus
DACK0:3, 5:7#	7	OUT	TP	4	-1	240	-	ISA Bus
SLOW# / LREQ1#	1	IN	-	-	-	-	10K	Turbo Button
FLUSH# / LGNT1#	1	OUT	TS	4	-1	30	-	CPU
MEMR#, MEMW#	2	OUT	TS	24	-3	240	10K	ISA Bus, 4041. Driven for DMA cycles only.
SMEMR#, SMEMW#	2	OUT	TP	24	-3	240	-	ISA Bus, 4041. Always driven.
IOR#, IOW#	2	I/O	TS	24	-3	240	10K	ISA Bus, 4041. Driven for DMA cycles only.
IOCHRDY	1	I/O	OC	24	-	-	1K	ISA Bus, 4041
SBHE#	1	OUT	TS	24	-3	240	-	ISA Bus, 4041
SA0:7	8	I/O	TS	24	-3	240	-	ISA Bus, F245
A8:9	2	I/O	TS	8	-2	65	-	CPU, 4041
SA17 / LREQ2#	1	I/O	TS	24	-3	240	-	ISA Bus / VL Master
SA18 / LGNT2#	1	OUT	TS	24	-3	240	-	ISA Bus / VL Master
SA19 / IOCS#	1	I/O	TS	24	-3	240	-	ISA Bus / I/O decode
A17:19	3	I/O	TS	8	-2	65	-	CPU, 4041. Driven for DMA, in to gen SA.
A20:23	4	I/O	TS	8	-2	65	-	CPU, 4041. Driven for DMA.
A20M# / TEST#	1	I/O	TS	4	-1	30	10K	CPU
XD0:7	8	I/O	TS	4	-1	65	-	4041, LS244, 8042, ROM
IRQ1, 3:7, 9:11,14:15	11	IN	-	-	-	-	-	ISA Bus (IRQ1 from 8042)
IRQ12	1	I/O	TS	4	-1	240	-	ISA Bus & internal mouse option
INTR	1	OUT	TP	4	-1	30	-	CPU
FERR# / IRQ13	1	IN	-	-	-	-	-	CPU or coprocessor logic
IGNNE#/INTCLR#/RTCIRQ#	1	OUT	TP	4	-1	30	-	CPU or coprocessor logic
SPKR	1	OUT	TP	4	-1	30	-	Speaker buffer
32KX1	1	IN	-	-	-	-	-	Crystal Circuit
32KX2	1	OUT	TP	*	*	*	-	Crystal Circuit
PSRSTB	1	IN	-	-	-	-	RC	RC circuit
LIN	1	IN	-	-	-	-	-	4041
LOUT	1	OUT	TP	4	-1	30	-	4041
VCC	4							
GND	6							
Total	100							

* Analog pin. Recommended external crystal circuit should be used (2 x 20 pF, 10K series, 10MΩ biasing).
TP = Totem-Pole, TS = Tri-State, OC = Open Collector.

2.4. 84041 Pin Descriptions

CLOCKs and RESET

CLKIN	14	IN	Input from Oscillator. Either 1x or 2x as determined by NMI at powerup (high for 1x, low for 2x). Used to create CLK2OUT, SCLKOUT and (optionally) BUSCLK. Also used as a time base for the power management timers.
CLK2OUT	205	OUT	2x system clock (when CLKIN is 2x). In full speed mode CLK2OUT is a buffered version of CLKIN. In slow mode it is the output of the clock divider. When CLKIN is 1x CLK2OUT will also be 1x, and will be at the same frequency and phase as SCLKOUT. In either mode it has a very low skew with respect to SCLKOUT and CPUCLK. It is externally fed back to CLK2 of the 4041, and any other logic requiring a 2x clock.
SCLKOUT	204	OUT	The 1x system clock for everything except the CPU. This output buffered and fed back to SCLK of the 4041, goes to the 4045, and the VL-Bus. The unbuffered SCLKOUT is used as CWS#.
SUSPA#	156	IN	Suspend Acknowledge from the Cyrix CPU. May be left floating if not used. Required only for Cyrix CPUs which contain a PLL. This pin is enabled with Configuration Register 38h bit 2.
CPUCLK	203	OUT	1x CPU clock. This output is the same as SCLKOUT except that it may be stopped by the power management hardware.
STPCLK#	2	OUT	Stop Clock signal to the CPU. Used for S series CPUs to stop the clock between the CPU PLL and the CPU core. The 4041 will optionally drive this pin low before changing clock frequencies.
CLK2	15	IN	2x clock input. Used for the DRAM state machine. It is also used as the source for the ISA bus clock divider.
SCLK	17	IN	1x clock input.
CWS#	16	IN	Cache Write strobe. This is an advanced 1x clock used for the cache write strobe in order to meet the data hold time of the SRAMs. It is normally advanced by 3-5nS from SCLK.
BUSCLK	207	OUT	ISA bus clock. Generated by dividing the CLKIN pin down by a variety of factors. BUSCLK should be driven onto the ISA bus through a non-inverting buffer. (BALE is generated during the low phase of BUSCLK).
SYSRESET	53	IN	System reset from the 4045.

Arbitration

HLDA	51	IN	CPU Hold Acknowledge. When low, indicates that the CPU has control of the local bus. When high, either a local master, the DMA controller, or an ISA master has the bus. The 4041 chip generates parity for DRAM write cycles when this pin is high. Cache line fills are only done when HLDA is low. The DRAM controller may switch timing modes based on HLDA to provide relaxed timing for alternate masters.
DGNT#	50	IN	DMA controller grant. When low, indicates that either the DMA controller or an ISA bus master has control of the bus. The 4041 becomes an ISA slave and floats the ISA bus commands when this signal is low.

MASTER#	97	IN	ISA bus master signal. Indicates that an ISA master has the bus. The 4041 uses this to determine the difference between DMA and ISA master cycles. It is used in determining the timing for IOCHRDY generation.

CPU & Local Bus control signals

SMIACT# / SMIADS#	92	IN	Indicates SMM memory accesses. The function depends on the CPU type. For Intel SMM it is an SMIACT# status signal. For Cyrix SMM it is the ADS# for SMM cycles.
ADS#	19	I/O	Address Strobe. Input for CPU and local master cycles, output for DMA and ISA master cycles.
W / R#	21	I/O	Write/Read status signal. Input for CPU and local master cycles, output for DMA and ISA master cycles.
D / C#	22	I/O	Data/Code status signal. Input for CPU and local master cycles, output for DMA and ISA master cycles.
M / IO#	20	I/O	Memory/IO status signal. Input for CPU and local master cycles, output for DMA and ISA master cycles.
RDY#	27	I/O	Non Burst ready. Output when 4041 is a slave. Input from a local bus slave or external cache controller.
BRDY#	26	I/O	Burst Ready. Output when 4041 is a slave. Input from a local bus slave or external cache controller.
KEN#	23	OUT	Cache Enable to the CPU. Always driven. Only local DRAM is cached in the CPU. Certain areas may be marked non-cacheable.
FLUSH#	84	OUT	Flush L1 cache. May be used when entering SMM.
BLAST#	25	I/O	Burst Last. Driven (low) for DMA and ISA master cycles.
EADS#	24	I/O	External Address Strobe. Used to snoop and invalidate the 486 cache on DMA and ISA master memory cycles. Floated when a local master has the bus. Optionally driven active for writes to write protected memory.
HITM#	52	IN	Hit Modified. Input from the CPU indicating that the result of the snoop is a dirty cache line, i.e., the CPU cache contains the only valid copy of data that an alternate master is attempting to read. See also WBACK# below.
WBACK#	114	OUT	Writeback. Output to the 4045 and VL bus slots based on the HITM# input, indicating that the CPU needs to perform an L1 cache writeback operation before an alternate master receives the data that the master is attempting to read. The 4041 determines when to allow the writeback to occur in relation to other system activity. The bus cycle from an alternate master will be aborted to allow the CPU to write back the data. When the 4041 has control of the local bus (ISA masters or DMA) it will back off the bus while WBACK# is low. As required by VL bus protocol, a local bus master (LBM) must be capable of aborting a cycle (without RDY# or BRDY#), then restarting the cycle again after the writeback operation is completed. In response to WBACK#, the 4045 drops HOLD long enough to give control the CPU, then re-asserts HOLD and gives control back to the alternate master after the CPU has finished the writeback operation and re-asserted HLDA.

ISA Bus

BALE	117	OUT	Buffered Address Latch Enable. Direct drive of the ISA bus.
MEMR#	109	I/O	Memory Read Strobe. Direct drive of the ISA bus. Output when HLDA or LGNT# are low. Input when they are both high.
MEMW#	108	I/O	Memory Write Strobe. Direct drive of the ISA bus. Output when HLDA or LGNT# are low. Input when they are both high.
IOR#	116	I/O	I/O Read Strobe. Direct drive of the ISA bus. Output when HLDA or LGNT# are low. Input when they are both high.
IOW#	115	I/O	I/O Write Strobe. Direct drive of the ISA bus. Output when HLDA or LGNT# are low. Input when they are both high.
IOCHRDY	91	I/O	ISA bus ready. Output when an ISA slave (DMA and ISA master accesses to local DRAM or local bus slaves). Open collector. Input for CPU or local master accesses to the ISA bus.
0WS#	90	IN	ISA bus Zero wait state signal. An ISA bus slave will drive this signal low when a memory command falls to force a 0 wait state cycle. It may also be used to force a 2 wait state cycle for 8 bit memory or I/O.
MEMCS16#	89	I/O	Output when an ISA slave (DMA and ISA master accesses to local DRAM or local bus slaves). Open collector. Input for CPU or local master accesses to the ISA bus.
IOCS16#	93	IN	Input for CPU or local master accesses to the ISA bus.

Address

BE0:3#	160, 159, 158, 157		
		I/O	Byte enables. Input for CPU or local master accesses to the ISA bus. Output for DMA or ISA master cycles. Generated from XA0:1 and SBHE#.
SBHE#	107	I/O	ISA Bus BHE#. Output for CPU or local master accesses to the ISA bus. Input for DMA or ISA master cycles.
XA0:1	106, 105I/O		Output for CPU or local master accesses to the ISA bus. Input for DMA or ISA master cycles.
A2:9	195, 194, 193, 192, 191, 185, 184, 183		
		IN	Local Bus Address bus. Always inputs.
A10:16	182, 181, 180, 177, 176, 175, 174		
		I/O	Local Bus Address bus. Output for DMA cycles. A10:16 of the DMA address is sent on XD0:6 from the 4045 and latched in the 4041.
A17:23	173, 172, 171, 169, 168, 167, 166		
		IN	Local Bus Address bus. Always inputs.

A24:27, A31 165, 164, 163, 162, 161

I/O Local bus address. Driven low for DMA and ISA master cycles.

ROMCS# / KBCS#

104 OUT Logical OR of the ROM chip select and 8042 chip select. The 8042 chip select is active for I/O ports 60 and 64. The ROM chip select is programmable.

LDEV0:2# 85,86,87IN Local Device. A local bus slave or cache controller drives these signals low to indicate that it will handle the cycle. This signal is sampled either at the end of the first or second T2. LDEV1# and LDEV2# may be redefined as other inputs. They are disabled at power up. LDEV0# is enabled at power up.

Cache Controller

CA2 7 OUT Cache address bit 2 for a single bank cache. Upper most data SRAM address bit (A15,16,17,18,or 19) for a double bank cache.

CA3A, CA3B 8, 9 OUT Cache address bit 3 for each bank of a double bank cache. CA3A used for a single bank.

CRDA#, CRDB# 6, 5 OUT Cache Read strobe for each bank of a double bank cache. CRDA# used for a single bank.

#CWEA, CWEB# 11, 10 OUT Cache Write strobe for each bank of a double bank cache. CRDA# used for a single bank. Generated from CWS#.

TAGWE# 12 OUT Tag Write enable. Driven low during L2 read miss cycles and when changing the dirty bit from clean to dirty.

TAG0:10 201, 200, 199, 198, 197, 196, 190, 189, 188, 187, 186

I/O Tag RAM bits. TAG0 is the dirty bit. 8, 9, and 11 bit tags are supported. TAG0:7 is always used for 8 bit tag, TAG0:8 for 9 bit tag. Unused bits must be pulled up.

DRAM Controller

RAS0:7# 151, 150, 149, 148, 155, 154, 153, 152

OUT RAS for each of 8 DRAM banks. Direct Drive.

CAS0:3# 146, 145, 143, 142

OUT CAS0:3 for each byte of DRAM Direct drive for up to 2 banks of DRAM.

DWE# 141 OUT DRAM write enable. Direct drive for 2 banks of DRAM. Also used to control the direction of external DRAM data buffers if used.

MA0:1 140, 139

MA2:9/XD8:15 138, 137, 136, 132, 131, 130, 129, 128

MA10:11 135, 134

I/O DRAM address/upper data bus. Direct drive for up to 2 banks of DRAM. During ISA cycles MA2:9 become XD8:15 respectively (the upper byte of ISA bus data). They are buffered with a 245 to generate SD8:15. MA0:1 and 10:11 are output only.

Data Bus

D0:7	29, 30, 31, 32, 33, 35, 36, 37		
D8:15	38, 39, 41, 42, 43, 44, 45, 46		
D16:23	47, 48, 54, 55, 56, 57, 58 ,59		
D24:31	61, 62, 63, 64, 65, 66, 67, 68		
		I/O	Local bus data.
DP0:3	125, 124, 123, 122		
		I/O	DRAM parity bits. Connected to DP0:3 of the 486 also. The 486 generates parity when it has the bus. For local masters, DMA cycles, and ISA masters, the 4041 generates parity. Parity is checked by the 4041 for all cycles.
XD0:7	121, 120, 119, 118, 113, 112, 111, 110		
		I/O	Intermediate data bus. Connected directly to the 4045, BIOS ROM, 8042, and, through an LS245, to the ISA bus. For XD8:15, see MA2:9.
SDIR0:1	98, 99	OUT	Direction control for the XD0:7 to SD0:7 (SDIR0) and the XD8:15 to SD8:15 (SDIR1) buffer. 0=SD to XD. 1= XD to SD.
SDEN#	100	OUT	SD bus enable. Connects to the enable of the XD to SD buffers. Goes high to disable the buffers during the local bus portion of DMA and ISA master cycles, allowing the MA bus to be used for the DRAMs.

Local IDE

IDEIOR#	76	OUT	IDE controller I/O read strobe. Driven by either the local bus IDE logic or the ISA bus IOR# logic, depending on the cycle.
IDEIOW#	77	OUT	IDE controller I/O write strobe. Driven by either the local bus IDE logic or the ISA bus IOR# logic, depending on the cycle.
IDECS0#	78	OUT	IDE controller Chip Select 0. Decodes addresses 1F0:1F7 (when the default IDE address is used).
IDECS1#	79	OUT	IDE controller Chip Select 1. Decodes addresses 3F6:3F7 (when the default IDE address is used).
IDEEN#	80	OUT	IDE Bus driver enable. Connected to the enable pins of the IDE data bus drivers.
FDD7	81	IN	Used for the floppy disk controller disk change bit for 3F7 reads. Normally this pin is connected directly to SD7. It may be connected directly to the DSKCHG# pin of the floppy connector instead.

Interrupts

IOCHCK#	70	IN	Parity error indicator from the ISA bus. Generates an NMI and sets the IOCHCK# flag.
NMI	72	I/O	NMI to the CPU. Generated for DRAM parity errors and when IOCHCK# has gone low. Each of these has enable bits plus a final mask. This pin is also sampled at the end of SYSRESET (falling edge) to determine whether the 4041 will operate in 1X clock mode (pin pulled up) or 2X mode (pin pulled down).
SMI#	73	OUT	System Management Interrupt. Output of the power management logic.
INTR	71	IN	CPU INTR pin from the 4045. Used to detect system events. Specifically used to restart the CPU clock when it has been stopped.

Control Link & Keyboard

LIN	82	IN	Control Link input from the 4045. Transfers the following information: Port B bit 5, Refresh Request, Refresh Complete, Address Strobe.
LOUT	83	OUT	Control Link output to the 4045. Transfers the following information: Interrupt Acknowledge cycle, CPU Reset Request, Refresh Acknowledge, and keyboard interrupt (from internal keyboard controller).
KBDATA/GATEA20	94	I/O	Keyboard data to internal keyboard controller. If the internal keyboard controller is disabled, this signal becomes the Gate A20 signal from the external 8042. The 4041 detects transitions on this pin and transmits them to the 4045 over the control link.
KBCLK / KBRESET#	95	I/O	Keyboard clock to the internal keyboard controller. If the internal keyboard controller is disabled, this signal becomes the CPU reset from the external 8042. When this signal goes low, the 4041 sends a code across the control link to inform the 4045 of this. The 4045 will perform the CPU reset (restart).
MDATA	102	I/O	Mouse data to internal 8042.
MCLK	101	I/O	Mouse clock to internal 8042.

Multifunction Pins

GPIOA	1	IN	Multifunction pin. Always an input.
GPIOB	4	OUT	Multifunction pin. Always an output.
GPIOC	103	OUT	Multifunction pin. Always an output.
GPIOD	208	OUT	Multifunction pin. Always an output.

VCC, GND

VCC	(8 pins)	13, 34, 49, 69, 96, 133, 170, 202
GND	(15 pins)	3, 18, 28, 40, 60, 74, 75, 88, 126, 127, 144, 147, 178, 179, 206

84045 Pin Descriptions

Clocks

14MX1	2	IN	14.31818MHz crystal input. Divided internally by 12 for the 8254 clock inputs. May also be used as a source for the DMA clock.
14MX2	3	OUT	14.31818MHz crystal output. This pin is buffered externally and used as the ISA bus OSC signal.
SCLK	99	IN	1x CPU clock input. Used for arbitration logic, reset generation, and selectively divided down to make the DMA clock.

Resets

PWRGOOD	98	IN	From the power supply or power on clear circuit.
CPURESET	21	OUT	RESET to the CPU only. Synchronized to SCLK. Connects to SRESET of an S series CPU.
SYSRESET	54	OUT	RESET to the rest of the system. Has the same timing as CPURESET, but is only active following PWRGOOD being low. Connects to RESET of an S series CPU if CPURESET is used for SRESET.

Arbitration

HOLD	20	OUT	HOLD to the CPU. Synchronous to SCLK.
HLDA	19	IN	HLDA from the CPU. The 4045 assumes it is synchronous to SCLK.
LREQ0#	6	IN	Bus request from the local masters. Arbitrated with the other two LREQN# signals (if used) and with the DMA controllers and CPU for control of the bus.
LGNT0#	5	OUT	Bus Grant to local bus masters. This signal goes low to give control to a local bus master. Two additional LREQ/LGNT pairs are provided on multifunction pins. Multiple sets may also be created externally with a PAL. The 4045 will optionally preempt a local master off the bus when an unmasked DMA request occurs.
DGNT#	8	I/O	DMA controller hold acknowledge (an output only). Indicates that the DMA controller or ISA master has control of the bus. Used for buffer steering and goes to the 4041. When PWRGOOD is low or SYSRESET is high this pin is an input to select the SA17:19 configuration. High = SA17:19. Low for alternate functions.
WBACK#	11	IN	This signal is used to take HOLD to the CPU low for 4 clocks regardless of the state of the arbitration. It is used to allow a CPU with a write back cache to perform its write back following a snoop of a DMA, ISA Master, or local bus master cycle. It also floats address lines A8:9, and A17:23 if a DMA cycle is in progress to allow the CPU to drive these lines.
MASTER#	34	IN	ISA masters pull this signal low after gaining control of the bus through a DMA channel's DREQ / DACK# signals. When this signal goes low the AEN output is taken low.
REFRESH#	59	I/O	ISA bus refresh signal. The 4045 drives this signal low during refresh cycles. During refresh cycles while an ISA master has control of the bus, the master drives it low.

AEN	96	OUT	DMA address enable. High for DMA and refresh cycles, low at all other times, which includes when the CPU, local masters and ISA masters have control of the bus. The main function of this signal is to disable all IO decodes in the system.
TC	97	OUT	DMA terminal count. This signal goes high during the final cycle of a DMA transfer. It is used mainly by the floppy disk controller, but may also be used by other DMA devices.
DREQ0:3 DREQ5:7	77, 74, 71, 68 63, 56, 53	IN	DMA requests. 0:3 are 8 bit channels. 5:7 are 16 bit channels. Any may be used for ISA masters, but 5:7 are preferred because there is less arbitration overhead.
DACK0:3# DACK5:7#	76, 73, 70, 67 62, 55, 52	OUT	DMA acknowledges.
SLOW# / LREQ1#	4	IN	Dual Function pin, determined by a configuration register. Turbo switch input. Low is slow, high is fast. The Performance Control registers must be set up before this signal has any effect. The purpose is to emulate the speed of an 8MHz AT for software (mostly games and copy protect schemes) which assume a certain execution speed. LREQ1#. Additional VL-Bus master request. Internally arbitrated with the other 1 or 2 LREQ#s.
FLUSH# / LGNT1#	12	OUT	Dual Function pin, determined by a configuration register. FLUSH#. This signal, when used, is connected to the CPU FLUSH# pin. It is used in conjunction with the performance control. The CPU cache is optionally flushed each time the CPU is put into HOLD for performance control to better control the speed of execution. LGNT1#. Additional VL-Bus master grant.

ISA Bus (4045)

MEMR#	49	I/O	Memory Read strobe. Connected directly to the ISA bus. Output during DMA cycles, input at all other times to generate SMEMR#.
MEMW#	48	I/O	Memory Write strobe. Connected directly to the ISA bus. Output during DMA cycles, input at all other times to generate SMEMR#.
SMEMR#	94	OUT	Memory Read strobe for the bottom 1Mbyte. Connected directly to the ISA bus. Output at all times. This signal is a function of A20:23 and MEMR#, and is low when all of those signals are low.
SMEMW#	95	OUT	Memory Write strobe for the bottom 1Mbyte. Connected directly to the ISA bus. Output at all times. This signal is a function of A20:23 and MEMW#, and is low when all of those signals are low.
IOR#	47	I/O	I/O Write strobe. Connected directly to the ISA bus. Output during DMA cycles, input at all other times, to access internal I/O.
IOW#	46	I/O	I/O Write strobe. Connected directly to the ISA bus. Output during DMA cycles, input at all other times, to access internal I/O.

IOCHRDY	35	I/O	ISA bus ready. Input during DMA cycles to add wait states to the command strobes. Output during accesses to the DMA controller registers to optionally add a wait state.

Address Bus (4045)

SBHE#	45	OUT	ISA bus byte high enable. Driven only during DMA cycles. For 8 bit DMA (channels 0:3) it is driven with the inverse of A0. For 16 bit DMA (channels 5:7) it is driven low.
SA0:7	79, 78, 75, 72, 69, 66, 64, 61	I/O	ISA bus address bits 0:7. Direct drive of the ISA bus. Outputs during DMA cycles and refresh cycles. Inputs at all other times.
A8:9	44, 43	I/O	Local bus address bits 8:9. Connected to the CPU local bus. Outputs during DMA cycles. Inputs at all other times.
SA17 / LREQ2#	60	I/O	Dual Function. ISA bus address bit SA17. Direct drive of the ISA bus. Output at all times except ISA master cycles, where it is floated. It is driven from A17. LREQ2#. Additional VL-Bus master request. Internally arbitrated with the other 1 or 2 LREQ#s.
SA18 / LGNT2#	58	I/O	Dual Function. ISA bus address bit SA18. Direct drive of the ISA bus. Output at all times except ISA master cycles, where it is floated. It is driven from A18. LGNT2#. Additional VL-Bus master grant.
SA19 / IOCS#	57	I/O	Dual Function. ISA bus address bit SA19. Direct drive of the ISA bus. Output at all times except ISA master cycles, where it is floated. It is driven from A19. IOCS#. Internal I/O chip select. When high the internal I/O is disabled. May be connected to a decode of A10:15, which may be provided by the 4041.
A17:19	42, 41, 40	I/O	Local bus address bits 17:19. Connected to the CPU local bus. Outputs during DMA cycles, inputs (to drive SA17:19) at all other times.
A20:23	39, 38, 37, 36	I/O	Local bus address bits 20:23. Connected to the CPU local bus. Outputs during DMA cycles, floated at all other times. Also used as inputs to generate SMEMR# or SMEMW#.
A20M# / TEST#	14	OUT	Connected to the CPU A20M# pin. This is the OR of the emulated keyboard GATEA20 and the fast GATEA20 (port 92 bit 1). After reset this pin is the TEST# input. If pulled low the 4045 will go into test mode. It becomes the A20M# output after a configuration bit has been set, at which time it begins driving the pin and ignoring it as an input. A 10K pull-up should be connected to this pin to prevent test mode from being entered and to keep A20M# high at power up.

Data Bus (4045)

XD0:7	22, 23, 24, 25, 26, 27, 28, 29		
		I/O	Data bus. Outputs for I/O reads of internal registers. Inputs for I/O writes of internal registers. Also outputs to pass A10:16 to the 4041 at the start of DMA cycles.

Interrupts

IRQ1	88		
IRQ3:7	87, 86, 85, 84, 83		
IRQ9:12	82, 89, 90, 91		
IRQ14:15	92, 93		
		IN	Interrupt inputs to the 8259s. Active high.
FERR# / IRQ13	7	IN	Dual function pin. Normally it is FERR# from the CPU and goes to the internal coprocessor error logic. It may optionally be IRQ13, if the coprocessor error logic is external (for Weitek support). IRQ13 is active high.
IGNNE# / INTCLR# / RTCIRQ#			
	13	OUT	Dual function pin. Normally it is IGNNE# to the CPU and comes from the internal coprocessor error logic. It may optionally be INTCLR# (active low write strobe for I/O ports F0 & F1), if the coprocessor error logic is external (for Weitek support). When PWRGOOD is low this pin provides the RTC alarm interrupt output. It is open collector at this time. When an RTC alarm occurs, this pin is driven low.
INTR	15	OUT	Interrupt request to the CPU.

Timer & RTC

SPKR	9	OUT	Speaker output. Connected to the speaker, externally buffered if needed for adequate speaker drive. When the speaker is idle (Port 61h bit 1 = '0'), this output remains continuously high.
32KX1/IRQ8#	32	IN	Real time clock 32KHz crystal input. When internal RTC is disabled, the function of this pin changes to become the IRQ8# input. A logic low level on IRQ8# causes INTR to be asserted (high), subject to the programmed INTC mode.
32KX2	33	OUT	Real time clock 32KHz crystal Output.
PSRSTB#	10	IN	Real Time clock Power Strobe. This signal being low indicates that the real time clock has lost power. It should be connected the 4045 VCC pins (which receive battery backed power) through a 100K resistor, and to GND through a 0.1uF cap.

Control Link, Etc. (4045)

LIN	17	IN	Control Link input from the 4041. Transfers the following information: Interrupt Acknowledge cycle, CPU Reset Request, Refresh Acknowledge, and keyboard interrupt (from internal keyboard controller).
LOUT	18	OUT	Control Link output to the 4041. Transfers the following information: Port B bit 5, Refresh Request, Refresh Complete, Address Strobe.

VCC, GND

VCC	1, 31, 51, 81
GND	15, 30, 50, 65, 80, 100

3. Registers

3.1. I/O Ports and Configuration Registers

Table 3.1: 84041 and 84045 I/O Port Summary

Ports	4041	4045	Description
00-0F		X	DMA controller #1 (8 bit DMA).
20-21		X	Interrupt Controller #1 (IRQ0-7).
22-23	X	X	Configuration register Address and Data Port.
26-27	X	X	SMM Configuration register Address and Data Port.
40-43		X	Timer Chip (8254).
60	X	X	Keyboard Data & Mouse port.
61	X	X	"Port B"
64	X		Keyboard Command/Status port.
70	X	X	Real Time Clock Address Port & NMI mask. (4041: bit 7, write only)
71		X	Real Time Clock Data Port.
80-8F		X	DMA Page Registers
92	X	X	Fast CPU reset & GATEA20.
A0-A1		X	Interrupt Controller #2 (IRQ8-15).
C0-DF		X	DMA Controller #2 (16 bit DMA).
F0-F1		X	NPU Error Reset.
170-177	X		IDE (Secondary)
1F0-1F7	X		IDE (Primary)
376-377	X		IDE (Secondary)
3F6-3F7	X		IDE (Primary)
570-577	X		IDE (Fourth)
5F0-5F7	X		IDE (Third)
776-777	X		IDE (Fourth)
7F6-7F7	X		IDE (Third)

Table 3.2: 84041 and 84045 Configuration Register Summary

Indexes	4041	4045	Description
01		X	IPC DMA controller wait states, clock.
05	X		ISA bus command delays
06	X		ISA bus wait states, address hold.
07	X		ISA bus clock selection
08		X	Performance Control
09		X	4045 Misc. Control
0A		X	DMA Clock selection.
0B		X	VL Arbitration and WBACK# control
0C		X	Port 92 and RTC feature control
0D-0F	-	-	(Reserved - for 4045)
10	X		DRAM Timing
11	X		DRAM Setup
12	X		DRAM Configuration, blocks 0 & 1
13	X		DRAM Configuration, blocks 2 & 3
14	X		DRAM block 0 Starting Address
15	X		DRAM block 1 Starting Address
16	X		DRAM block 2 Starting Address
17	X		DRAM block 3 Starting Address
18	X		Video shadow and local bus control.
19	X		DRAM shadow read enable.
1A	X		DRAM shadow write enable.
1B	X		ROMCS enable.
1C	X		SMM Shadow RAM enable.
1D	X		DRAM Block Parity Enable & Flash ROM shadow enable
1E	X		Secondary DRAM Timing register
1F			(Reserved) for Secondary Timing Select Register.
20	X		Cache Controller Mode
21	X		Cache Controller Configuration
22	X		Cache Testing Control
23	X		Data port for tag testing, bits 0:7
24	X		Data port for tag testing, bits 8:10
25-27			(Reserved)
28	X		I/O Decode #0 Address Low.
29	X		I/O Decode #0 Address High.
2A	X		I/O Decode #0 Size and Mask.
2B	X		I/O Decode #0 Configuration.
2C	X		I/O Decode #1 Address Low.
2D	X		I/O Decode #1 Address High.
2E	X		I/O Decode #1 Size and Mask.
2F	X		I/O Decode #1 Configuration.
30	X		Memory Decode #0 Address Low.
31	X		Memory Decode #0 Address High.
32	X		Memory Decode #0 Size and Destination.
33	X		Memory Decode #0 Attributes.
34	X		Memory Decode #1 Address Low.
35	X		Memory Decode #1 Address High.
36	X		Memory Decode #1 Size and Destination.
37	X		Memory Decode #1 Attributes.

Table 3.3: 84041 and 84045 Configuration Register Summary (continued)

Indexes	4041	4045	Description
38	X		CPU Modes
39	X		8042 Modes.
3A-3B			(Reserved).
3C	X		Multifunction Pin selection register A
3D	X		Multifunction Pin selection register B
3E	X		General Purpose Output pin data register
3F	X		General Purpose Input pin data register
40			(Reserved) for IDE.
41	X		IDE Enables and Control Functions
42	X		IDE Drive 0:3 Timing Modes
43	X		IDE Drive 4:7 Timing Modes
44	X		IDE Timing A Read and Write pulse Widths
45	X		IDE Timing A Command Recovery and Address Setup
46	X		IDE Timing B Read and Write pulse Widths
47	X		IDE Timing B Command Recovery and Address Setup
48:4F			(Reserved) for IDE.
80	X		EventA Selection 0. Interrupts.
81	X		EventA Selection 1. I/O and memory accesses
82	X		WakeA Event Selection 0. Interrupts
83	X		WakeA Event Selection 1. I/O and Memory Accesses
84	X		EventB Selection
85	X		WakeB Selection
86	X		Port Selection for Events
87	X		Interrupt Acknowledge 0 base.
88	X		TimerA Control Register
89	X		TimerA Count Register
8A	X		TimerB Control Register
8B	X		TimerB Count Register
8C	X		Time Base Selection
8D	X		Clock Switching modes.
8E	X		Software Commands and Status
8F	X		EXT0 and EXT1 Pin Mode
90	X		SMI Status Register 0.
91	X		SMI Status Register 1.
92	X		SMI Enable Register 0.
93	X		SMI Enable Register 1.
94	X		SMI Modes.
95	X		I/O Restart Selection
96	X		Port 70 shadow register

3.2. 84041 I/O Port Addresses

Note: "Default" register values refer to the power-on hardware defaults established automatically following hardware reset, before any alternate values have been written by the BIOS. "Typical" values refer to typical settings for normal system operation.

Addr	Bits	Description
22		**Configuration register Address Port.** Write only port which holds the address of the Chips and Technologies Index register to be accessed through I/O port 23. This register must be written before each access to port 23, even if the same index register is being accessed twice in a row.
23		**Configuration register data.** Accessing this port accesses the Configuration register pointed to by port 22. A second access to port 23 without writing port 22 in between will be ignored. Unless otherwise noted in the register descriptions, reserved or undefined index registers should not be written to, and reserved bits within a defined index register should be written as zero (or written with the same value previously read).
26		**SMM Configuration Register Address Port** Write only. The address written here is stored separately from port 22. This register is used as the config register index when port 27 is read or written.
27		**SMM Configuration Register Data Port** Accessing this port accesses the Configuration register pointed to by port 26. A second access to port 27 without writing to port 27 in between will be ignored. The "accessed" bit is separate for the port 26/27 and 22/23 windows.
60		**Keyboard Data port.** Used for keyboard GATEA20 and Fast Reset function.

(Continued on next page)

Addr Bits Description

61 "Port B"

This is an AT compatible port with miscellaneous information. Bits 0-3 are read/write. Bits 4-7 are read only. Only bits 2, 3, 6, & 7 are contained in the 4041. The remainder are in the 4045. On I/O reads, half of the bits come from each chip. Default = 20h.

0 (Timer 2 gate). This bit is in the 4045 and enables or disables the 1.19 MHz clock input to Timer 2. The output from Timer 2, in conjunction with bit 1 below, provides the signal for the speaker. If this bit is a 1, Timer 2 is enabled, and (if programmed to do so) will produce a square wave of the programmed frequency. When this bit is a 0, bit 5 below is forced to 1 and the speaker output signal will be high or low depending on bit 1 below.

1 (Speaker Data). This bit is in the 4045. This bit is ANDed with the output of timer 2 and inverted to produce the signal actually sent to the speaker. When the gate (bit 0 above) is low, this bit gives direct software control of the speaker. The speaker signal will be high or low when this bit is 0 or 1, respectively. When the speaker is idle, bits 0 and 1 normally will both be 0 and the speaker output signal will be high.

2 Enable Parity Check. 0 enables local DRAM parity checking. A 1 disables local DRAM parity checking and clears the local parity error flip-flop. This bit is inverted and sent to the active low preset of a flip-flop. The Q output is PCK# and is fed to the NMI logic. A parity error clocks the flip-flop to a 0. There is also an index register bit to block local DRAM parity errors. It prevents the flip-flop from being clocked. The flip-flop is an F74 on the AT, where the Preset has precedence on Q and Clear has precedence on Q#.

3 Enable IOCHCK. 0 enables the IOCHCK interrupt. A 1 disables IOCHCK and clears the IOCHCK flip-flop. This bit is inverted and sent to the active low clear of a flip-flop. IOCHCK# is sent to the active low Preset input. The Q# output is fed to the NMI logic. The Q output is sent to bit 6 of this register. The flip-flop is an ALS74 on the AT, where the Preset has precedence on Q and Clear has precedence on Q#.

4 (Refresh Detect). This bit is in the 4045. This read only bit toggles on each refresh. It should toggle whenever timer 1 produces a pulse (about every 15us). This should be done even if ISA refresh is disabled. Some software uses this as a time delay.

5 (Timer 2 output). This bit is in the 4045. Read only. This bit allows software to monitor the output of timer 2, which is ANDed with bit 1 of this register and inverted to produce the speaker signal. The speaker signal is low when bits 1 and 5 are both 1. If either bit is 0, the speaker output is high. By setting bits 1 and 0 to '01', software can use Timer 2 and bit 5 without generating any speaker output.

6 Channel Check latch. A 1 indicates that IOCHCK# has been activated. This bit is the Q output of the flip-flop mentioned in bit 3 of this register.

7 Parity Check latch. A 1 indicates that a local parity error has occurred. It is the Q# output of the flip-flop mentioned in bit 2 of this register.

Port	D7	D6	D5	D4	D3	D2	D1	D0
61	Parity err	CHCK	(Tmr 2)	(Ref Detect)	chck enable	parity enab	(spkr data)	(tmr2 gate)

64 Keyboard Command/Status port.

Used for keyboard GATEA20 and Fast Reset function.

70 Real Time Clock Address Port & NMI mask.

0-6 Write only shadow register. Read back through index 96.

7 NMI Mask. This bit is inverted and ANDed with the NMI sources (the OR of several sources). The result of the AND function is NMI to the CPU. This allows the CPU "Non-Maskable Interrupt" to be maskable externally.

Addr	Bits	Description
92		**Fast CPU reset.** Write only in the 4041. Read in the 4045
	0	Fast CPU reset. A 0 to 1 transition activates a CPU reset. Detected in the 4041 for redirection to SMI. Write only.
	7:1	(Not used in the 4041). Some of these bits are contained in the 4045.
1F0:1F7 **3F6:3F7**		**IDE Controller Primary Address.** When the fast IDE is enabled at this address, 1F0 (byte, word, or double word access) activates the internal IDE state machine. 1F0:1F7 generate IDECS0#. 3F6:3F7 generate IDECS1#. The data is redirected to the IDE buffers.
170:177 **376:377**		**IDE Controller Secondary Address.** When the fast IDE is enabled at this address, 170 (byte, word, or double word access) activates the internal IDE state machine. 170:177 generate IDECS0#. 376:377 generate IDECS1#. The data is redirected to the IDE buffers.
5F0-5F7 **7F6:7F7**		**IDE Controller Third Address.**
570-577 **776:777**		**IDE Controller Fourth Address.**

The following I/O ports are decoded for event detection, etc., but the actual I/O ports themselves reside externally to the 4041 (except for the IDE and keyboard, which may be contained in the 4041):

60 & 64		Keyboard Controller
170:177		Secondary IDE
1F0:1F7		Primary IDE
2E8:2EF	COM4	
278:27F		LPT (3)
2F8:2FF		COM2
3E8:3EF	COM3	
370:375		Secondary Floppy
376:377		Secondary IDE
378:37F		LPT (2)
3B0:3BB	VGA (monochrome section)	
3BC:3BE	LPT (1)	
3C0:3DF	VGA (extended and color sections)	
3F0:3F5		Floppy
3F6:3F7		Primary IDE
3F8:3FF		COM1

3.3. 84041 Index Configuration Registers

Note: Various suggestions for programming the Configuration Registers are included in the register descriptions. The following bits are affected by the CPU type; refer to the respective bit descriptions for specific programming suggestions:
Index 38h bits 0, 1, 2.
Index 8Dh bits 3 through 7.
Index 93h bit 7.
Index 94h bits 0, 1, 3, 4.

Index   Bits   Description

05   **ISA bus command delays** Default = 05. Typical setting = 05.
Two bits for each cycle type.
00 0 BCLK delay (command active at falling edge of ALE - default for 16 bit memory).
01 1 BCLK delay (default for all cycles except 16 bit memory).
10 2 BCLK delay
11 3 BCLK delay
1:0 I/O cycle command delay. Default is 1.
3:2 8 bit memory command delay Default is 1.
5:4 16 bit memory command delay Default is 0.
7:6 (Reserved). Write 0s.

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
05	ISA cmd dly	-	-	16b m dly	16b m dly0	8b m dly1	8b m dly0	IO dly1	IO dly0

06   **ISA bus Wait States, Address hold.** Default = 00. Typical setting = 24h.
1:0 (Reserved)
3:2 8 bit AT bus wait states
00 5 BUSCLK wait states (default)
01 4 BUSCLK wait states.
10 3 BUSCLK wait states.
11 2 BUSCLK wait states.
5:4 16 bit AT bus wait states
00 3 BUSCLK wait states (default)
01 2 BUSCLK wait states.
10 1 BUSCLK wait states.
11 0 BUSCLK wait states.
6 (Reserved).
7 AT bus address hold time (ready delayed by an extra clock after the AT command goes inactive).
0 no additional hold time (default)
1 additional hold time.

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
06	ISA wait state	add hold	-	16b wait1	16b wait0	8b wait1	8b wait0	-	-

Index Bits Description

07 ISA Bus Clock selection. Default = 00.

3:0 Clock select for ISA bus and internal ISA state machine. The source of the clock is CLKIN except when 14.3 MHz is selected as the source. The internal ISA state machine clock runs at twice the externally generated ISA BUSCLK rate. The CLKIN frequency in parenthesis indicates which settings to use to yield an 8.00 or 8.33MHz BUSCLK rate. CLKIN will be the same frequency as the CPU when using the 1X mode, or twice the CPU speed when using the 2X mode (see CLKIN pin description).

0000	CLKIN/20	(Power-up default. 160 MHz CLKIN)
0001	CLKIN/16	(133.3 MHz CLKIN)
0010	CLKIN/12	(100 MHz CLKIN)
0011	(Reserved)	
0100	(Reserved)	
0101	(Reserved)	
0110	(Reserved)	
0111	14.3 MHz clock input (a multifunction pin) divided by 2	
1000	CLKIN/10	(80 MHz CLKIN)
1001	CLKIN/8	(66.7 MHz CLKIN)
1010	CLKIN/6	(50 MHz CLKIN)
1011	CLKIN/5	(40 MHz CLKIN)
1100	CLKIN/4	(32 or 33.3 MHz CLKIN)
1101	CLKIN/3	(25 MHz CLKIN)
1110	(Reserved)	
1111	(Reserved)	

7:4 (Reserved)

This register is usually set as follows:

0Dh for CLKIN = 25 MHz (CLKIN/3)
0Ch for CLKIN = 33.3 MHz (CLKIN/4)
0Bh for CLKIN = 40 MHz (CLKIN/5)
0Ah for CLKIN = 50 MHz (CLKIN/6)
09h for CLKIN = 66.7 MHz (CLKIN/8)
07h for BUSCLK = 14.3 MHz / 2 *

* The 14 MHz mode requires 14.3 MHz brought into the 4041 on either the LDEV1# or GPA pin, and Configuration Register 3Ch programmed to select 14.3 MHz.

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
07	ISA clock sel	-	-	-	-	ISA clk3	ISA clk2	ISA clk1	ISA clk0

08-0F (4045 chip).

Index Bits Description

10 **DRAM Timing** Default = 00. Typical setting for 70 ns DRAM = 01h for 33.3 MHz and 1X clock mode, or 00h for 66.7 MHz CLKIN and 2X clock mode.

1:0 Read Timing Mode.

00 3-2-2-2 page hits (default). **Valid in 2X clock mode only.**
01 4-3-3-3 page hits
10 5-4-4-4 page hits (50MHz)

2 Read cycle RAS to CAS timing

0 CAS generated 2 T states (1.5 for 3-2-2-2 mode) after RAS (default)
1 CAS generated 3 T states (2.5 for 3-2-2-2 mode) after RAS

3 (Reserved)

4 Write wait states, single write.

0 1 wait state writes (3 T state minimum cycle)
1 2 wait state writes (4 T state minimum cycle)

5 RAS Precharge Time.

0 2 T states of RAS precharge time.
1 3 T states of RAS precharge time.

6 Write burst Timing.

0 -2-2-2 burst write timing (default)
1 -3-3-3 burst write timing

7 Refresh RAS pulse width.

0 3 T states
1 4 T states (40MHz 80ns RAMs, 50MHz 70 & 80ns RAMs)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
10	dram timing	ref RAS width	write burst	RAS prechg	write ws	-	RAS-CAS	read burst1	read burst0

11 **DRAM Setup** Default = 00. Typical setting = C0h for no interleaving, or C3h to interleave banks 0 and 1.

0 Interleave for bank 0. Should be zero if bank 4 is enabled.

If a bank is interleaved, its address range is doubled and it is active only when the interleave bit (A11 or A12) compares. Banks 0 & 2 compare the interleave bit to a 0, and banks 1 & 3 compare it to a 1. For proper interleaving, two banks must be the same size and have the same starting address. See the text on interleaving for more information on what must be done to interleave properly.

0 Do not interleave the bank
1 Interleave.

1 Interleave for bank 1. Should be zero if bank 5 is enabled.
2 Interleave for bank 2. Should be zero if bank 6 is enabled.
3 Interleave for bank 3. Should be zero if bank 7 is enabled.
5:4 (Reserved)
6 DRAM refresh enable.

0 DRAM refresh disabled. ISA bus refresh handshaking may still occur.
1 DRAM refresh enabled.

7 Enable Local DRAM parity. 1=enabled. This bit provides an additional way to disable parity checking over and above I/O Port 61. For parity checking to occur, both this bit and I/O Port 61 bit 2 must be enabled (a 0 for port 61). The NMI mask must also be set to send the NMI to the CPU (I/O port 70 bit 7). For selective parity enabling by bank, see Index 1Dh.

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
11	dram setup	En Loc Par	dram refresh	ref rate	-	Int 3	Int 2	Int 1	Int 0

Index Bits Description

12 **DRAM Configuration, blocks 0 & 1.** Default = 00. Typical setting = 02h for single-bank 1M deep DRAM (4MB total). Block 0 consists of banks 0 and 4 (RAS0# & RAS4#). Block 1 consists of banks 1 and 5 (RAS1# & RAS5#).

2:0 Banks 0 & 4 DRAM type
- 000 bank disabled
- 001 256K deep DRAMs (256Kx1 or 256Kx4)
- 010 1M deep DRAMs (1Mx1 or 1Mx4)
- 011 4M deep DRAMs (4Mx1 or 4Mx4)
- 100 16M deep DRAMs (16Mx1)
- 101 (Reserved).
- 11x (Reserved).

3 # banks installed in block 0
- 0 Bank 0 only (RAS0#)
- 1 Banks 0 & 4 (RAS0# & RAS4#)

6:4 Banks 1 & 5 DRAM type
Bit definitions same as for banks 0 & 4.

7 # banks installed in block 1
- 0 Bank 1 only (RAS1#)
- 1 Banks 1 & 5 (RAS1# & RAS5#)

13 **DRAM Configuration, blocks 2 & 3.** Default = 00. Typical settings: 00h if blocks 2 and 3 are empty, or C0h if block 2 is empty and block 3 contains 16M deep dual-bank DRAM (128MB total). Block 2 consists of banks 2 and 6 (RAS2# & RAS6#). Block 3 consists of banks 3 and 7 (RAS3# & RAS7#).

2:0 Banks 2 & 6 DRAM type
Bit definitions same as for banks 0 & 4.

3 # banks installed in block 2.
- 0 Bank 2 only (RAS2#)
- 1 Banks 2 & 6 (RAS2# & RAS6#)

6:4 Block 3 DRAM type
Bit definitions same as for banks 0 & 4.

7 # banks installed in block 3.
- 0 Bank 3 only (RAS3#)
- 1 Banks 3 & 7 (RAS3# & RAS7#)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
12	dram config	#banks1	size1	size1	size1	#banks0	size0	size0	size0
13	dram config	#banks3	size3	size3	size3	#banks2	size2	size2	size2

Index Bits Description

14-17 **DRAM block 0-3 Starting Address** Default = 00. Typical setting = 00h for the lowest addressed block in the system, 01h for a block that starts at 1MB, and so on. Largest blocks should be programmed for lowest starting addresses.

14 DRAM block 0 Starting address
15 DRAM block 1 Starting address
16 DRAM block 2 Starting address
17 DRAM block 3 Starting address

7:0 A27-A20 of starting address

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
14	start add 0	A27	A26	A25	A24	A23	A22	A21	A20
15	start add 1	A27	A26	A25	A24	A23	A22	A21	A20
16	start add 2	A27	A26	A25	A24	A23	A22	A21	A20
17	start add 3	A27	A26	A25	A24	A23	A22	A21	A20

18 **Video area shadow and local bus control.** Default = 00. Typical setting = 80h.

0 A0000-AFFFF Shadow enable.
- 0 Access goes to the ISA bus.
- 1 Access goes to local DRAM.

1 B0000-BFFFF Shadow enable.
- 0 Access goes to the ISA bus.
- 1 Access goes to local DRAM.

2 Shadow RAM L1 cache disable
- 0 Shadow RAM is cacheable in L1
- 1 L1 Cache disabled for shadow RAM (A0000:FFFFF DRAM).

Do not set this bit to '1' unless bit 3 is also '1'. (This restriction does not apply to Index 18h bit 7 or Index 94h bit 3.)

3 Shadow RAM L2 cache disable
- 0 Shadow RAM is cacheable in L2.
- 1 L2 Cache disabled for shadow RAM (A0000:FFFFF DRAM).

4 Local Bus time-out
- 0 No time out (default)
- 1 Time out enabled

5 LDEV# Sample point.
- 0 end of first T2.
- 1 end of second T2. This delays the start of all ISA bus accesses.

6 Shadow RAM WB / WT#
- 0 Shadow RAM may be write back in L1.
- 1 Shadow RAM is write through in L1 (may be WB in L2)

7 Write Protect Method.
- 0 Write protected DRAM <u>is not</u> put in the 486 cache.
- 1 Write protected DRAM <u>is</u> placed in the 486 cache, and EADS# is generated on all writes.

If bit 7 is '1', bit 6 typically should also be set to '1' when using shadow RAM for SMM with a writeback CPU, unless SMM is non-cacheable in L1. (see section 5.9.5.1.)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
18	vid shad, etc.	wprotmode	eads lbm	local sample	local timeout	sh ram L2 dis	sh ram L1 dis	shadow B0	shadow A0

Index Bits Description

19 **DRAM shadow read enable.** Default = 00. Typical setting = 43h if system BIOS and video BIOS are both shadowed. If a bit in the list below is 0, reads in the corresponding address range come from the ISA bus. If a bit is a 1, reads in that range come from local DRAM. See also Index 1Dh.

1A **DRAM shadow write enable.** Default = 00. Typical setting = 00h for write-protected shadow RAM. If a bit in the list below is 0, writes to the corresponding address range go to the ISA bus. If a bit is a 1, writes to that range go to the local DRAM. See also Index 1Dh.

1B **ROMCS enable.** Default = 60h. Typical setting = 60h for system BIOS (Fxxxxh) and extension firmware (Exxxxh) both in ROM, or 63h for system BIOS (Fxxxxh) and extension firmware (Exxxxh) and video BIOS (C0xxxh-C7xxxh) all in one ROM. If a bit is 1, ISA bus reads from that location activate ROMCS#. If a bit is a 0, ROMCS# is not activated. ROMCS# will not be activated if an access is directed to local DRAM, i.e., indexes 19h and 1Ah have precedence over 1Bh. Bit 7 determines whether to activate ROMCS# on writes.

Bit assignments for registers 19, 1A, and 1B:
0 C0000-C3FFF.
1 C4000-C7FFF.
2 C8000-CBFFF.
3 CC000-CFFFF.
4 D0000-DFFFF.
5 E0000-EFFFF.
6 F0000-FFFFF.

Note: Memory read or write cycles in the range FFFx xxxxh (top 1MB) always go to ROM (ROMCS# generated, ISA cycle timing) regardless of the value in Index 1Bh, and WPROT# (Index 3Dh) remains high.

19,1A 7 (Reserved)
1B 7 Activate ROMCS on writes also. default=0.
0 do not activate on writes.
1 activate ROMCS# on ISA writes to the addresses selected by bits 0-6.

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
19	shadow rd	-	RD F0000	RD E0000	RD D0000	RD CC000	RD C8000	RD C4000	RD C0000
1A	shadow wr	-	WR F0000	WR E0000	WR D0000	WR CC000	WR C8000	WR C4000	WR C0000
1B	ROMCS	Rom WR	Rom F0000	Rom E0000	Rom D0000	Rom CC000	Rom C8000	Rom C4000	Rom C0000

1C **SMM shadow enable.** Default = 00. Typical setting = 77h.
This register is used instead of registers 19 and 1A bits 6:4 while in SMM mode, allowing DRAM to be enabled only during SMM mode. If a bit is 0, reads or writes from that location come from the ISA bus. If a bit is a 1, reads or writes from that location come from local DRAM. See also Index 1Dh.
0 D0000-DFFFF read enable.
1 E0000-EFFFF read enable.
2 F0000-FFFFF read enable.
3 (Reserved)
4 D0000-DFFFF write enable.
5 E0000-EFFFF write enable.
6 F0000-FFFFF write enable.
7 (Reserved)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
1C	shad SMM	-	sm w F0000	sm w E0000	sm w D0000	-	sm r F0000	sm r E0000	sm r D0000

Index Bits Description

1D **DRAM Block Parity Enable and 32KB Shadow Control.** Default = 00. Typical setting = 0Fh to enable parity checking on all four blocks. This register allows parity to be enabled or disabled in individual DRAM blocks. Parity checking occurs only if the corresponding DRAM block is enabled via Index 12h or 13h. See also Index 11h bit 7. 0=Parity not checked. 1=Parity checked.

0 Block 0 (RAS0# and RAS4#)
1 Block 1 (RAS1# and RAS5#)
2 Block 2 (RAS2# and RAS6#)
3 Block 3 (RAS3# and RAS7#)

Bits 7:4 below provide a means of enabling shadow RAM with 32KB granularity in the Exxxxh and Fxxxxh ranges. This allows flash ROM programming when only a 64KB memory segment is available to the software. Shadow RAM is enabled by either these bits or the bits in Indexes 19h and 1Ah. Unlike 19h and 1Ah, these bits enable both reads and writes, both for user mode and SMM. These bits usually should be turned on before turning off the bits in Indexes 19h and 1Ah to make sure that the executing program remains accessible.

4 E0000h-E7FFFh Shadow RAM enable.
0 Disabled (shadow RAM controlled by Indexes 19h and 1Ah).
1 Enable Shadow RAM read and write for this area for both user mode and SMM.

5 E8000h-EFFFFh Shadow RAM enable.
6 F0000h-F7FFFh Shadow RAM enable.
7 F8000h-FFFFFh Shadow RAM enable.

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
1D	shdw/parity	F8shdw	F0shdw	E8shdw	E0shdw	ParBlk3	ParBlk2	ParBlk1	ParBlk0

1E **Secondary DRAM Timing register.** Default = 00. Typical setting should be same as Index 10h or one speed slower. This register sets the timing mode used for the DRAMs when HLDA is high, i.e., when DRAM is accessed by a local bus master, ISA master, or DMA transfer.

1:0 Read Timing Mode.
00 3-2-2-2 page hits (default). Valid only in 2X clock mode.
01 4-3-3-3 page hits
10 5-4-4-4 page hits (50MHz)

2 RAS to CAS timing
0 CAS generated 2 T states (1.5 for 3-2-2-2 mode) after RAS (default)
1 CAS generated 3 T states (2.5 for 3-2-2-2 mode) after RAS

3 (Reserved)

4 Write wait states, single write.
0 1 wait state writes.
1 2 wait state writes.

5 RAS Precharge Time.
0 2 T states of RAS precharge time.
1 3 T states of RAS precharge time.

6 Write burst Timing.
0 -2-2-2 burst write timing (default)
1 -3-3-3 burst write timing.

7 (Reserved) (Refresh RAS pulse width always uses the register 10h mode)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
1E	dram sec tim	-	write burst	RAS prechg	write ws	-	RAS-CAS	read burst1	read burst0

Index Bits Description

20 **Cache Controller Mode.** Default = 00. Typical setting = 09h for normal operation with a dual-bank secondary cache in writeback mode, or 03h for cache initialization.

0 WRMODE Write Mode. Selects the write-hit policy. Never change the state of this bit while bit 3 is '1'.

- 0 Write through
- 1 Write back (preferred for highest performance)

1 INITCACHE Initialize cache. All cacheable memory read cycles update the cache. No castouts are performed. Write-hit cycles are executed as write through. This bit should never be changed from '0' to '1' unless bit 3 is '0' or is changed to '0' at the same time. While this bit is '1', bit 0 has no effect. Similarly, bit 3 should not be changed from '0' to '1' unless bit 1 is '0' or is changed to '0' at the same time.

- 0 Cache responds to read hits
- 1 Read hits are disabled.

2 FRZCDIR Freeze cache directory (intended for diagnostic testing)

- 0 Normal cache operation
- 1 No directory update.

3 Cache Enable. 0 (default) = disable cache. 1 = enable cache.

Never change this bit from '0' to '1' without initializing the cache first. Also, if bit 0 is '1', never change bit 3 from '1' to '0' without performing a cache flush first.

5:4 Cache write timing mode

- 00 2-1-1-1 burst write (for typical cache configurations)
- 01 (Reserved)
- 10 3-2-2-2 burst write (for single-bank and slow SRAM)
- 11 4-2-2-2 burst write

7:6 Cache read timing mode

- 00 2-1-1-1 bursts (for typical cache configurations)
- 01 2-2-2-2 bursts (for single-bank and slow SRAM)
- 10 3-2-2-2 bursts
- 11 (Reserved)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
20	cache mode	cache read1	cache read0	cache write1	cache write0	cache enable	frzcdir	initcache	wrmode

Index	Bits	Description

21 **Cache Controller Configuration** Default = 00. Typical setting = 40h for single-bank 128KB cache with 8-bit tag, or A0h after upgrade to dual-bank 256KB.

0 (Reserved. Write as 0.)

2:1 Tag width

00	8 bits (default)
01	9 bits
10	11 bits
11	(Reserved).

3 (Reserved. Write as 0. May be used for future expansion of tag width field).

6:4 Cache size:

000	64K bytes (default)
001	128K bytes
010	256K bytes
011	512K bytes
100	1Mbyte

7 Single or Dual bank mode

0	Single bank
1	Dual Bank

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
21	cache config	cache banks	cache size2	cache size1	cache size0	-	tag width 1	tag width 0	tag wr time

22 **Cache Testing Control.** Default = 00. Typical setting = 00h for normal operation.

0 Data SRAM test mode. 0 = disabled (default). 1 disables CAS for DRAM cycles within the test window.

1 Tag RAM test mode. 0 = disabled (default)

2 (Reserved).

3 SRAM and TAG RAM test window select.

0	Test Window for 64K, 128K, & 256K cache: 40000h-7FFFFh;
	for 512K cache: 20000:9FFFF.

1	Test Window for all cache sizes: 100000h-1FFFFFh

5:4 (Reserved)

7:6 Code and data caching. These bits affect secondary cache line fills only. They do not affect cache read or write hits, which must function as normal. The L1 cache (via the KEN# signal) is not affected.

00	Code and data cache.
01	Code only
10	Data only
11	(Reserved)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
22	cache test	code/data1	code/data0	-	-	test window	-	tag test	SRAM test

Index	Bits	Description

23 **Data port for tag testing, bits 0-7.** Default = undefined (may vary from one power up to the next). Typical setting = 00h for normal operation (value has no effect). When tag test mode is enabled, writes to the test window cause the value in this register to be written to bits 0-7 in the TAG RAM. Reads from the test window cause TAG bits 0-7 to be written into this register.

24 **Data port for tag testing, bits 8-10.** Default = undefined (may vary from one power up to the next). Typical setting = 00h for normal operation (value has no effect). See index 23h description. Register bits 0-2 are tag bits 8-10. Register bits 3-7 are reserved.

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
23	cache tag tst	TAG7	TAG6	TAG5	TAG4	TAG3	TAG2	TAG1	TAG0
24	cache tag tst	-	-	-	-	-	TAG10	TAG9	TAG8

25-27 **Do not exist.**

28 **I/O Decode #0 Address Low.** Default = 00. Typical setting doesn't matter (no effect) unless Index 2Bh is non-zero.

7:0 Address bits 7:0 to be compared for the decode.

29 **I/O Decode #0 Address High.** Default = 00. Typical setting doesn't matter (no effect) unless Index 2Bh is non-zero.

7:0 Address bits 15:8 to be compared for the decode.

2A **I/O Decode #0 Size and Mask.** Default = 00. Typical setting doesn't matter (no effect) unless Index 2Bh is non-zero.

2:0 Decode size bits. Provides the mask for bits 6:0 for address.

000	1 byte (port)
001	2 bytes (ports)
010	4 bytes (ports)
011	8 bytes (ports)
100	16 bytes (ports)
101	32 bytes (ports)
110	64 bytes (ports)
111	128 bytes (ports)

3 Mask bit for A7. 0=decode, 1=ignore the address bit.

4 Mask bit for A8. 0=decode, 1=ignore the address bit.

5 Mask bit for A9. 0=decode, 1=ignore the address bit.

6 Mask bit for A10. 0=decode, 1=ignore the address bit.

7 Mask bit for A15:11. 0=decode, 1=ignore the address bits.

Index Bits Description

2B **I/O Decode #0 Configuration.** Default = 00. Typical settings: 00h to disable Decode #0, B0h to enable a strobed chip select for I/O read and write, or 08h to enable LDEV# internally for a local bus slave that doesn't provide an LDEV# signal.

2:0 (Reserved)
3 Local bus I/O range. 1= forces the LOCAL# function for the range
4 Activate Chip Select for I/O reads. 0= don't activate, 1= activate. See note in text on limitation.
5 Activate Chip Select for I/O writes. 0= don't activate, 1= activate. See note in text on limitation.
6 Allow Chip Select for ISA Master accesses. 0= disable for master, 1=allow for master.
7 Chip select or strobe.
  0 CS0 acts as a chip select decoded from CPU address and M/IO# only
  1 CS0 is ANDed with the read and/or write strobe.

2C **I/O Decode #1 Address Low.** Default = 00.
2D **I/O Decode #1 Address High.** Default = 00.
2E **I/O Decode #1 Size and Mask.** Default = 00.
2F **I/O Decode #1 Configuration.** Default = 00. To disable Decode #1, use 00h.
See CS0 for descriptions of the bits.

Index cs0	Index cs1	function	D7	D6	D5	D4	D3	D2	D1	D0
28	2C	add low	A7	A6	A5	A4	A3	A2	A1	A0
29	2D	add high	A15	A14	A13	A12	A11	A10	A9	A8
2A	2E	mask/size	mask15:11	mask10	mask9	mask8	mask7	size2	size1	size0
2B	2F	config	cs/stb	master	I/O writes	I/O reads	local	-	-	-

30 **Memory Decode #0 Address Low.** Default = 00. Typical setting = FFh if Decode #0 is disabled.
7:0 Address bits 23:16

31 **Memory Decode #0 Address High** Default = 00. Typical setting = FFh if Decode #0 is disabled.
7:0 Address bits 31:24

Index Bits Description

32 **Memory Decode #0 Size and Destination** Default = 00. To disable Decode #0, write 00h to this register and program Index 3Ch to disable the MEMCS0# output signal. Memory Decode #0 can be used for generating MEMCS0#, creating a hole in local DRAM, forcing an internal LDEV# signal for local bus slaves that don't generate LDEV#, creating a non-cacheable area, or controlling the WB/WT mode of L1 cache in a selected local or non-local address range.

3:0 Range Size

0000	64K byte range	0111	8Mbyte range
0001	128K byte range	1000	16Mbyte range
0010	256K byte range	1001	32Mbyte range
0011	512K byte range	1010	64Mbyte range
0100	1Mbyte range	1011	128Mbyte range
0101	2Mbyte range	11xx	(Reserved)
0110	4Mbyte range		

4 (Reserved).

5 Hole in DRAM

0 Range is allowed in DRAM

1 DRAM within this range is disabled.

6 Local bus area.

0 disabled.

1 Force the LDEV# signal for this range. Accesses to this range will be on the local bus. Setting this bit will NOT prevent a local DRAM access. The memory range must be either outside of the local DRAM decode or bit 5 above must also be set.

7 (Reserved).

33 **Memory Decode #0 Attributes.** Default = 00. Typical setting = 00h if Decode #0 is disabled.

1:0 Cache Mode. This affects the cache mode regardless of the cycle's destination. The CHIPSet does not support caching any memory other than DRAM (which could be SMM memory), however.

00 Use default cache status for this area (based on destination, etc). The defaults are typically: cache DRAM in writeback (if the CPU supports it), do not cache non-DRAM areas.

01 Non-Cache. Do not cache the memory in either L1 or L2 cache.

10 Cache as write through in L1 cache. The L2 cache is unaffected by this mode, and is cached according to the current L2 mode.

11 Cache in write back mode in L1. The L2 cache is unaffected by this mode, and is cached according to the current L2 mode. If the selected address range is not entirely within local DRAM, the L1 WB mode won't work if a local bus master, ISA Master, or DMA read occurs in the selected range and the target memory is unable to back off if needed (due to L1 data in the CPU being more current than the data in the target memory).

7:2 (Reserved)

Index	Bits	Description
34		**Memory decode #1 address low.** Default = 00. See Index 30h.
35		**Memory decode #1 address high.** Default = 00. See Index 31h.
36		**Memory decode #1 size and destination.** Default = 00. See Index 32h.
37		**Memory decode #1 attributes.** Default = 00. See Index 33h.

dec0 Index	dec1 Index	function	D7	D6	D5	D4	D3	D2	D1	D0
30	34	add low	A23	A22	A21	A20	A19	A18	A17	A16
31	35	add high	A31	A30	A29	A28	A27	A26	A25	A24
32	36	destination	-	VL local	hole	-	Size3	Size2	Size1	Size0
33	37	attrib	-	-	-	-	-	-	cache1	cache0

38 **CPU Modes.** Default = 00. Typical setting = 00 for 1X clock mode and WT CPU cache.

0 HITM# Sampling.

- 0 L1 Write Through. DRAM and cache controllers do not wait for HITM#. The HITM# pin is disabled in this mode and may be used as a general purpose input bit.
- 1 L1 Write Back. DRAM and cache controllers wait for HITM# on all non-CPU generated memory cycles.

1 HITM# Sample Point

- 0 2 clocks after ADS# (end of first T2)
- 1 3 clocks after ADS# (end of second T2)

2 SUSPA# Enable

- 0 Disable SUSPA# pin
- 1 Enable SUSPA# pin (intended for Cyrix CPUs)

3:6 (Reserved)

7 CLKIN Mode (Latched state of NMI pin during power-on reset. Not writeable.)

- 0 NMI sampled high during reset, signifying 1X CLKIN mode
- 1 NMI sampled low during reset, signifying 2X CLKIN mode

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
38	CPU modes	CLKINmode					gen EADS	hitm# sample	L1 wb

Index Bits Description

39 **Soft Reset & GATEA20.** Default = 04h. Typical setting = BFh for internal keyboard/mouse controller, or B2h or 02h for external controller.

0 Mouse Interrupt enable on the control link
- 0 Disabled (mouse interrupt not sent to 4045)
- 1 Enabled (send mouse interrupt notification to 4045)

1 LIN pin function.
- 0 Test Function (default).
- 1 Link input (normal operation).

This bit should be set to 1 before enabling 4045 LOUT (Index 09h bit 4). Also, the 4041 LIN pin should have a pull-up resistor to keep it high during and immediately following reset. If 4041 Test Mode is desired (for hardware testing), the 4041 LIN pin may be externally pulsed low while bit 1 in this register is 0.

2 Enable Internal 8042.
- 0 Disabled. Keyboard pins are GATEA20 and KBRESET#.
- 1 Internal 8042 enabled. Keyboard pins are KBCLK and KBDATA.

3 Enable Mouse port
- 0 Disabled.
- 1 Internal 8042 mouse port enabled. Bit 2 must also be set.

4 8042 RESET2 Emulation.
- 0 Disabled.
- 1 The Emulated 8042 KBRESET# function will cause a Soft CPU Reset.

5 8042 GATEA20 Emulation.
- 0 The Emulated 8042 GATEA20 logic ignores 8042 commands.
- 1 8042 commands are enabled to the Emulated 8042 GATEA20 logic

Bits 4 and 5 are optional with either internal or external keyboard controller, but setting these bits normally will provide a performance benefit.

6 CPU reset. Always reads as 0. Effect on write:
- 0 No effect (do nothing).
- 1 Reset the CPU if SMI mode is disabled. Otherwise, generate an SMI to request CPU reset.

7 Disable IOW to 8042 for emulated commands.
- 0 All 8042 commands go to the 8042.
- 1 GATEA20 and RESET2 commands to the 8042 have IOW# blocked to speed up the operation of the 8042 commands.

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
39	kb cont	dis 8042 IOW	CPU reset	EMU kbga20	EMU kbres	IntMouse	Int8042	LIN pin	Mouse Iout

Index	Bits	Description
3A-3B	**(Reserved)**	

3C **Multifunction Pin Selection Register A.** Default = 00

This register determines the function of 4 pins. In the future, more functions may be added. When added to input pins, these bits should be programmed as "General Purpose Inputs", which disables their other functions.

1:0 LDEV1# pin usage. This pin is always an input.

00	Gen Pur Input	10	14MHz clock (to pwr mgmt & BUSCLK)
01	#LDEV1	11	(Reserved)

3:2 LDEV2# pin usage. This pin is always an input.

00	Gen Pur Input	10	KB Inhibit input
01	LDEV2#	11	External Event 1 (to pwr mgmt)

5:4 GPA pin usage. This pin is always an input.

00	Gen Pur Input	10	XDIO#
01	14 MHz clock	11	External Event 0

7:6 GPB pin usage. This pin is always an output.

00	WB / WT# pin	10	IRQ12 (Mouse Interrupt)
01	ISAEN	11	General Purpose Output

3D **Multifunction Pin Selection Register B.** Default = 00

This register determines the function of 4 pins, all of which are outputs.

1:0 RAS6# pin usage. This pin is always an output.

00	RAS6#	10	MEMCS0# (memory chip select 0)
01	MA12	11	General Purpose Output

3:2 RAS7# pin usage. This pin is always an output.

00	RAS7#	10	MEMCS1# (memory chip select 1)
01	IOC4045# (4045 chip sel)	11	General Purpose Output

5:4 GPC pin usage. This pin is always an output.

00	WPROT# (CPU wrt prot)	10	IOCS0# (I/O chip select 0)
01	CACHECS#	11	General Purpose Output

7:6 GPD pin usage. This pin is always an output.

00	DRAMCS#	10	IOCS1# (I/O chip select 1)
01	ISAEN	11	General Purpose Output

3E **General Purpose Output data register.** Default = 00

This register supplies the data for pins which are selected as general purpose Output bits.

0 Data for RAS6# pin when programmed as a general purpose output bit.
1 Data for RAS7# pin when programmed as a general purpose output bit.
2 Data for GPB pin when programmed as a general purpose output bit.
3 Data for GPC pin when programmed as a general purpose output bit.
4 Data for GPD pin when programmed as a general purpose output bit.
5 Data for FLUSH# pin when programmed as a general purpose output bit.
7:6 Function of the FLUSH# pin.

00	FLUSH#
01	General Purpose Output bit.
10	IOCS0# (I/O Chip Select 0)
11	General Purpose Output Bit

Index Bits Description

3F **General Purpose Input data register (read only).**

This register reads the current state of the listed pins. These pins may be read regardless of the selected function for the pins. This register is intended to be read-only. Writing to this register may cause GPOUT signals to malfunction.

0 LDEV1# pin
1 LDEV2# pin
2 GPA pin
3 (Reserved)
4 MCLK pin
5 MDATA pin
7:6 (Reserved)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
3C	GP select	gpBsel1	gpBsel0	GPAsel1	gpAsel0	ldev1sel1	ldev1sel0	ldev0sel1	ldev0sel0
3D	GP select	gpDsel1	gpDsel0	gpCsel1	gpCsel0	ras6sel1	ras7sel0	ras6sel1	ras6sel0
3E	GPout data	flushsel	-	FLUSH#	GPD	GPC	GPB	RAS7#	RAS6#
3F	GPin data	-	-	MDATA	MCLK	-	GPA	LDEV2#	LDEV1#

40 **(Reserved)**

41 **IDE Enables and Control Functions.** Typical setting = C3h for 2 local IDE connectors.

0 IDE Connector 1 enable (Addresses 1F0:1F7, 3F6:3F7)
1 IDE Connector 2 enable (Addresses 170:177, 376:377)
2 IDE Connector 3 enable (Addresses 5F0:5F7, 7F6:7F7)
3 IDE Connector 4 enable (Addresses 570:577, 776:777)
    0 Disabled. Accesses go to the ISA bus. IDEEN#, IDECS0:1# not active.
    1 Enabled. Accesses go to the 4041 IDE logic.

4 IDE Command Start. This specifies the earliest that the IDE commands may go active for local bus data port accesses.
    0 IDE commands may go active at the end of the first T2.
    1 IDE commands may go active at the end of the Second T2

5 IDE Data hold. This bit specifies how long IDEEN# is held low following IDEIOW# going high, and how long RDY# is delayed.
    0 1 T state
    1 2 T states.

6 Force Defaults.
    0 Default value forcing is disabled.
    1 IDECS0:1#, XA0:1, and SBHE# are forced to 0 by default.

7 Use SBHE# for A2.
    0 SBHE# is always SBHE#. Address setup timer starts at T1.
    1 SBHE# is used for IDEA2. Address setup timer starts when the addresses are forced to their defaults.

Index Bits Description

42 **IDE Drive 0:3 Timing mode.** Typical setting = F4h for 2 local IDE connectors.

1:0 Drive 0 Timing Mode (First IDE connector Drive 0)
3:2 Drive 1 Timing Mode (First IDE connector Drive 1)
5:4 Drive 2 Timing Mode (Second IDE connector Drive 0)
7:6 Drive 3 Timing Mode (Second IDE connector Drive 1)

00	Use Timing Mode A
01	Use Timing Mode B
10	(Reserved)
11	ISA bus cycles are used for the data port (use for slow or 8 bit drives).

43 **IDE Drive 4:7 Timing mode.** Typical setting = 55h.

0 Drive 4 Timing Mode (Third IDE connector, Drive 0)

0	Use Timing Mode A
1	Use Timing Mode B

1 Drive 5 Timing Mode (Third IDE connector, Drive 1)
2 Drive 6 Timing Mode (Fourth IDE connector, Drive 0)
3 Drive 7 Timing Mode (Fourth IDE connector, Drive 1)
4:7 (Reserved)

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
40	-	-	-	-	-	-	-	-	-
41	misc IDE	forceDefault	UseSBHE	datahold	tstatestart	ideen76	ideen54	ideen32	ideen10
42	tim drv3:0	drv3sped1	drv3sped0	drv2sped1	drv2sped0	drv1sped1	drv1sped0	drv0sped1	drv0sped0
43	tim drv7:4					drv7sped	drv6sped	drv5sped	drv4sped

44 **IDE Timing A Read and Write pulse Widths**

3:0 Read Pulse Width for Timing A. Selects 1 to 16 CPU clocks (1x) for the IDEIOR pulse width on data port reads.

0000	16	0100	4	1000	8	1100	12
0001	1	0101	5	1001	9	1101	13
0010	2	0110	6	1010	10	1110	14
0011	3	0111	7	1011	11	1111	15

7:4 Write Pulse Width for Timing A. Selects 1 to 16 CPU clocks (1x) for IDEIOW pulse width on data port writes. Encoding is the same as reads.

Typical Settings:

IDE Mode	44h (46h)	45h (47h)
"0"	66h	3Ch
"1"	55h	29h
"2"	44h	25h or 2Ch*
"3"	33h	23h
"4"	33h	11h

*IDE Mode 2 (with 2Ch instead of 25h) may be optimum for many existing IDE drives, but the drive specifications should always be checked before using the modes listed above.

Index Bits Description

45 IDE Timing A Command Recovery and Address Setup

3:0 I/O recovery time for IDE commands, Timing A. Selects the minimum number of clocks which IDEIOR# and IDEIOW# must remain high between accesses. Used only by the local bus IDE timing state machine. Encoding is as follows:

0000	16	0100	4	1000	8	1100	12
0000	24	0100	4	1000	8	1100	16
0001	1	0101	5	1001	10	1101	18
0010	2	0110	6	1010	12	1110	20
0011	3	0111	7	1011	14	1111	22

6:5 Address setup time to IDE commands, Timing A. Selects the minimum number of clocks between the address being setup and the command going active. The timing begins when the default values are forced onto the address and CS pins, or from the start of T1 if SBHE# is not set to provide A2 (A2 provided from the CPU).

00	4	10	2
01	1	11	3

7:6 (Reserved)

46 IDE Timing B Read and Write pulse Widths

See register 44

47 IDE Timing B Command Recovery and Address Setup

See register 45

Index	Function	D7	D6	D5	D4	D3	D2	D1	D0
44	IDE timing A	wrA3	wrA2	wrA1	wrA0	rdA3	rdA2	rdA1	rdA0
45	IDE timing A			asuA0	asuA0	inacA3	inacA2	inacA1	inacA0
46	IDE timing B	wrB3	wrB2	wrB1	wrB0	rdB3	rdB2	rdB1	rdB0
47	IDE timing B			asuB0	asuB0	inacB3	inacB2	inacB1	inacB0

80 EventA Selection 0: Interrupts.

Setting these bits to a 1 will enable the specified occurrence to generate an EventA, which normally restarts TimerA. EventA has no other function. TimerA, in turn, can automatically slow the CPU clock or trigger an SMI, or both. If the bit is a 0, the specified occurrence will be ignored. The events selected indicate system activity, and that the system should not be slowed down. IRQs are detected by the corresponding INTA cycle. INTR is detected by a high level.

0 IRQ0 (Timer Tick)
1 IRQ1 (Keyboard)
2 IRQ3,4,5, or 7 (serial and parallel ports)
3 IRQ6 or 14 (floppy and hard disk)
4 IRQ9 or 13 (video or coprocessor)
5 IRQ8,10,11,12, or 15 (Misc Interrupts)
6 INTR. Any Time INTR goes high (indicating an unmasked interrupt)
7 NMI. A low to high transition in the NMI pin.

Index Bits Description

81 EventA Selection 1: I/O And Memory Accesses.

0 Disk accesses. Which ones are determined by the Index 86h.
1 Serial or Parallel Ports. Which addresses are determined by Index 86h.
2 Keyboard controller (60h & 64h)
3 Video Memory (A0000h to BFFFFh)
4 Ext0 Pin (A multifunction pin). Polarity, etc, selected elsewhere.
5 Master (HLDA high). Any VL Master, ISA Master, or DMA cycle
6 Programmable I/O address 0 (see Indexes 28:2B)
7 Programmable I/O address 1 (see Indexes 2C:2F)

index	function	D7	D6	D5	D4	D3	D2	D1	D0
80	EventA	NMI	INTR	IRQ8,10:12,15	IRQ9,13	IRQ6,14	irq3,4,5,7	IRQ1	IRQ0
81	EventA	prog IO 1	prog IO 0	master	Ext0	video mem	kb	com/lpt	disk

82 WakeA Event Selection 0. Interrupts.

Setting these bits to a 1 will enable the specified occurrence to generate WakeA, which may generate an SMI and/or switch the CPU clock back to full speed. If the bit is a 0, the specified occurrence will be ignored. The events selected indicate an occurrence which should bring the system back to operating speed due to either the operator or an external event. All events are detected identically to EventA and the bit positions are identical. Only INTR, NMI (i.e., IOCHCK#), and Ext0 are detectable as WakeA events while the CPU clock is stopped.

0 IRQ0 4 IRQ9,13
1 IRQ1 5 IRQ8,10,12,15
2 IRQ3,4,5,7 6 INTR
3 IRQ6,14 7 NMI

83 WakeA Event Selection 1. I/O And Memory Accesses.

Identical selections as register 81.

0 Disk 4 Ext0 pin
1 Serial/Parallel Ports 5 Master
2 Keyboard 6 Programmable I/O 0
3 Video Memory 7 Programmable I/O 1

index	function	D7	D6	D5	D4	D3	D2	D1	D0
82	WakeupA	NMI	INTR	IRQ8,10:12,15	IRQ9,13	IRQ6,14	irq3,4,5,7	IRQ1	IRQ0
83	WakeupA	prog IO 1	prog IO 0	master	Ext0	video mem	kb	com/lpt	disk

Index Bits Description

84 EventB selection.

Setting these bits to a 1 will enable the specified occurrence to generate an EventB, which normally restarts TimerB. EventB has no other function. TimerB, in turn, can automatically stop the CPU clock or trigger an SMI, or both. If the bit is a 0, the specified occurrence will be ignored. The events selected usually indicate short term system activity, and that the system should not be slowed down. The events available are those which will often not be selected to reset TimerA, but should reset TimerB.

0 IRQ0
1 Ext1 Pin (A multifunction pin).
2 INTR
3 Video Memory.
4 Keyboard
5 (Reserved)
6 Programmable I/O range 0
7 EventA (the output of the EventA logic)

index	function	D7	D6	D5	D4	D3	D2	D1	D0
84	EventB	EventA	prog IO 0		keyboard	video mem	INTR	Ext 1	IRQ0

85 WakeB selection.

Setting these bits to a 1 will enable the specified occurrence to generate a WakeB, which normally restarts a stopped clock and may generate an SMI. If the bit is a 0, the specified occurrence will be ignored. The events selected usually indicate any request for CPU usage. The events available are those which can occur when the clock is stopped.

0 (Reserved)
1 INTR
2 NMI going from low to high (e.g., IOCHCK#)
3 SMI# low.
4 (Reserved)
5 Alternate master active (HLDA active)
6 External pin 1 as selected by register 8Fh
7 WakeA (the output of the WakeA logic)

index	function	D7	D6	D5	D4	D3	D2	D1	D0
85	WakeupB	wakeA event	Ext 1	master		SMI	NMI	INTR	-

86 Port Selection for Events.

Bits 0,1, and 2 of this register selects which serial, and parallel ports indicate an event when the COM/LPT bit is set in the EventA or WakeA registers. Bits 4, 5, and 6 select which I/O ports cause an event when the DISK bit is set in the EventA or WakeA registers.

0 COM1 and COM2 addresses included in COM/LPT events (3F8:3FF & 2F8:2FF)
1 COM3 and COM4 addresses included in COM/LPT events (3E8:3EF & 2E8:2EF)
2 LPT1, LPT2, and LPT3 addresses included in COM/LPT events (3BC:3BE, 378:37F, & 278:27F).
3 (Reserved)
4 Floppy ports included in DISK events (3F0:3F1 & 3F3:3F5). 3F2 is NOT included.
5 IDE1 ports included in DISK events (1F0:1F7, 3F6:3F7)
6 IDE2 ports included in DISK events (170:177, 376:377)
7 (Reserved)

index	function	D7	D6	D5	D4	D3	D2	D1	D0
86	select which		IDE2	IDE1	floppy		lpt1,2,3	com3&4	com1&2

Index Bits Description

87 **Interrupt Acknowledge 1 base.** Typical setting = 70h for use with DOS, subject to change automatically whenever the INTC2 interrupt vector base is changed by software. This register is used by the 4041 to determine whether an INTA cycle was caused by IRQ8:15 or not, since the 4041 does not receive the IRQ pins directly but does have access to the interrupt vector returned to the CPU during an INTA cycle. (See also Indexes 80h and 82h.) If the INTA vector matches the contents of this register (bits 7:3), it came from IRQ8:15. If not, the interrupt came from IRQ0:7. The vector itself (bits 2:0) then indicates which specific IRQ within the group is being acknowledged. Writing the interrupt vector base via A0h and A1h automatically causes the value in this index register to match the new value, but writing to this index does not alter the value written via A0h and A1h since the actual INTC2 resides in the 4045. Software or BIOS normally should never write to this register (Index 87h), since its value automatically tracks whatever is written to INTC2.

2:0 (Reserved)

7:3 Specifies the upper 5 bits of the interrupt acknowledge vector which corresponds to IRQ8:15. Typically set (via ports A0h and A1h) to 01110 by DOS, or 01011 by Windows.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
87	INTA base	intabase7	intabase6	intabase5	intabase4	intabase3	-	-	-

88 TimerA Control Register

This register provides the rate and function of TimerA and the function of the WakeA Event. TimerA may also cause an SMI. This is enabled in the SMI enable register.

2:0 TimerA count rate.

000	Off. Counter remains at restart value.
001	64uS
010	1mS
011	16mS
100	256mS
101	4 seconds
110	64 seconds
111	(Reserved)

3 (Reserved)

4 Slow Down the clock on TimerA timeout.

0 Disabled

1 TimerA timing out will cause the CPU clock to be slowed down. This is done by either the internal divider or external synthesizer, as specified elsewhere.

6:5 (Reserved)

7 Switch to the full speed CPU clock on WakeA Events.

0 Disabled.

1 WakeA event switches the clock back to full speed.

89 TimerA Count Register

Writing to this register sets the count value. A value between 1 and FF may be written. This value is reloaded in the timer each time an EventA occurs. Reading this register gives the current value of the timer. With the rate set to OFF, the current value of the timer will be the restart value last programmed here.

7:0 TimerA restart value.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
88	timerA cntrl	WakeAFast			TimerASlow		rateA2	rateA1	rateA0
89	timerA	tmrA7	tmrA6	tmrA5	tmrA4	tmrA3	tmrA2	tmrA1	tmrA0

Index Bits Description

8A TimerB Control Register

This register provides the rate and function of TimerB and the function of the WakeB Event. TimerB may also cause an SMI. This is enabled in the SMI enable register.

2:0 TimerB count rate.

000	Off.	100	256mS
001	64uS	101	4 seconds
010	1mS	110	64 seconds
011	16mS	111	(Reserved)

3 (Reserved)

4 Stop the CPU clock on TimerB timeout.

0 Disabled

1 TimerB timing out will cause the CPU clock to be stopped. The selected Stop-Clock protocol will be followed, as specified elsewhere.

7:5 (Reserved)

8B TimerB Count Register

Writing to this register sets the count value. A value between 1 and FF may be written. This value is reloaded in the timer each time an EventB occurs. Reading this register gives the current value of the timer. With the rate set to OFF, the current value of the timer will be the restart value last programmed here.

7:0 TimerB restart value.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
8A	timerB cntl	-	-	-	Stop clk	-	rateB2	rateB1	rateB0
8B	timerB	tmrB7	tmrB6	tmrB5	tmrB4	tmrB3	tmrB2	tmrB1	tmrB0

8C Time Base Selection & Slow Clock frequency

This register selects the clock source and the divider necessary to make an approximately 1MHz time base for the timer and PLL delay functions.

2:0 Divider Selection.

000	Divide by 24.	100	Divide by 14
001	Divide by 32	101	(Reserved)
010	Divide by 40	11x	(Reserved)
011	Divide by 48		

3 Clock source selection.

0 CLKIN (divided by 2 if 2x clock mode).

1 Multifunction pin (used for 14.31818MHz clock in).

6:4 Slow clock divider. Specifies the divider used for the system clock when the slow mode is selected. No switching is actually done for the 000 setting.

000	Full speed	100	Divide by 8
001	Divide by 2	101	(Reserved)
010	Divide by 4	110	(Reserved)
011	Divide by 6	111	(Reserved)

7 STPCLK# pin usage

0 STPCLK# pin is in fact STPCLK#

1 STPCLK# pin is switched to the CLKSPEED pin. 1=Full Speed, 0=reduced speed. May be used to control an external clock generator.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
8C	Time Base	stpclk pin#	slowclk2	slowclk1	sloclk0	cksel	div2	div1	div0

Index Bits Description

8D Clock switching modes.

These bits select the clock switching and stopping modes.

0 Stop On Halt. 1=Stop the CPU clock on a HALT bus cycle.

1 Stop On Halt while Slow. 1=Stop the CPU clock on a HALT bus cycle only when the CPU clock is in SLOW mode.

2 Stop Clock Mode.
- 0 Pull STPCLK# only.
- 1 Pull STPCLK# and then actually stop the CPU clock.

3 Pull STPCLK# when switching the CPU clock.
- 0 Do not pull STPCLK# when switching the clock frequency.
- 1 Activate STPCLK# before any clock change.

6:4 PLL delay. Delay between changing the clock frequency and removing STPCLK#

000	No delay	100	256uS delay
001	32uS delay	101	512uS delay
010	64uS delay	110	1mS delay
011	128uS delay	111	2mS delay

7 Wait for Stop Clock Acknowledge bus cycle.
- 0 Do no wait for the Stop Clock Acknowledge bus cycle.
- 1 Wait for Stop Clock Acknowledge before changing the clock.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
8D	Stop Clock	waitstopack	plldelay2	plldelay1	plldelay0	stpclk4slow	stopmode	StopHaltnslo	StopOnHalt

8E Software Commands and Status.

Writing to this register causes specific actions to be taken. If the bit is a 1 the action occurs. If it is a 0 no action is taken for that function. There is no actual register which holds the values written here. Reading this register provides status information for internal hardware functions.

Writing:

0 GoSlow command. Writing a 1 causes the clock to be slowed down by whatever means is selected. Bits 0 and 1 should not both be written as 1s.

1 GoFast command. Writing a 1 causes the clock to return to full speed.

2 Stpclk. This will cause the CPU clock to be stopped using the selected stop clock method.

3 Generate an SMI. Writing a 1 will generate a software SMI.

7:4 (Reserved, write as 0.)

Reading:

0 Clock Speed. 0=slow clock, 1=full speed.

7:1 (Reserved)

index	function	D7	D6	D5	D4	D3	D2	D1	D0
8E	Commands					GenSMI	Stpclk	GoFast	GoSlow

8F External Event pin function.

1:0 EXT0 function:

00	Active low level triggered
01	Active high level triggered
10	Active low edge triggered (negative edge)
11	Active high edge triggered (positive edge)

3:2 EXT1 function (same bit definitions as EXT0)

7:4 (Reserved)

Index Bits Description

90 SMI Status Register 0

This register indicates what events are currently causing an SMI. The bit will be a 1 if it is currently pulling SMI low. Writing a 1 to a bit location resets that bit (except for bit 7).

0 TimerA Timeout
1 TimerB Timeout
2 WakeA Event
3 WakeB Event
4 I/O Restart
5 Halt cycle
6 (Reserved).
7 Any bit in SMI status Register 1.

91 SMI Status Register 1

Additional SMI sources. The bit will be a 1 if it is currently pulling SMI low. Writing a 1 to a bit location resets that bit. The OR of the bits in this register appear in bit 7 of the SMI Status Register 0.

0 Software SMI
1 CPU restart request.
2 External SMI 0
3 External SMI 1
7:4 (Reserved)

92 SMI Enable Register 0

This register enables SMI sources. 0 disables, 1 enables.

0 TimerA Timeout
1 TimerB Timeout
2 WakeA Event
3 WakeB Event
4 I/O Restart
5 Halt cycle
6 (Reserved).
7 Global SMI enable bit. Disables all SMI sources if 0. SMM routines should set this bit to 0 and then back to 1 again before leaving SMM, to guarantee an edge on the SMI# pin if any SMI sources are still active. Otherwise, the CPU may fail to return to SMM in response to the active source. (This is not necessary if no other sources are active, indicated by reading 00h in Index 90h.)

93 SMI Enable Register 1

Additional SMI enables.

0 Software SMI
1 CPU restart request.
2 External SMI 0 (EXT0 pin)
3 External SMI 1 (EXT1 pin)
6:4 (Reserved)
7 SMI# Mode
 0 Intel Mode
 1 Cyrix Mode

index	function	D7	D6	D5	D4	D3	D2	D1	D0
90	SMI statusA	next register	-	Halt	I/O Restart	WakeupB	WakeupA	TimerB	TimerA
91	SMI statusB	-	-	-	-	External1	External0	CPU restart	Software smi
92	smi enableA	global enable	-	Halt	I/O Restart	WakeupB	WakeupA	TimerB	TimerA
93	smi EnableB	smi mode	-	-	-	External1	External0	CPU restart	Software smi

Index Bits Description

94 SMM modes.

This register controls the SMM modes of the CPU.

0 SMIACT3 / SMIADS# mode
- 0 SMIACT# function (Intel method)
- 1 SMIADS# function (Cyrix method)

1 Force FLUSH on SMM.
- 0 Do not flush the CPU cache upon entry into SMM
- 1 Flush the CPU cache by pulsing FLUSH# when SMIACT# goes low. This should only be done with an Intel CPU. It need not be done when the SMM windows below 1M are not used.

2 Force A20M# on SMM.
- 0 SMM has no effect on A20M#
- 1 Drive A20M# high with SMIACT#

3 Disable KEN# on SMM.
- 0 SMM has no effect on KEN#
- 1 Drive KEN# high for all SMM accesses.

4 Soft redirection.
- 0 Do not redirect soft resets.
- 1 Disable soft resets and redirect to SMI.

5:7 (Reserved, write as '000')

index	function	D7	D6	D5	D4	D3	D2	D1	D0
94	smm modes				SoftResRedir	DisKen	ForceA20m	flushOnSmm	smiact pin

95 I/O Restart selection.

This register selects which I/O ports cause an I/O restart. A 1 in the bit position causes that range to generate an I/O restart cycle. I/O restart means that the 4041 asserts SMI during a CPU I/O cycle, which the CPU recognizes as an I/O restart SMI. The CPU then enters SMM with status information allowing the SMM routine to determine what I/O resource is being addressed. The SMM routine performs peripheral power-up and re-initialization as needed, then exits SMM, allowing the CPU to re-execute the same I/O operation that caused the SMI. This register should always be set to '00h' if Index 92h bit 4 is '0'. Otherwise, cycle timing may malfunction during I/O to an enabled range.

0 COM1 (3F8:3FF)

1 COM2 (2F8:2FF)

2 LPT2 (378:37F)

3 Floppy and hard disk (1F0:1F7, 3F0:3F1, 3F3:3F5, 170:177, 370:377)

4 VGA I/O (3B0:3BB, 3C0:3DF)

5 Keyboard (60 & 64)

6 Programmable I/O range 0

7 Programmable I/O range 1

index	function	D7	D6	D5	D4	D3	D2	D1	D0
95	I/O restart	prog IO 1	prog IO 0	KB	VGA	floppy/HD	LPT(378)	com2	com1

96 Port 70 shadow register.

This register provides a means of reading back the last value written to port 70h, including the NMI enable bit, which would otherwise be unreadable. This register is read only. Do not write to this register.

7:0 Port 70 shadow read.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
96	Shadow70	port70d7	port70d6	port70d5	port70d4	port70d3	port70d2	port70d1	port70d0

3.4. 84045 I/O Port Addresses

Note: "Default" register values refer to the power-on hardware defaults established automatically following hardware reset, before any alternate values have been written by the BIOS. "Typical" values refer to typical settings for normal system operation.

Addr Bits Description

00-0F **DMA controller #1 (8 bit DMA).**

These ports are contained in the IPC megacell. The DMA controller will NOT respond to accesses to 10-1F (in the original AT 00-0F repeats at 10-1F).

20-21 **Interrupt Controller #1 (IRQ0-7).**

These ports are contained in the IPC megacell. The Interrupt controller does NOT respond to ports 22-3F.

---

22 **Configuration register Address Port.**

Write only port which holds the address of the Chips and Technologies Index register to be accessed through I/O port 23. This register must be written before each access to port 23, even if the same index register is being accessed twice in a row.

23 **Configuration register data.**

Accessing this port accesses the Configuration register pointed to by port 22. A second access to port 23 without writing port 22 in between will be ignored. Unless otherwise noted in the register descriptions, reserved or undefined index registers should not be written to, and reserved bits within a defined index register should be written as zero (or written with the same value previously read).

26 **SMM Configuration Register Address Port**

Write only. The address written here is stored separately from port 22. This register is used as the config register index when port 27 is read or written.

27 **SMM Configuration Register Data Port**

Accessing this port accesses the Configuration register pointed to by port 26. A second access to port 27 without writing to port 27 in between will be ignored. The "accessed" bit is separate for the port 26/27 and 22/23 windows.

40-43 **Timer Chip (8254).**

These ports are contained in the IPC megacell. The timer does NOT respond to ports 44-4F.

---

60 **Keyboard and Mouse Interrupt Clear.**

Reading from port 60h resets the keyboard and mouse interrupt latches, IRQ1 and IRQ12. All other data read/write functions for this port are implemented in the 4041.

Addr Bits Description

61 "Port B"

This is an AT compatible port with miscellaneous information. Bits 0-3 are read/write. Bits 4-7 are read only. Only bits 0 to 5 are contained in the 4045 (bits 2 and 3 are there for readback purposes only). The remainder are in the 4041. On I/O reads the 4045 drives bits 0-5 to their proper values, and bits 6 & 7 to 0s. Default = 20h.

0 Timer 2 gate. This bit enables or disables the 1.19 MHz clock input to Timer 2. The output from Timer 2, in conjunction with bit 1 below, provides the signal for the speaker. If this bit is a 1, Timer 2 is enabled, and (if programmed to do so) will produce a square wave of the programmed frequency. When this bit is a 0, bit 5 below is forced to 1 and the speaker output signal will be high or low depending on bit 1 below.

1 Speaker Data. This bit is ANDed with the output of timer 2 and inverted to produce the signal actually sent to the speaker. When the gate (bit 0 above) is low, this bit gives direct software control of the speaker. The speaker signal will be high or low when this bit is 0 or 1, respectively. When the speaker is idle, bits 0 and 1 normally will both be 0 and the speaker output signal will be high.

2 Enable Parity Check. 0 enables local DRAM parity checking. A 1 disables local DRAM parity checking and clears the local parity error flip-flop. This bit is inverted and sent to the active low preset of a flip-flop. The Q output is PCK# and is fed to the NMI logic. A parity error clocks the flip-flop to a 0. There is also an index register bit to block local DRAM parity errors. It prevents the flip-flop from being clocked. The flip-flop is an F74 on the AT, where the Preset has precedence on Q and Clear has precedence on Q#.

3 Enable IOCHCK. 0 enables the IOCHCK interrupt. A 1 disables IOCHCK and clears the IOCHCK flip-flop. This bit is inverted and sent to the active low clear of a flip-flop. IOCHCK# is sent to the active low Preset input. The Q# output is fed to the NMI logic. The Q output is sent to bit 6 of this register. The flip-flop is an ALS74 on the AT, where the Preset has precedence on Q and Clear has precedence on Q#.

4 Refresh Detect. This read only bit toggles on each refresh. It should toggle whenever timer 1 produces a pulse (about every 15us). This should be done even if ISA refresh is disabled. Some software uses this as a time delay.

5 Timer 2 output. Read only. This bit allows software to monitor the output of timer 2, which is ANDed with bit 1 of this register and inverted to produce the speaker signal. The speaker signal is low when bits 1 and 5 are both 1. If either bit is 0, the speaker output is high. By setting bits 1 and 0 to '01', software can use Timer 2 and bit 5 without generating any speaker output.

6 (Channel Check latch). This bit is not in the 4045. It is contained in the 4041. A 1 indicates that IOCHCK# has been activated. This bit is the Q output of the flip-flop mentioned in bit 3 of this register. The 4045 will drive this bit to a 0 on reads.

7 (Parity Check latch). This bit is not in the 4045. It is contained in the 4041. A 1 indicates that a local parity error has occurred. It is the Q# output of the flip-flop mentioned in bit 2 of this register. The 4045 will drive this bit to a 0 on reads.

Port	D7	D6	D5	D4	D3	D2	D1	D0
61	(Parity err)	(CHCK)	Tmr 2	Ref Detect	chck enable	parity enab	spkr data	tmr2 gate

70 Real Time Clock Address Port & NMI mask.

0-6 Real Time clock address. The value written to these bits becomes the address of RTC/CMOS RAM which will be read or written through port 71.

7 (NMI Mask.) This bit is not in the 4045. It is used in the 4041. This bit is inverted and ANDed with the NMI sources (the OR of several sources). The result of the AND function is NMI to the CPU. This allows the CPU "Non-Maskable Interrupt" to be maskable externally.

Addr Bits Description

71 Real Time Clock Data Port.

Reading or writing this port will read or write the RTC register pointed to by the last write to port 70.

80-8F DMA Page Registers

These ports are contained in the IPC megacell. They provide A16-23 for 8-bit DMA accesses and A17-23 for 16-bit DMA.

80 Read/Write, but not used during normal system operation. This port is written by BIOS routines to indicate BIOS status. A pair of HEX 7 segment LEDs are often put on test boards to display this information.

81 Channel 2 page register

82 Channel 3 page register

83 Channel 1 page register

84-86 Not used.

87 Channel 0 page register

88 Not used.

89 Channel 6 page register

8A Channel 7 page register

8B Channel 5 page register

8C-8E Not used.

8F Refresh page register. Bits 1-3 from this register define the state of SA17-19 during refresh. Since refresh is hidden, the 4045 has no access to SA16 and LA20-23 during refresh. Bit 0 and bits 4-7 are not used. All bits are read/write. Typical setting = 00h.

92 Fast CPU Reset & GATEA20.

0 Fast CPU reset. A 0 to 1 transition activates a CPU reset.

1 Fast GATEA20. ORed with other GATEA20 signals (from 8042, for instance).

  0 Drive A20M- low to the CPU (if all other A20M- sources are low).

  1 Force A20M- high to the CPU, causing the CPU to leave A20 unmasked.

2 (Reserved).

3 RTC Password Protect. If Index 0Ch bit 1 is 0, writing to this bit has no effect and the bit always reads as 0. If Index 0Ch bit 1 is 1, writing 1 to this bit has the following effects: (a) All further reads and writes to the RTC CMOS RAM locations 38:3F are disabled, and (b) this bit remains 1 (reads back as 1) and cannot be cleared except by a system reset (PWRGOOD cycling, causing SYSRESET to cycle).

7:4 (Reserved). read as 0s.

Port	D7	D6	D5	D4	D3	D2	D1	D0
92	-	-	-	RTC passpro	-	-	Fast gatea20	fast reset

A0-A1 Interrupt Controller #2 (IRQ8-15).

These ports are contained in the IPC megacell. The interrupt controller does NOT respond to ports A2-BF.

C0-DF DMA Controller #2 (16 bit DMA).

These ports are contained in the IPC megacell. Only the even numbered ports are used. Reads or writes to the odd numbered ports will access the same register as its corresponding even numbered port.

F0-F1 387 Error Reset.

Writing to either F0 or F1 causes the error latch (which also generates IRQ13) to be cleared. The data is ignored.

3.5. 84045 Configuration Registers

Index Bits Description

01 **DMA WAIT STATE CONTROL.** Default = C0h. Typical setting = 00h.
See Index 0Ah for programming of 4045 internal BUSCLK frequency.

7:6 IPC I/O wait states. Controls the number of wait states (4045 internal BUSCLK cycles) added to any I/O access to an IPC I/O port (00-0Fh, 20-21h, 22-23h, 40-43h, 71h, 80-8Fh, A0-A1h, C0-DFh). Port 70h is decoded outside the IPC megacell does not have any added wait states. The 4045 pulls IOCHRDY as needed to insert the wait states.

00	One wait state
01	Two wait states
10	Three wait states
11	Four wait states

5:4 16-bit DMA wait states. These bits control the number of wait states (DMACLK cycles) inserted during 16-bit DMA transfers.

00	One wait state (default)
01	Two wait states
10	Three wait states
11	Four wait states

3:2 8-bit DMA wait states. These bits control the number of wait states (DMACLK cycles) inserted during 8-bit DMA transfers.

00	One wait state (default)
01	Two wait states
10	Three wait states
11	Four wait states

DMA wait states are defined as DMACLK cycles (see Index 0Ah) added to the IOW# or MEMW# command low time. Minimum command low time (one wait state) is two DMACLK cycles total. The 4041 may pull IOCHRDY as needed to add additional DMA wait states for local bus timing.

1 DMA MEMR# signal extension. In the IBM PC/AT, the assertion of MEMR# is delayed by one DMACLK cycle compared to IOR#. This may not be desirable in some systems.

0	Delay MEMR# by one DMACLK (default, PC/AT compatible)
1	MEMR# not delayed; follows same timing as IOR#

0 DMACLK clock select.

0	DMACLK cycle is two 4045 BUSCLK cycles (default, as in PC/AT)
1	DMACLK cycle is one 4045 BUSCLK cycle

Addr Bits Description

08 **Performance control** Default = 00. Typical setting = 00h for no performance reduction, or F0h or more for DX-33 or DX2-66 reduction to 6 or 8 MHz equivalent performance.

6:0 CPU hold pulse width. These bits set the amount of time in which the CPU is put in hold following each AT bus refresh. Once the count is set, the mode must be enabled by a separate register or the "Turbo" button (SLOW# pin). The values below are the number of 4045 internal BUSCLKs (see Index 0Ah) for which the CPU is kept in hold. Normally this occurs about every 15uS.

000000 = No hold request (default)
000001 = Minimum speed reduction (one 4045 BUSCLK)
...
111111 = Maximum speed reduction (127 4045 BUSCLKs).

The programmed value should not exceed the time between refresh cycles. For 15 us refresh rate and 8.33 MHz 4045 BUSCLK, the maximum usable value is about 120 (14.4 us). Values approaching 120 (78h) may be needed to reduce the performance of 486DX-33 or DX2-66 systems to 8 MHz equivalent.

7 486 cache flush. 0 = do not flush cache during slow mode HOLD. 1 = flush cache during each slow mode hold request. Setting this bit to a 1 prevents the 486 from executing from internal cache during the slow mode hold, which generally will be necessary to successfully reduce the performance of a DX-33 or DX2-66 system to 8 MHz equivalent.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
08	Pefrm cntl	486 flush	hold width6	hold width5	hold width4	hold width3	hold width2	hold width1	hold width0

Index Bits Description

09 **4045 Misc Control** Default = 00. Typical setting = 3Ch for external keyboard controller, 7Ch for internal controller.

0 Deturbo (Performance Control Enable). This enables the performance control programmed in Register 08h, and has nothing to do with the slow CPU clock mode. This bit is ORed with the invert of the Turbo Switch (SLOW# input pin).

- 0 Normal mode (default)
- 1 Performance Control Enabled.

1 Preemptive protocol for LGNT#.

- 0 Non-preemptive protocol. The arbitration will not take LGNT# inactive until LREQ# has gone inactive (default).
- 1 Preemptive protocol. The arbitration logic will take LGNT# low when the DMA controller requests the bus. It will wait for LREQ# to go inactive before granting the bus to the DMA controller. This is VL-Bus compatible.

2 Refresh Request Enable. This bit blocks the Timer 1 Refresh request when disabled. This prevents reset problems, which may occur when a refresh request is generated during the reset sequence. The 8254 timing is not disabled at reset.

- 0 Block the Timer 1 refresh requests.
- 1 Enable Timer 1 refresh requests. (Essential for DRAM refresh.)

3 ISA bus refresh enable.

- 0 Disable ISA bus refreshes. The Refresh request is still sent to the 4041 or 4031, and Master Refresh cycles are still performed. Disabling ISA refresh may allow a TTL chip to be deleted in systems that don't need ISA refresh.
- 1 Enable ISA bus refresh. Needed for full ISA compatibility.

4 A20M# / TEST# and 4045 LOUT pin enable function.

- 0 The A20M# / TEST# pin is an input, forcing the 4045 into test mode when it is low. The 4045 LOUT pin is also floated to allow the 4041 LIN pin to be used for the test mode control on the 4041. (Default).
- 1 The A20M# / TEST# pin is an output, driving A20M#. The test mode is disabled. The 4045 LOUT pin is driven.

4041 LIN should be enabled (Index 39h bit 1) before setting this bit to 1.

5 GATEA20 emulation disable.

- 0 The A20M# pin is only the Port 92 GATEA20 signal (which is low after reset). (default).
- 1 The A20M# pin is the OR of the Port 92 GATEA20 and the emulated 8042 GATEA20 information received across the control link.

6 Keyboard interrupt mode.

- 0 IRQ1 is received on the pin directly (from external 8042).
- 1 IRQ1 is received over the control link from the 4041 (internal KBC).

7 Floating point error mode.

- 0 Internal (486) mode. FERR# and IGNNE# pins are provided. IRQ13 is generated internally (default).
- 1 External mode. IRQ13 and INTCLR pins are provided. The remainder of the logic is provided externally. This mode is used where a pin for IRQ13 is required (such as a system with a Weitek Coprocessor).

index	function	D7	D6	D5	D4	D3	D2	D1	D0
09	misc	fpintmode	kbintmode	8042em	ga20/test	ISA ref	ref en	lbm proto	Deturbo

Index Bits Description

0A **DMA Clock selection.** Default = 00. Typical setting = 09h for 33.3 MHz SCLK, or 0Ah for 25 MHz SCLK.

3:0 Clock select for DMA controllers. Also determines 4045 internal BUSCLK for performance control (Index 08h) and IPC I/O wait states (Index 01h). The IPC may optionally further divide this clock by 2 before sending it to the DMA controllers (it should be programmed to do so via Index 01 bit 0). The source of the clock is SCLK, which is the 1X CPU clock, or 14.3 MHz. The resulting clock should be about 8MHz, which the IPC megacell will then divide by 2 (if so programmed via Index 01 bit 0) to get the 4 MHz used by the DMA controllers. Higher clock rates may be used at the risk of incompatibilities with the DMA devices. External BUSCLK for the ISA bus comes from the 4041 (Index 07h). 4045 internal BUSCLK doesn't need to be in phase with 4041 external BUSCLK, although the two BUSCLKs normally will be programmed to be the same frequency.

0000	SCLK/10	(80 MHz SCLK)
0001	SCLK/8	(66.67 MHz)
0010	SCLK/6	(50 MHz SCLK)
0011	(Reserved)	
0100	(Reserved)	
0101	(Reserved)	
0110	14.31818 MHz divided by 1.5 (9.545 MHz)	
0111	14.31818 MHz divided by 2 (7.159 MHz)	
1000	CLKIN/5	(40MHz SCLK)
1001	CLKIN/4	(33MHz SCLK)
1010	CLKIN/3	(25MHz SCLK)
1011	CLKIN/2.5	(20MHz SCLK)
1100	CLKIN/2	(16MHz SCLK)
1101	CLKIN/1.5	
1110	(Reserved)	
	1111	DMA clock is 14.31818MHz divided by 3.5 (4.091 MHz)

6:4 (Reserved)

7 External RTC

0	Internal RTC enable
1	External RTC. 32KX1 becomes the IRQ8# input.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
0A	dma clk	int RTC	-	-	-	dma clk3	dma clk2	dma clk1	dma clk0

Index Bits Description

0B **VL Arbitration and WBACK# control.** Default = xx00 0000 (the 'x's are set by the invert of the state of the DGNT# at power up. Typical setting = 00h for minimum system with WT CPU.

0 Arbitration lock. Used to prevent the CPU from going into HLDA while switching the CPU clock (required by Intel S-Series CPUs).
- 0 Normal operation
- 1 HOLD will not go high. If HOLD is already high, the arbitration logic will operate normally until HOLD goes back low.

1 New Write Back mode
- 0 4035 WBACK# mode (HOLD goes low when WBACK# is low)
- 1 4045 WBACK# mode (HOLD pulsed and A8:9, & A17:23 floated when WBACK# goes low).

2 LREQ1# input enable. Bit 3 must be a 1 for this function to work properly.
- 0 LREQ1# input disabled.
- 1 LREQ1# input enabled.

3 LREQ1# / LGNT1# Select
- 0 SLOW# and FLUSH# functions provided on pins 4 and 12
- 1 LREQ1# and LGNT1# functions provided on pins 4 and 12

4 LREQ2# input enable. Bit 6 must be a 1for this function to work properly.
- 0 LREQ2# input disabled.
- 1 LREQ2# input enabled.

5 IOCS# input enable. Bit 6 must be a 1 for this function to work properly.
- 0 IOCS# input disabled. Internal I/O is decoded from A0:9 only.
- 1 IOCS# input enabled. Internal I/O conditioned with IOCS# low.

6 SA17:19 function enable/disable. Default is the invert of DGNT# at power up. Can be changed by software. To enable the proper mode at power-up (probably essential for successful BIOS startup from ROM), the DGNT# pin needs a pull-up or pull-down resistor.
- 0 SA17:19 driven on pins 60, 58, and 57. (DGNT# pulled up.)
- 1 LREQ2# on pin 60, LGNT2# on pin 58, and IOCS# on pin 57. (DGNT# pulled down.)

7 SA17:19 indicator pin. This input bit is the INVERTED status of the DGNT# signal at power up. It cannot be changed by software. Bit 6 will also be set to this value at power up. Bit 6 actually determines the functions of the SA17-19 pins.
- 0 DGNT# was high. System configured for SA17:19 mode.
- 1 DGNT# was low. System configured for LREQ2# / LGNT2# and IOCS# mode.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
0B	arb control	sa default	sa function	iocs enab	lreq3# enab	slo/flush func	lreq1# enab	wb mode	arb lock

Index Bits Description

0C **Port 92 and RTC feature control.** Typical setting = 00h for external 8042, 30h for 4041 internal KBC.

0 Port 92 reset disable.
- 0 Port 92 bit 0 causes a CPU restart.
- 1 Port 92 bit 0 restart disabled. The 4041 will handle port 92 restarts by causing an SMI.

1 Port 92 Password protect enable.
- 0 Disable the Port 92 password protect feature. Setting this bit to a 0 will NOT re-enable accesses to the CMOS RAM password area if it has already been protected.
- 1 Enable the Port 92 password protect feature.

2 (Reserved)

3 CPU Reset Mode. Newer CPUs may need to be in HOLD before being reset.
- 0 Reset CPU only when the CPU is NOT in HOLD.
- 1 Reset the CPU only when the CPU is in HOLD.

4 CPU restart alternate code disable.
- 0 Link code 101 generates a CPU restart
- 1 Link code 101 does not generate a CPU restart. Must be set if this code is used for the mouse interrupt. When used with the 4041 it may always be set to a 1 (for the 4031 it should be a 0).

5 IRQ12 output enable.
- 0 Disabled.
- 1 The mouse interrupt flip flop drives IRQ12. It is open collector.

6 (Reserved)

7 IPC Core Reset Disable. Prevents loss of IPC register contents during 0V suspend (system powered down except for 4045; also known as "suspend to disk"). Software should set this bit to 1 before initiating a 0V suspend sequence, so that IPC registers will be preserved during resume. When resuming, software should write a 0 to this bit. This bit is cleared by PSRSTB# instead of PWRGOOD.
- 0 Reset the IPC core when PWRGOOD is low.
- 1 Do not reset the IPC core when PWRGOOD is low.

To guarantee IPC core reset for manufacturing test or other reasons, PWRGOOD and PSRSTB# must both be low at the same time. This condition will exist automatically if main power is off (PWRGOOD low) and the system battery is being changed or connected for the first time (PSRSTB# momentarily low). PSRSTB# is internally blocked (ignored) when PWRGOOD is high.

index	function	D7	D6	D5	D4	D3	D2	D1	D0
0C	RTC, pt92	-	-	IRQ12 out	link code	-	-	rtc password	pt92 res dis

4. System Level Functions and Cross References

This section describes each CHIPSet function from a system standpoint. Each subsection lists the basic features, board level implementation and options, and which chip(s) the function is contained in. The detailed operation is described in the individual chip specs, which follow this section. Extensive cross-references to the other sections have been provided to aid in locating all applicable information on a given topic.

4.1. Clocks

Either a 1x or 2x clock is provided to the 4041 CLKIN pin from an oscillator. The 2x clock is needed only for the highest speed DRAM mode (3-2-2-2 burst read), which is used at lower clock frequencies. The oscillator goes through the clock dividing and switching logic, which allows the power management system to slow down the system clock. This goes to three clock outputs, all produced at low skew with respect to each other:

CLK2OUT 2x clock, fed back into the 4041. Does not stop. This clock is 1x if a 1x clock is provided on CLKIN.
SCLKOUT 1x clock for everything except the CPU. Does not stop.
CPUCLK 1x clock to the CPU. May be stopped in stop-clock mode

These clocks are normally buffered with a 74F244, and distributed to the system, including being fed back into the 4041 for use by the rest of the internal logic.

Clock switching optionally may be done with a clock generator chip. Some clock generator chips switch frequencies slowly enough for the CPU VCO to remain locked. This allows power management with a standard 486 CPU.

For further information, see Sections 5.2 and 6.2.

4.2. Reset and GATEA20

The reset and GATEA20 logic is contained in the 4045.

The 4045 receives the reset (PWRGOOD) from the power supply or power up clear logic. It also receives CPU restart commands from the following sources: the 8042 (or emulation of the 8042) across the control link from the 4041; port 92, which is internal to the 4045; and config register reset. The 4045 generates SYSRESET and CPURESET. SYSRESET goes active only in response to PWRGOOD (due to power up or pushing a RESET button). CPURESET goes active for CPU soft restarts also.

CPU restarts may be redirected to generate an SMI rather than reset the CPU to avoid SMI and SRESET collision problems.

Table 4.1: CPU Chip Reset Signal Routing

CPU Type	SYSRESET signal	CPURESETsignal
Standard 486	-	RESET pin
CPU with SRESET input	RESET pin	SRESET pin

A20 is gated in the CPU only. The 4045 provides A20M#, which is the OR of the port 92 GATEA20 and the GATEA20 from the 4041, which is sent across the link. This consists of the 8042 GATEA20 (internal, emulated, or external) and the SMM GATEA20.

For further information, see Sections 5.3, 6.3, and 6.4. The Control Link is described in Section 4.3.1.

4.3. Arbitration

The 4045 contains most of the arbitration logic. It arbitrates between the CPU, local bus masters, DMA, and ISA masters. Refresh is always hidden, and occurs when either the CPU or local master has control of the bus.

Local bus masters are supported in accordance to the VL-Bus standard. The 4045 will preempt the local bus master off of the bus when an unmasked DMA request occurs. The 4035 SIPC provided one set of LREQ# / LGNT# pairs, allowing more to be created in an external PAL. This is the default mode for the 4045, but it may also be configured to provide 2 or 3 LREQ# / LGNT# pairs.

The SIPC arbitration logic may be locked to prevent the CPU from going into HOLD when switching CPU clock frequencies. S-series CPUs may not respond properly to snoop cycles or HOLD requests when the internal CPU VCO frequency has not yet stabilized following a change in the input clock frequency.

For further information, see Sections 5.4 and 6.5.

4.3.1 Control Link

As noted in the pin descriptions (Section 2), a simple control link is used for communication between the 4041 and 4045. 4041 LOUT is connected to 4045 LIN, and 4045 LOUT is connected to 4041 LIN.
The following events are communicated from 4045 to 4041:

Refresh Request (normal or ISA Master)
Refresh Complete
DMA Address Strobe

The following events are communicated from 4041 to 4045 using bit-serial event codes:

Refresh request acknowledge
Interrupt Acknowledge cycle
CPU Reset Request
Keyboard controller A20 Gate
Keyboard and mouse interrupts (if using internal keyboard/mouse controller)

4.4. Refresh

The AT-compatible Timer 1 in the 4045 generates refresh requests internally to the 4045. The 4045 arbitrates refresh requests with CPU reset requests, DMA and ISA Master requests (DREQ# inputs), and local bus master requests (LREQ# inputs). When the 4045 is ready for a refresh operation to proceed, it signals this to the 4041 via the control link (LIN and LOUT signals). The 4041 then arbitrates the refresh request with CPU activity (DRAM and/or ISA Bus accesses). When the 4041 is ready for ISA refresh to proceed, it acknowledges the refresh request via the control link back to the 4045. The 4041 performs the DRAM refresh cycle and the 4045 performs the ISA bus refresh cycle. When the 4045 has completed the ISA bus refresh, the 4045 signals this to the 4041 via the control link. The DRAM refresh may occur before, during, or after the ISA refresh.

During ISA bus refresh, the 4045 drives REFRESH#, MEMR# and SA0:7. DRAM refresh is always CAS-before-RAS (no external refresh address needed). In addition, refresh is always hidden, i.e., the CPU is allowed to continue running during refresh (HLDA inactive). As long as the CPU doesn't try to access local DRAM or the ISA Bus, there is no conflict. In particular, the CPU may continue to perform primary and/or secondary cache hit cycles and local bus slave accesses while a refresh operation is in progress. If the CPU tries to access local DRAM or the ISA Bus during a refresh operation, READY# and BRDY# are withheld and the CPU cycle is delayed as needed to allow the refresh to complete. The CPU can access local DRAM as soon as the DRAM refresh cycle is finished, even if the ISA refresh is still in progress (as will usually be the case).

For further information, see Sections 5.10.4 and 6.7.

4.5. Co-processor Logic

The co-processor logic is in the 4045 chip. It has pins for FERR# and IGNNE#. IRQ13 is generated internally. A write to I/O ports F0 or F1 clears the interrupt. The FERR# and IGNNE# pins may be converted to IRQ13 and INTCLR# respectively to allow external coprocessor error logic to be used.

For further information, see Section 6.12.

4.6. ISA Bus and ISA Features

CPU and local master accesses to the ISA bus are handled by the 4041 Chip. The 4045 is an ISA slave at that time. The 4045 contains the DMA controllers, and becomes the master for DMA cycles, and provides the arbitration for ISA masters. The 4041 is an ISA slave at that time, and converts ISA cycles to local bus cycles for local DRAM, cache, and local bus slaves.

For further information, see Sections 5.5.1, 5.11, 6.8, and 6.10.

Other ISA compatible features:

- Performance control using HOLD — see Section 6.6.
- Standard IPC functions and 4045 enhancements — see Section 6.9.
- 16-bit I/O decoding — see Section 6.11.
- Port 61h and Speaker output — see Sections 3.2, 3.4, and 6.14.

4.7. Local Bus Support

The CS4041 CHIPSet fully supports the VL-Bus 2.0. Both local bus slaves and masters are supported with very little external logic. Below is a brief description of the VL-Bus support. Refer to the individual chip specs for more details:

4041 Chip:

Samples LDEV# at the start of each bus cycle (either the end of the first or second T2) and allows a local bus slave to capture the cycle if active.

Translates DMA and ISA master cycles to local bus cycles when LDEV# is active.

Allows local masters to access most all of the system resources as if they are the CPU.

Performs snooping of the CPU L1 write back cache when enabled.

4045 Chip:

Provides 1, 2, or 3 sets of LREQ# and LGNT# signals for local bus masters.

Takes the CPU out of HOLD to write back a modified cache line.

For further information, see Sections 4.3, 5.5, 5.11, and 5.13.

4.8. DRAM controller

The DRAM controller supports 8 banks. This is configured as 4 blocks which may each contain 1 or 2 banks. This allows 4 double bank SIMMs to be installed. A "bank" is defined as a dword-wide physical memory controlled by a single RAS signal. The 4041 provides a total of 8 RAS lines, one for each bank. A single set of four CAS lines provides individual byte write control in each bank.

Each block has a programmable DRAM size, number of banks installed (1 or 2), and starting address. This allows maximum flexibility in DRAM installation, and does not require that banks be installed in any particular physical order.

The timing modes provided allow for timing optimization based on DRAM speed and CPU speed. A separate, less aggressive timing mode may be selected for VL masters since they are a less controlled (by the system designer) variable in the timing equation, and may not be as fast as the particular CPU used by the system designer. The timing modes are:

Burst Reads: 3-2-2-2, 4-3-3-3, or 5,4-4-4 timing modes
Single writes: 1 or 2 wait state.
Burst Writes: 3-2-2-2, or 4-3-3-3.
RAS to CAS timing: 2 or 3 T states (1.5 or 2.5 for 3-2-2-2 burst reads).
RAS pulse width for refresh cycles: 3 or 4 T states.
RAS precharge: 2 or 3 clocks

256K, 1M, 4M, and 16M deep DRAMs are supported. Direct drive is provided for two banks. Beyond that, buffering is dependent on the capacitive load provided by the DRAM configuration. 12/10 addressing is supported for 4Mx4 DRAM. 13/11 addressing is supported for 16M deep DRAMs by using a multifunction pin for MA12.

The DRAMs may be placed on the local data bus directly or buffered with 4 F245s. A buffer control signal is provided on a multifunction pin.

Throughout this document, the term "local DRAM" means DRAM controlled directly by the RAS-CAS signals of the 4041, not memory slaves residing on the VL bus and utilizing an LDEV# signal to claim memory cycles.

For further information, see Section 5.10.

4.9. Cache Controller

The cache controller has the following features:

Direct Mapped.
Standard SRAMs
External Tag RAM
Internal tag comparator
Operation up to 50MHz
16 byte line size

64K, 128K, 256K, 512K, and 1M cache size
Write back or write through
Single bank or dual bank (word interleaved) cache.
2-1-1-1, 2-2-2-2, or 3-2-2-2 reads. (2-2-2-2 mode for single bank only)
0ws or 1ws writes
2-1-1-1 or 3-2-2-2 burst writes.

The cache controller allows for cost performance tradeoffs. Full speed modes are supported for performance, and slower modes are supported for cost effective systems, especially with single bank caches and higher CPU speeds. Single bank 2-1-1-1 is supported.

A system board is easily upgradeable from a single bank to a dual bank cache without any jumpers on the data SRAMs. Some upgrade configurations may require a jumper on the tag RAMs since it connects directly to the CPU address bus and must match the cache size.

The BIOS may autoconfigure the cache, determining the size and configuration of the data RAMs and making sure the tag RAM size matches, as well as determining its width (which determines the cacheability range).

For further information, see Sections 5.5.2, 5.9, and 5.13.

4.10. Keyboard/Mouse Controller

An internal keyboard controller state machine is provided, which directly connects to the keyboard connector. It also includes a PS/2 style mouse port.

The mouse port or the entire keyboard controller may be disabled if special functions are required of an external 8042.

For further information, see especially Section 5.15. See also Sections 5.3.2, 5.13, and 6.13..

4.11. IDE Controller

Features:

Fast accesses to IDE drive data port
32 bit I/O cycles supported.
Up to 8 drives supported (up to 4 IDE addresses).
Programmable I/O read length, I/O write length, command inactive, and address setup time.
Each drive may select either of 2 programmed timing sets.
May coexist with other IDE controllers at different addresses, either local bus or ISA.
When enabled, it may disable an existing controller on the ISA bus.
Transparently supports the 3F7 register sharing with the floppy, wherever the floppy is located.

The 4041 IDE controller has a direct local bus interface to speed up access to the IDE drive. Drive accesses are speeded up by three means:

Eliminating the overhead of an ISA bus access.
Allowing 32 bit I/O cycles to reduce CPU overhead (converted to two 16 bit IDE cycles)
Shortening the command timing to the drive

For further information, see Section 5.12.

4.12. CPUs Containing Writeback Caches

The CS4041 CHIPSet fully supports 486-class CPUs that contain writeback caches, including those made by Intel, AMD, Cyrix, and other compatibles. There are several effects of a CPU write back cache, also known as "Level 1" or L1 WB:

- Memory write cycles often go only to the CPU internal WB cache and do not appear on the CPU local data bus. This provides a performance advantage over CPU write-through caches, in which all memory writes go to the CPU local data bus as well as to the cache (if the write is an L1 cache hit).

- Data in the CPU cache may be more up-to-date than the corresponding data in DRAM. Whenever a read from DRAM is cacheable by the CPU and is a CPU cache miss, the CPU must first write the existing L1 cache data back out to DRAM before filling the cache line with read data from a different DRAM location. For highest system performance, the CS4041 supports these writeback cycles by allowing them to occur as burst writes.

- When an ISA Master, Local Bus Master, or DMA controller performs a memory read, the CPU cache may contain the only valid copy of the data. The CPU "Hit Modified" signal, HITM# (or.equivalent), alerts the CS4041 CHIPSet that this is happening. The CHIPSet then responds by allowing the CPU to write the data out to DRAM and/or L2 cache before allowing the alternate master to complete the read operation.

For further information, see Sections 1.1, 5.7, 5.8, and 5.13.1.

4.13. CPU Capabilities for Power Management

4.13.1 CPU Clock Control

Some CPUs allow the CPU clock to be stopped completely, while others may require a minimum clock frequency as well as a maximum. Of the CPUs that allow the clock to be stopped, some allow it to be stopped instantly, while others require a handshake protocol to be followed. System power savings can often be achieved simply by slowing or stopping the CPU clock and the corresponding chipset clocking, even with CPUs that do not support a System Management Mode.

For further information, refer to Sections 1.1, 4.1 and 5.7.

4.13.2 System Management Mode (SMM)

Most newer 486-class CPUs support one of several possible types of System Management Mode (SMM), based on a system Management Interrupt (SMI) that puts the CPU into SMM. SMM, in turn, is recognized by the CS4041 chipset, which can make special DRAM areas available exclusively for use while running in SMM. SMM, in conjunction with power management software, allows system power consumption to be reduced without having to turn the system completely off or reboot the system every time full system operation is needed. CS4041 supports a variety of event detectors and timers for generating SMI#, both to signal the power management software to suspend appropriate parts of the system, and to wake up from a suspended state.

CS4041 also supports the I/O restart capability usually found in CPUs that have SMM. I/O restart means that when the CPU starts an I/O operation in an address range that has been previously programmed for I/O restart, the 4041 issues SMI instead of performing an I/O cycle on the ISA bus. The CPU then goes into SMM and provides information sufficient to allow the SMM routine to determine which I/O resource was being addressed. The SMM routine can then perform any peripheral power-up and initialization needed. On exit from SMM, the CPU will re-execute the I/O operation. I/O restart is useful in systems where peripherals can be selectively powered down transparently to the end user for power savings during system operation.

For further information, refer especially to Section 5.14. See also Sections 1.1, 5.3.1, 5.6, 5.7, 5.13, and 6.11.1. 0V Suspend (also known as suspend to disk) is discussed in Section 6.3.1.

5. 84041 Functional Description

5.1. DRAM/Cache/ISA Controller Chip

The 4041 controls the local DRAM, secondary cache, and ISA bus.

5.2. Clocks

5.2.1. Clock Generating Logic

The 84041 receives a 1x or 2x clock from an external oscillator on the CLKIN pin and generates the following clocks:

BUSCLK	ISA bus clock. Normally about 8MHz.
CLK2OUT	2x clock output. 1x when the CLKIN pin is driven from a 1x oscillator.
SCLKOUT	1x clock output. Used for the VL bus, 84045, and most of the 84041 logic.
CPUCLK	1x clock output for the CPU. May be stopped by power management logic.

CLKIN goes to a programmable speed reduction divider for slowing the system down. For full speed this divider is bypassed. The output of the speed reduction circuit is always passed out the CLK2OUT pin. SCLKOUT and CPUCLK are derived by either dividing by 2 when CLKIN is 2x or sending them out directly when CLKIN is 1x.

1X or 2X clock mode is signaled to the 4041 by a reset strapping option on the NMI pin. While SYSRESET is high, the NMI pin is an input to the 4041. An external pull-up or pull-down resistor determines the state of the NMI pin during this time. The pin state is sampled and latched on the falling edge of SYSRESET. If the pin is sampled high, the 4041 will operate in 1X clock mode. If NMI is sampled low, the 4041 will operate in 2X clock mode. The clock mode can be verified for test purposes by comparing CLKIN to SCLKOUT. In 1X mode, they will be the same frequency. The clock mode can also be checked via Index Register 38h, bit 7.

A 10K resistor should be sufficient for the NMI pull-up or pull-down, as long as there are no TTL loads on the NMI signal. Otherwise, a lower value pull-down may be needed for 2X clock mode. To assure proper detection of the clock mode, there is a short delay after the falling edge of SYSRESET before the NMI pin becomes driven as the NMI output signal for the CPU.

5.2.1.1. Speed Reduction Divider

The clock divider is used to provide the slow clock mode. The programming of the divider must not be done while the slow clock is enabled. The division choices are as follows, with the resulting frequency shown for typical full speed frequencies. It is programmed in register 8C, bits 6:4.

Table 5.1: Clock Divider

Mode	CLK Out	25MHz	33MHz	40MHz	50MHz
000	CLKIN	(25)	(33)	(40)	(50)
001	CLKIN/2	12.5	16.67	20	25
010	CLKIN/3	8.33	11.1	13.3	16.67
011	CLKIN/4	6.25	8.25	10	12.5
100	CLKIN/5	5	6.67	8	10
101	CLKIN/6	4.167	5.5	6.67	8.33
110	CLKIN/8	3.125	4.167	5	6.25
111	CLKIN/10	2.5	3.3	4	5

The power management hardware and/or software selects between the full speed and slow clocks.

5.2.2. Clock Inputs

The 84041 receives the following clocks:

CLKIN. Main clock input. Source for CLK2OUT, SCLKOUT, CPUCLK, and BUSCLK.
CLK2. 2x system clock. Used only for the 2x DRAM CAS state machine.
SCLK. 1x system clock. Used for all CPU related functions.
CWS#. Advanced 1x CPU clock. Used for the cache write enables.

5.3. Reset and GATEA20

The 4041 receives SYSRESET as an input from the SIPC chip. This input resets all registers and state machines to a known state. The clock divider circuit (the 2x to 1x circuit) operates while SYSRESET is active since the SIPC receives its clock from the 4041 and needs SCLK to generate the resets.

Both the CPURESET and A20M# (final GATEA20 to the CPU) pins are contained in the SIPC chip, but the 4041 chip provides information to the SIPC for both functions. This information is sent to the SIPC over the control link. The 4041 performs the following functions:

Detects a CPU Shutdown
Detects a CPU reset request from the internal or external 8042.
Optionally detects Port 92 bit 0 transitions to reset the CPU.
Optionally emulates the 8042 CPU reset request.
Detects a change on the GATEA20 signal from the internal or external 8042
Optionally emulates the 8042 GATEA20 signal, speeding up the operation.
Detects a 1 being written to Index 39h bit 6 (special CPU reset).

The keyboard GATEA20 and KBRESET come from one of three sources, as selected from the following table:

Table 5.2: GATEA20 & KBRESET Source.

	Internal KB Ctrl enable	Emulate GATEA20	Emulate KB RESET
GATEA20 Source			
External 8042	0	0	-
Internal Keyboard Controller	1	0	-
Emulated Fast GATEA20 logic	X	1	-
KB reset Source			
External 8042	0	-	0
Internal Keyboard Controller	1	-	0
Emulated Fast KB reset	X	-	1

5.3.1. CPU Reset and SMI

The Intel S series CPUs must not have an SRESET occur during SMM operation, or shortly following SMM operation. To avoid this, the 4041 may optionally redirect all CPU restart requests to cause an SMI rather than reset the CPU. (See Index 94h bit 4 and Index 0Ch bit 0.) System management mode may then perform the CPU reset function by jumping to the reset vector upon exiting SMM mode.

5.3.2. Emulated 8042 KBRESET# and GATEA20

The CPU reset and GATEA20 functions are normally done by the external 8042 or internal KBC. The CPU issues I/O commands to the 8042 to manipulate these signals. The 4041 chip may be configured to monitor these signals and simulate the RESET and GATEA20 functions of the 8042. Optionally, the commands may be blocked to the 8042 in order to speed up the operation.

The commands are written to the 8042 through address 64h. The commands which affect GateA20 and KBRESET# are:

AA Self Test. KBRESET# and GATEA20 go high.
D1 Write Output Port. Next byte written to port 60 is the data. Bit 0=KBRESET#, bit 1=GATEA20.
Fx Pulse output ports.
F0, F1, F4, F5, F8, F9, FC, and FD take GATEA20 high.
All of the even ones cause a KBRESET#

To take GATEA20 high a program could use either the D1 or Fx codes. (Pulsing GATEA20 will pulse it low, then high, leaving it high). To take GATEA20 low, Write Output Port must be used. The self test must be decoded simply because it does effect the pin. Programs do not use this as a method of changing it.

A CPU Reset requires the Fx command. Using the D1 command to do it will keep CPU reset low forever, recoverable only by hardware reset.

The Fx and D1 commands are optionally blocked from the 8042. The 4041 does this by not issuing an IOW# for them. F0, F1, F2, F4, F5, F6, F8, F9, FA, FC, FD, and FE commands are blocked. The D1 commands are a bit more complicated and follow these rules:

D1 commands are blocked.
Following a D1 command, the next write to port 60 is blocked. If another (non D1) command occurs before port 60 is written, the next port 60 write is not blocked.

> Following a blocked port 60 write, if an FF command (writing FF to port 64) occurs as the next write to the keyboard controller, it is blocked also. If multiple FF commands are issued only the first one is blocked.

This will block all three writes in the typical sequence, which is as follows:

Write the D1 command to port 64.
Write the data to port 60.
Write an FF command to the keyboard controller to detect when it has finished updating GATEA20.

The GATEA20 and KBRESET# pins of the 8042 may remain connected to the 4041 chip's GATEA20 input pin when the GATEA20 and KBRESET# functions are being emulated. They will continue to be ORed with the emulated versions.

5.4. Arbitration

The arbitration logic is contained in the SIPC. The 4041 chip receives the CPU HLDA, DGNT#, and MASTER# to determine what device is in control of the bus. The encoding is as follows:

Table 5.3: Bus Owner Indication

Bus Owner	HLDA	DGNT#	MASTER#
CPU Writeback during ISA Master access	0	0	0
CPU Writeback during DMA cycle	0	0	1
(Invalid)	0	1	0
CPU	0	1	1
ISA Bus Master	1	0	0
DMA controller	1	0	1
(Invalid)	1	1	0
Local Bus Master	1	1	1

The major functions controlled by the arbitration signals are as follows:

The 4041 chip does not generate parity for CPU writes to DRAM, but does for local masters, DMA and ISA masters (HLDA used to select).

The control link functions change (particularly the SIPC to 4041 direction) with DGNT#.

When DGNT# is high, the 4041 chip is a local bus slave and the ISA bus master. ISA bus cycles are generated in response to local bus activity.

When DGNT# is low, the 4041 chip is the local bus master and an ISA bus slave. Local bus cycles are generated in response to ISA bus activity. CPU writeback cycles cause the 4041 to become a slave on the CPU side, while holding the DMA controller or ISA Master in wait states.

When DGNT# is low and HLDA is high and MASTER# is high, the 4041 chip drives A10-16 with the contents of the DMA latch.

5.5. Address Mapping

5.5.1. I/O Addressing

All I/O accesses go to the ISA bus except:

VL-Bus claimed I/O accesses (LDEV# low)

Fast IDE controller I/O accesses (various addresses).

There are several fixed and several programmable I/O decodes.

The fixed decodes are for internal I/O including the IPC, configuration registers, and the fast IDE controller (when enabled).

5.5.1.1. Programmable I/O Decodes

There are two programmable I/O decodes. These may be used to provide a chip select or strobe to external logic or as an event detection to the power management logic. The I/O restart may also be enabled for these ranges. Each has a 16 bit starting address and an 8 bit mask, and the following attribute bits:

Force to VL bus (Forces LDEV# internally).
Generate chip select for I/O reads
Generate chip select for I/O writes
Generate chip select for ISA masters
Select asynchronous chip select, or read and/or write strobe (ANDed with I/O commands)

The address decode for A15 through A0 may be programmed. The upper and lower address bits may be masked in different ways to provide the decode size/range. For bits A6:A0 there is a three bit size selector which determines which address bits to decode. Within these bits the address must be contiguous. For bits A10:7 there are individual mask bits. This allows a decode to be active for non-contiguous spaces such as COM1 and COM2. Address bits A15:11 have a common mask bit. The programming is summarized in the following tables:

Table 5.4: I/O Decode Lower Bit Mask (A6:0)

Size Bits	Decode Size (bytes/ports)	Placement Boundary	Address Bits Decoded
000	1	any	A6:0
001	2	2 bytes	A6:1
010	4	4 bytes	A6:2
011	8	8 bytes	A6:3
100	16	16 bytes	A6:4
101	32	32 bytes	A6:5
110	64	64 bytes	A6:6
111	128	128 bytes	none

Table 5.5: I/O Decode Upper Bit Mask (A15:7)

Mask Bit	Repeats At	Address Bits Ignored
A7	128 ports	A7
A8	256 ports	A8
A9	512 ports	A9
A10	1024 ports	A10
A15:11	all upper.	A15:11

A multifunction pin may be programmed to provide the I/O chip select. The function of the chip select is selected by the 4 bits in the I/O decode's registers. The bits are as follows:

- **Activate for Reads**

- **Activate for Writes**

These bits select whether the pin will be active for I/O reads, I/O write, both, or neither.

Chip Select or Strobe

When Chip Select is selected, the decode is an asynchronous decode of the address and will be setup to and held after IOR# and IOW#. It acts differently depending on the current bus master:

CPU or VL master: M / IO# is decoded alone with W / R# (according to the read and write activation bits).

ISA master: The cycle type (memory or I/O, read or write) cannot be determined until the strobe goes active. If only one of the read and write activation bits are set, the output ANDed with it and will act the same as in the strobe mode. If both are active, it will be an asynchronous decode of the address (and will be active for memory cycles also).

When Strobe is selected, the decode is ANDed with the IOR#, and IOW# strobes (as determined by the read and write activation bits). This allows the pin to be connected to a latch gate or clock, or to a three state buffer enable.

• Disabled or Enabled During ISA Master Accesses

This bit allows the decode to be disabled during ISA bus master cycles.

Index CS0	Index CS1	D7	D6	D5	D4	D3	D2	D1	D0
28	2C	A7	A6	A5	A4	A3	A2	A1	A0
29	2D	A15	A14	A13	A12	A11	A10	A9	A8
2A	2E	mask15:11	mask10	mask9	mask8	mask7	size2	size1	size0
2B	2F	cs/stb	master	I/O writes	I/O reads	local	-	-	-

5.5.2. Memory Addressing

The default for memory cycles is the ISA bus, but most get claimed by the internal DRAM/Cache controller. Memory decodes in the 4041 include the following:

DRAM block decodes
640K-1M shadow bits
Programmable memory decodes
SMM memory decode.
Cache test window.

DRAM Block Decodes. These are explained more fully in the DRAM controller section. Each block has a starting address, DRAM size, and number of banks installed (1 or 2) bit. This is the same as the CS4021 and CS4031 (the CS4031 always had 1 bank per block).

640K to 1M Shadow RAM. Each of the following memory ranges has a separate shadow RAM bit for reads and writes. If the bit is a 0 the access goes to the ISA bus, ignoring the DRAM bank decodes. If the bit is a 1, the access goes to the DRAM (assuming there is DRAM present there, which there always is in a normal system). The D0000, E0000, and F0000 have separate shadow enable bits for user mode and SMM mode. This allows SMM memory to appear only in SMM mode.

C0000-C3FFF (VGA BIOS area)
C4000-C7FFF (VGA BIOS extension)
C8000-CBFFF (adapter board BIOS area)
CC000-CFFFF (adapter board BIOS area)
D0000-DFFFF (miscellaneous)
E0000-FFFFF (System BIOS extension)
F0000-FFFFF (System BIOS)

Table 5.6: DRAM Shadow Bit Encoding

DRAM Read Enable	DRAM Write Enable	Reads	Writes	Cache Writes	When Used
0	0	ISA	ISA	disabled	Memory on ISA bus
0	1	ISA	DRAM	enabled	Move ROM to RAM
1	0	DRAM	ISA	disabled	Shadow RAM, write protected.
1	1	DRAM	DRAM	enabled	Shadow RAM, read/write

The video areas each have a shadow ram enable bit. There are not separate bits for read and write. The address areas and bit definitions are listed below.

A0000-AFFFF (graphics mode area)
B0000-BFFFF (text mode area)

Flash ROM support under the Microsoft Chicago operating system requires that the shadow RAM be enabled on a 32KB boundary in the Exxxxh and/or Fxxxxh ranges. This is because Chicago only allocates a 64KB segment for the Plug-and-Play software, and programming the flash ROM requires accessing the ROM and shadow RAM (for the program) at the same time. To accomplish this, the 4041 has a separate set of shadow RAM bits (Index 1Dh) which are OR'd with the original set. They are available only for the Exxxxh and Fxxxxh ranges, but on a 32KB boundary.

To program the flash ROM, the appropriate flash shadow bits should be turned on in register 1Dh, followed by turning off the appropriate bits in registers 19h and 1Ah. (Doing this in reverse order may prevent CPU access to the program, causing a system crash or unrecoverable application error.) For additional information, refer to Index register 1Dh description.

High 1MB ROM. Memory reads from the 1MB range at the very top of the 4GB address space (FFFxxxxxh) always activate ROMCS# and MEMR# and cause data steering from XD-bus (for ROM), regardless of any shadow RAM or ROMCS or other bit settings. (ROM data on the XD bus is driven onto the ISA bus as well into the 4041.) This is a fixed decode, allowing the BIOS ROM to begin executing properly after any CPU reset. The first CPU instruction fetch following a CPU reset comes from FFFFFFF0h and normally is a far jump into the 000Fxxxxh area. Usually the ROM doesn't receive address bits higher than A17 or A18, so ROM accesses in the FFFFxxxxh range normally map to the same physical ROM space as accesses in the 000Fxxxxh range (when shadow RAM is disabled). Memory writes to the top 1MB range generate ROMCS# and MEMW#, even if no ROMCS enable bits are set as ones (Index 1Bh = 00h). Since the system BIOS normally is 64KB, the size of the high ROM area may be reduced to 64KB instead of 1MB in a future revision of the 4041.

256MB Address Space Limitation. The total memory address space accessible by the CPU's 32-bit address is 4GB (4096 MB, where M = 1024K and K = 1024). However, to make the most efficient use of available pin count on the 4041, CPU address bits A28, A29, and A30 do not connect to the 4041. A31 is connected to allow detection of coprocessor I/O accesses and is included in memory address decoding. Thus, the 4041 effectively divides the 4GB address space into two 2GB subdivisions, each of which is further divided into eight 256MB blocks. Since the ISA bus only receives A0-23, the ISA bus effectively is replicated at 16MB intervals throughout the 4GB address space, subject to cycle claiming by L2 cache, DRAM, VL slaves, or ROM.

Within the **lower 2GB** subdivision (A31 = 0), DRAM and L2 cache may occupy up to the full 256MB of each 256MB block. Any holes in DRAM or space left over from the top of DRAM to the top of 256MB may be occupied by ISA or VL memory. DRAM or ISA accesses above the first 256MB go to the same physical DRAM or ISA memory as accesses in the lowest 256MB, effectively making only the first 256MB useful in a system implementation. In fact, incoherency can arise between the CPU cache and external DRAM if the CPU cache contains data from different 256MB blocks, one of which is then written to.

All memory accesses to the **higher 2GB** (A31 = 1) go to the ISA bus or high ROM unless the access is claimed by a VL memory slave. As in the lower 2GB, the 4041 replicates address mapping at 256MB intervals.

The 256MB replication interval in the 4041 doesn't apply to VL memory slaves if they decode the full 32-bit address. VL memory potentially can reside uniquely anywhere in the entire 4GB space that doesn't overlap local DRAM. VL memory slaves can also claim ROM cycles, including accesses to the high 1MB area. Assertion of LDEV# during a ROM cycle doesn't necessarily prevent ROMCS# from being asserted, but it prevents MEMR# and MEMW# and gives the cycle to the VL slave.

5.5.2.1. Programmable Memory Decodes

Two programmable decodes provide attributes for specific memory ranges. The decodes have the following programmable bits:

Starting address (must be on a "size" boundary)
Size (64K, 128K, 256K, 512K, 1M, 2M, 4M, 8M, 16M, 32M, 64M)
Caching Status.
Hole in DRAM.
Force to VL bus

The starting address must be on a decode "size" boundary. For instance, if the decode is to be 1 megabyte, it must be placed on a 1meg boundary. The Size bits simply determine which address bits to ignore, according to the following table:

Table 5.7: Programmable Memory Decode Size and Placement

Size Bits	Decode Size	Placement Boundary	Address Bits Decoded
0000	64K	64K	A31, A27:16
0001	128K	128K	A31, A27:17
0010	256K	256K	A31, A27:18
0011	512K	512K	A31, A27:19
0100	1M	1M	A31, A27:20
0101	2M	2M	A31, A27:21
0110	4M	4M	A31, A27:22
0111	8M	8M	A31, A27:23
1000	16M	16M	A31, A27:24
1001	32M	32M	A31, A27:25
1010	64M	64M	A31, A27:26
1011	128M	128M	A31, A27

The attribute bits determine how the memory decodes are used. If all are 0 the decode has no effect. The bits control functions independently, and any combination of the bits may be used (although not all combinations are necessarily useful).

Hole in DRAM Bit

This bit disables the DRAM decode for the memory range, allowing a hole to be placed in DRAM. This would be used if an I/O card on the VL or ISA busses required a specific memory range, which would normally be in DRAM. If the programmed memory range does not hit the local DRAM, the bit will have no effect. While this function will work in the 640 to 1M range, there are specific bits for that range which are generally more versatile.

Putting a hole in DRAM will also make the range non-cacheable, since only local DRAM is cached. This has the effect of disabling cache line fills. It will not prevent cache hits if the memory range is already in the cache. If the hole is placed in DRAM after the cache has already been in use in WB mode, the cache contents should be flushed out by reading a cacheable memory range twice the size of the cache before setting this bit.

Force to VL-Bus Bit

This bit will force the access to the VL bus by forcing LDEV# low internally. The LDEV# pins will be ignored. Forcing LDEV# will only take the cycle away from the ISA bus, not the DRAM and cache controller. The Hole in DRAM bit should also be set to take the cycle away from the DRAM (which will also disable cache line fills).

Cache Attribute Bits

These bits set the cache attributes. Only local DRAM can be cached in the secondary cache, and normally also in the L1 cache. These bits allow a memory range to be forced to be non-cacheable or write through cacheable. If the range is not in DRAM it can also be forced to be cacheable in the L1 cache. Each configuration is described below:

00 Default mode. In this mode the decode has no effect on the cache, and the memory takes on the default cache mode, which is normally set to cache all of DRAM in either write back or write through mode.

01 Non-Cache. In this mode the programmed range is not cached in either the L1 or L2 cache.

10 Write through in L1. In this mode the programmed range is forced to write through in the L1 cache. The L2 cache is unaffected, and is cached according to the current L2 mode. If the CPU contains a write back cache, the WB / WT# pin of the CPU must be connected to a multifunction pin with this function selected in order to force it to write through. This mode will force any range to be cacheable in the L1 cache, even if it is on the ISA bus or VL-Bus.

11 Write Back Cache in L1. In this mode the programmed range is forced to write back in the L1 cache, even if the default mode that is programmed on the 4041 is write through. The CPU must be capable of write back operation, and this mode enabled in the CPU for this bit to have any effect. The WB / WT# pin of the CPU must also be connected to a multifunction pin with this function selected in order for this bit to have any effect. This bit does not affect the L2 cache in any way. This mode will force any range to be cacheable in the L1 cache in write back mode, even if it is on the ISA bus or VL-Bus. THIS IS VERY DANGEROUS, since the slave must be capable of being backed off during VL or ISA master/DMA snoops. This should only be done to ISA and VL memory if there is no chance that a VL master or ISA master or DMA will access the memory. The main use of this bit is to make only a specific section of main DRAM write back in the CPU.

Chip Select

The programmable decode may also provide an external chip select. This is done by selecting one of the multifunction pins to provide a chip select from one of the memory decodes. There is no enable for this function in the memory decode's registers, since the programming of the multifunction pin is sufficient.

dec0 Index	dec1 Index	D7	D6	D5	D4	D3	D2	D1	D0
30	34	A23	A22	A21	A20	A19	A18	A17	A16
31	35	A31	A30	A29	A28	A27	A26	A25	A24
32	36	-	VL local	hole	-	Size3	Size2	Size1	Size0
33	37	-	-	-	-	-	-	cache1	cache0

5.6. SMM Memory Support

5.6.1. 4041 SMM Memory Map Details

The 4041 places SMM memory anywhere in the D0000-FFFFF area, where the BIOSes reside. This has several advantages, including:

Allows cache to remain enabled, and no requirement to flush the CPU cache.
Main memory is all available at its normal address.
SMM memory may be (optionally) cached in L1 and/or L2

SMM code would be in the BIOS area. In most cases it would be in the E0000 area, but it could be in the F0000 or D0000 also. In the case of E0000 the area would probably be read-only in user space and read/write for SMM. In the case of D0000 it would be disabled in user space.

Using D0000 effectively gives SMM it's own address space. With memory managers such as QEMM, 386max, or the DOS 6 utilities, the C0000, D0000, and E0000 areas are checked for memory or peripherals. If nothing is found, these programs back fill the address space with extended memory using virtual-86 mode. Some of the area is used as the EMS page frame (if enabled) and the rest is used to load drivers high. Therefore, the addresses of D0000 or E0000 never appear on the CPU pins when in user mode after initialization, since the space is remapped. In SMM mode the paging is not used, and the actual address of D0000 or E0000 will come out of the CPU and access the shadow RAM. Effectively this looks like two separate sets of memory at D0000 and/or E0000.

The shadow RAM read and write registers are repeated for SMM space, and the alternate set is used on a cycle by cycle basis with SMIACT# or SMIADS# (depending on the CPU type)

The configuration register at index 1C provides the shadow DRAM mode while in SMM. It allows separate control of the D0000, E0000, and F0000 ranges. If all of the bits are set to 1s, the entire range from D0000-FFFFF will be enabled as read/write shadow RAM in SMM mode. In order to access these memory ranges on the ISA bus, the appropriate bits should be turned off. The following table shows how the shadow RAM bits interact:

Table 5.8: User/SMM Space Shadow RAM Bits

Memory Range	Cycle type	User Space bit	SMM Space bit
C0000-CFFFF	reads	Reg 19 bits 0:3	same as user space
C0000-CFFFF	writes	Reg 1A bits 0:3	same as user space
D0000-DFFFF	reads	Reg 19 bit 4	Reg 1C bit 0
D0000-DFFFF	writes	Reg 1A bit 4	Reg 1C bit 4
E0000-EFFFF	reads	Reg 19 bit 5	Reg 1C bit 1
E0000-EFFFF	writes	Reg 1A bit 5	Reg 1C bit 5
F0000-FFFFF	reads	Reg 19 bit 6	Reg 1C bit 2
F0000-FFFFF	writes	Reg 1A bit 6	Reg 1C bit 6

If shadow RAM is enabled in both user and SMM spaces, it is the same physical memory that appears in both modes. This allows the SMM code to be loaded from user space by enabling the shadow registers.

With the Intel CPUs the initial SMI will use the address of 38000 as the SMM base address. This is in the middle of the bottom 1meg of user space. The base address may be changed before the return is executed, allowing all future SMIs to vector to a new base. In the 4041, this initial SMI goes to user space. A dummy SMI should be generated, with code at 38000 to change the base address to the desired area in the D0000-FFFFF range.

5.6.2. Additional 4041 SMM Features

5.6.2.1. Flush CPU Cache on SMM Entry

The 4041 may optionally flush the CPU cache upon entry into SMM mode. It does this by driving FLUSH# low for 4 clock cycles when SMIACT# goes active (a high to low transition). This function is not supported (and is not needed) for CPUs that use SMIADS# instead of SMIACT#. This function is generally not required in any case.

The 4041 has no mechanism to flush the CPU cache upon exit from SMM mode, or to flush the secondary cache. No cache flushing should be needed in connection with SMM in any case if SMM space is allocated in shadow RAM as described in Section 5.6.1. Cache flushing considerations are discussed further in the next section.

5.6.2.2. SMM Caching and Cache Flushing

SMM memory may be optionally cached in the L1 cache. Index 94h bit 3 determines whether it is cached.

SMM memory cacheability in L2 cache can be controlled via Index 18h, bit 3, assuming SMM memory is in shadow RAM (as it normally will be). This bit controls L2 cacheability of the entire shadow RAM area from 0A0000h through 0FFFFFh, including the system BIOS area (assuming system BIOS is in shadow RAM, as it normally will be).

When deciding where to place SMM code and whether or not to make it cacheable, the potential need for cache flushing in connection with SMM should be considered. Flushing a writeback cache, in particular, can be extremely time consuming because of the potentially large number of "dirty" cache lines that might need to be written back to memory. L1 and L2 cache flushing in connection with SMM becomes necessary in two main cases (see also section 5.9.4):

(a) **Entering SMM.** If SMM address space is cacheable in user mode and is mapped to a different area of DRAM during SMM, then cache flushing is necessary upon entry into SMM to prevent unwanted cache hits during subsequent SMM accesses.

(b) **Exit from SMM.** If SMM address space is cacheable in SMM mode and is mapped to a different target area (e.g., DRAM or ISA bus) during user mode, then cache flushing is necessary upon exit from SMM to prevent unwanted cache hits during subsequent user mode accesses to the same address range.

No cache flushing is needed if the address range always maps to the same target memory area in both SMM and user mode, or if the address range is non-cacheable in both SMM and user mode.

Case (a) above should never arise in a CS4041 system if SMM code and data reside in shadow RAM as described in section 5.6.1. There is no way for the SMM address range to map to local DRAM (other than SMM space) during user mode. At most, the SMM address range might map to ISA bus memory or ROM during user mode. The SMM address range will automatically either be non-cacheable during user mode because it doesn't map to local DRAM, or will remain mapped to the SMM space.

Case (b) above can arise if the address range used for SMM maps to ISA bus memory or ROM during user mode. For example, 0Dxxxxh could be used for SMM space and also for user mode option ROMs. In that case, SMM space should be made non-cacheable in both L1 and L2 caches in order to avoid the need for cache flushing per case (b) above. SMM non-cacheability can be achieved by setting Index 18h bits 3:2 to '10' and setting Index 94h bit 3 to '1'. These settings allow other shadow RAM areas, such as system and video BIOSes, to remain cacheable in L1 cache.

Depending on system implementation objectives, 0Exxxxh might be a better choice for SMM code and data. During user mode, the 0Exxxxh area could remain mapped to SMM space (write protected) or be unused. Then case (b) above would be satisfied without having to make SMM space non-cacheable during SMM. SMM space could remain cacheable in both L1 and L2 caches.

It should be noted that using shadow RAM for SMM space does not prevent the user mode software from using the same address ranges for EMS windows. The window addresses used by software are remapped by the CPU paging mechanism prior to each L1/L2 cache or DRAM access, and thus do not interfere with the SMM address range during L1/L2 cache and DRAM accesses.

Some CPUs make SMM space non-cacheable automatically, which eliminates any need for the chipset to make SMM space non-cacheable in L1.

5.6.2.3. Force A20M# High During SMM

A20M# may be forced high during SMM cycles if desired (Index 94h, bit 2). This is not necessary to access SMM memory since it resides in an area where A20 is low anyway, but it is necessary to access user memory above 1M. When enabled, this feature will take SMIACT#, invert it, and OR it with the other sources of GATEA20 in the 4041 (keyboard GATEA20 and emulated keyboard GATEA20). This will send a SET GATEA20 and RESET GATEA20 code across the link to the 4045 upon entry and exit from SMI mode. If the internal GATEA20 was already high, no codes are sent. This feature is not available (and not necessary) for CPUs which use SMIADS# instead of SMIACT#.

5.6.2.4. Soft Reset Redirection

Section 5.3.1 describes CPU reset redirection to SMI. For additional information on reset in general, see section 4.2.

5.6.2.5. Port 26/27 Configuration Register Accesses

Since SMM mode may be entered at any time and cannot be disabled by normal user code, there are possibilities of splitting a port 22/23 access with an SMM call. If the SMM code accesses configuration registers (which is almost always will) this will interfere with the user mode port 22/23 access.

To solve this problem, a second window into the configuration register space is provided for SMM mode. Port 26 is the address port, 27 is the data port. There is a separate address storage register for port 26, and a separate "accessed bit" for ports 26/27. Accesses through ports 26/27 access the same configuration registers as ports 22/23. SMM code should always use ports 26/27. User code should always use ports 22/23.

5.6.2.6. Shadowing Port 70

I/O port 70 is the index pointer into the Real Time Clock/CMOS RAM, and is write only in the SIPC. To allow SMM code to access the RTC/CMOS RAM, port 70 is shadowed and readable through configuration register 96h.

To access the RTC/CMOS RAM, the SMM code should read index register 96h (through ports 26/27), access port 70 & 71 as necessary, then write the original value back to port 70.

5.7. CPUs Supported

CPUs with a 486 bus are supported, including those with write back caches and SMM support.

There is no configuration register where the CPU type is written. Instead, each feature which requires different operation in the 4041 for different CPUs is controlled separately. This is because of the wide variety of CPUs, and the expected evolution of CPUs in the future.

5.7.1. CPU SMM Differences

The major difference between the Intel and Cyrix SMM support is the handling of the SMIACT# / SMIADS# pin. For the Intel CPU it goes low before the first SMM bus cycle and stays low until after the end of the final bus cycle. It can be used as a static mode indicator. It will not transition during bus cycles and has plenty of setup time (many clock cycles). For the Cyrix CPU SMIADS# is generated instead of ADS#, and is on a bus cycle by bus cycle basis.

For Intel SMM A20M# must be taken high to access upper memory. This is done across the link in the CS4041 when SMIACT# goes low. It is returned to its original state when SMIACT# goes back high. KEN# is also sampled in SMM mode for the Intel CPUs. A Config bit (Index 94h bit 3) determines whether SMM space is cached. FLUSH# is also optionally generated upon SMM entry, but this is normally not required.

For the Cyrix CPU, A20M# is ignored in SMM mode, and assumed high. The CS4041 does not modify A20M# for Cyrix SMM. KEN# and FLUSH# are left unaffected also.

5.7.2. CPU Clock Differences

The major differences between the CPU clock support involves changing and stopping the clock. On some CPUs the clock cannot be changed at all, except under tight control (which is possible with some synthesizers). With other CPUs, a STPCLK# function allows the CPU clock to be changed or stopped. On still others the clock is static and may be changed at any time. The power control logic has options for clock control. See section 5.14 for details.

5.7.3. CPU L1 Cache Options

The cache protocol is basically the same for all CPUs. The L1 Write Back cache functions are a superset of the Write Through functions. There are some options to support subtle CPU differences:

WB / WT# pin — Multifunction pin allowing certain areas to be forced write through.

WriteProtect — Multifunction pin allowing a line to be write protected in the CPU cache.

Write Protect function — Options on how write protecting the BIOS is handled. It may be either non-cache in the CPU cache, or EADS# generated on CPU write cycles. If the WB WT# pin is used, write protected areas are always write through.

5.7.4. CPU Pin Connections

CLOCK. The 4041 provides a separate clock pin, CPUCLK, for the CPU which should have minimal skew relative to other clocks such as SCLK into the 4041 (i.e., use same buffer chip for clock buffering). CPUCLK is the same as SCLK except that CPUCLK can be stopped without affecting the other clock signals in the system. Also, some CPUs require that the clock not be running before VCC reaches a valid level. This must be accomplished externally by gating CLKIN with PWRGOOD, or by using PWRGOOD to disable the buffer that provides CPUCLK.

RESET. For non-SMM CPUs, where this is the only RESET pin, it should be connected to CPURESET from the 4045. For SMM CPUs, which have an SRESET or INIT pin, the RESET pin should be connected to SYSRESET.

SRESET or INIT. If this pin exists on the CPU, it should be connected to CPURESET. This reset will not clear the SMM base. On CPUs with a write back cache, the CPU handles it more like an interrupt to avoid corrupting the cache. The 4041 has the ability to redirect all CPU restart requests to SMM.

BS8#. Connect to a pull-up.

BS16#. Connect to a pull-up and to each VL bus slot.

BOFF#. Tie high.

AHOLD. Tie low.

EADS#. Connect to the 4041 and all VL bus slots. VL masters will drive this pin. The 4041 drives it during DMA and ISA Master cycles and for write protected areas.

A20M#. Connect to the 4045 A20M# pin.

FLUSH#. This signal may be provided by either the 4041 or the 4045, but generally is not required to be connected to either. The 4041 may optionally flush the CPU cache before entering SMM, but this is not required by the SMM memory scheme of the 4041. The 4045 may flush the cache as part of the Performance Control function, or the 4045 FLUSH# pin may be redefined as an additional LGNT# signal, which is probably a more common use. If FLUSH# is not connected to either the 4041 or 4045, it should be pulled high.

L1 Write back cache pins:

HITM#. CPU output which connects to the 4041.

WBACK#. 4041 output which connects to the 4045 and all VL slots.

WB / WT#. CPU input optionally used for making selected memory areas write-through instead of write back. The 4041 can provide this signal on a multifunction pin (GPB) which defaults to this function, and the 4041 can be programmed to make the programmable memory ranges (Index registers 30-37h) cacheable in L1 as WT only. If the 4041 pin is not used, a pull-up resistor should be provided to enable the WB mode for L1 cache.

CACHE#. CPU output, not used by the 4041 or 4045.

INV. CPU input indicating whether to invalidate the L1 cache or not when snooped. High = invalidate. May be connected to W / R# to invalidate only on writes, or tied high.

WPROT#. CPU input found on some non-Intel CPUs. The 4041 optionally provides this signal on a multifunction pin, which write protects the shadow RAM as specified in the configuration registers. If not used, WPROT# may float.

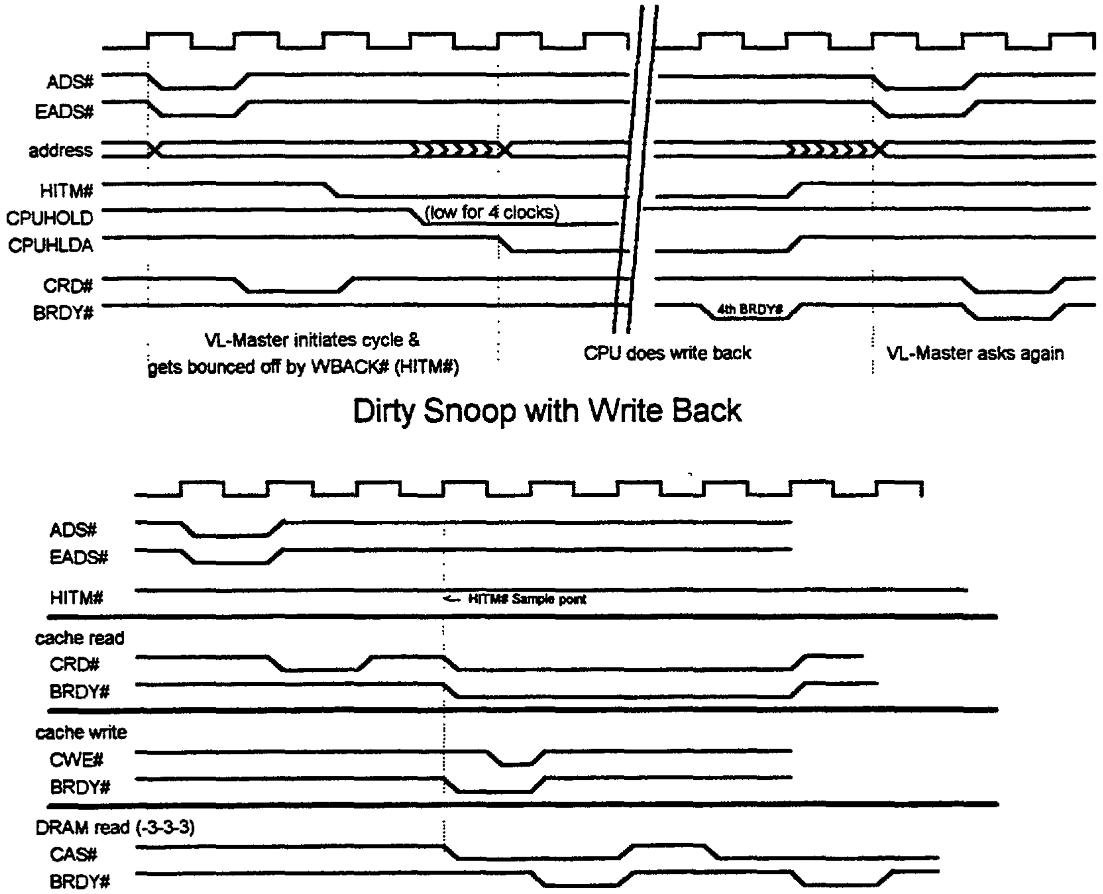
BLEN#. Used on the upgrade socket. Leave floating or pull high.

EWBE#. Used on the upgrade socket. External Write Buffer Empty input. Leave floating or pull high.

5.8. CPU Write Back Cache Snooping

When a CPU with an internal write back cache is used, that cache must be snooped ("peaked into") for local memory cycles which are initiated by any master other than the CPU itself. This includes cycles initiated by VL-Bus masters and cycles initiated by the 4041 on behalf of ISA masters and DMA. On a read, the CPU may contain the only valid copy of the data and must be allowed to write the data back out to secondary cache or local DRAM. On a write, the previous data may have been cached in the CPU and must be invalidated.

When a master other than the CPU owns the bus, an EADS# will be generated along with the ADS# for all memory read and write cycles. The EADS# is generated by the initiator of the local bus cycle (the VL-Master or the 4041). HITM# is generated by the CPU as a result of the snoop. It is driven at the end of the first T2, and may be sampled at the end of the 2nd T2. The 4041 may sample HITM# either at the end of the second T2 or at the end of the third T2 (to accommodate different CPU delays). If it is low, the CPU contains dirty data within that cache line. The current master is then removed from the bus, the CPU is taken out of HOLD, and allowed to write back the dirty data into the cache and/or DRAM. This will occur as a standard burst write, and is handled by the DRAM and cache controller the same way that burst writes normally are. The CPU is put back in HOLD when the write back is finished, and the current master reinitiates the cycle. Using the WBACK# output, the 4041 determines when to allow a writeback operation to begin after receiving HITM#.



Dirty Snoop with Write Back

Clean Snoop

Figure 5.1 L1 Write Back Cache Snoop

The "L1 write back" configuration bit determines whether the DRAM and cache controller wait for the HITM# signal when a master other than the CPU has the bus.

5.9. Secondary Cache Controller

The secondary (L2) cache controller has the following features:

Direct Mapped.
Standard SRAMs
External Tag RAM
Internal tag comparator
Operation up to 50MHz
16 byte line size

64K, 128K, 256K, 512K, and 1M cache size
Write back or write through
Single bank or dual bank (word interleaved) cache.
2-1-1-1, 2-2-2-2, or 3-2-2-2 reads. (2-2-2-2 mode for single bank only)
0ws or 1ws writes
2-1-1-1 or 3-2-2-2 burst writes.

The cache consists of "cache lines," each of which contains 16 data bytes and a tag address indicating what general area in DRAM the data bytes came from. As an example, a 256KB cache contains 16K cache lines (K=1024).

CPU address bits map onto the cache as follows:
A0-A1 (actually BE0-3 on the bus) address a specific byte within a dword.
A2-A3 select a specific dword within a cache line. Each cache line has 4 dwords.
A4 through An select a specific cache line, where "n" depends on the cache size. Table 6.8.1 lists the highest CPU address bit used for cache line selection ("An").
A(n+1) through Am are stored in the tag field of the cache line during a line fill, and subsequently compared with the tag field to determine cache hit or miss. "m" depends on tag size as well as total cache size. Table 6.8.1 lists the highest CPU address bit used for tag addresses. This, in turn, determines the maximum cacheable range, as shown in Table 6.8.2. Tag bit 0 is used as the "dirty bit" for write-back cache mode and is not used for write-through mode. This means that only 7, 8, or 10 CPU address bits are used for 8, 9,'or 11-bit tag sizes, respectively.

Secondary cache may be either single bank or dual bank. Dual-bank cache provides a performance advantage because the two banks can be interleaved on dword boundaries (CPU address bit A2 determines bank selection). Single-bank cache may offer a better balance between system cost and cache size for lower cost or entry level systems.

Secondary cache may be either write through (WT) or write back (WB), although WB mode normally will provide the best system throughput. In WT mode, DRAM writes that are cache hits go simultaneously to both the cache and the DRAM. In WB mode, such writes go only to the cache and thus take less time to execute. The updated data remains in the cache until a subsequent line fill is needed due to a cache read miss, at which time the "dirty" cache line is written back out to DRAM before the new cache line is filled. In either mode, line fills occur only on read miss cycles, never on writes (i.e., the 4041 does not "allocate on write"). The process of writing a "dirty" cache line back out to DRAM is also referred to as a "castout" operation and is performed as a four-dword burst write. The CPU or alternate master is held in wait states while the castout is occurring.

Table 5.8.1: CPU Address Bit Usage

	64KB	128KB	256KB	512KB	1MB
Highest line select bit	A15	A16	A17	A18	A19
Highest tag bit, 8-bit tag	A22	A23	A24	A25	A26
Highest tag bit, 9-bit tag	A23	A24	A25	A26	A27
Highest tag bit, 11-bit tag	A25	A26	A27	A27*	A27*

* To save pins, the 4041 doesn't connect to CPU address bits A28-A30.

Table 5.8.2: Cacheable Range

Highest Tag Bit	Maximum Cacheable Range
A22	8 MB
A23	16 MB
A24	32 MB
A25	64 MB
A26	128 MB
A27	256 MB

5.9.1. Pin Usage

The pin utilization is such that no jumpers are required on a board which offers all combinations of cache size, interleave and non-interleave, and tag width. The exception to this is that the proper address bits must be provided to the tag RAM based on the cache size. Since the unbuffered CPU address must go to the tag RAM for speed reasons, it is not possible for the CHIPSet to do the switching. Unused tag bits must be pulled up because they are compared to 1 during the cache hit-miss decision process. This allows faster decision time than would be possible with added internal logic to ignore the unused tag bits. Separate pull-ups should be used because software may drive the pins driven to different values during tag testing, so the pins should not be shorted together.

Table 5.9: Cache Pin Usage

SRAM Pin	Single Bank mode	Interleaved Bank mode
CA2	Cache RAM A2	Highest Address (A15, 16, 17, 18, or 19)
CA3A	Cache RAM A3	Even bank Cache RAM A3
CA3B	unused	Odd bank Cache RAM A3
CRDA#	Cache RAM OE#	Even bank Cache RAM OE#
CRDB#	unused	Odd bank Cache RAM OE#
CWEA#	Cache RAM WE#	Even bank Cache RAM WE#
CWEB#	unused	Odd bank Cache RAM WE#
CCS#	Cache Chip Select	Cache Chip Select (both banks)

Pin	Write Back	Write Through
TAGWE#	Tag RAM write enable	Tag RAM write enable
TAG1-10	Tag	Tag
TAG0	Dirty	unused

Table 5.10: Tag Bit Mapping

	64K cache			128K cache			256K cache			512K cache			1M cache		
Tag Width -->	8	9	11	8	9	11	8	9	11	8	9	11	8	9	11
Cacheability	8M	16M	64M	16M	32M	128M	32M	64M	256M	64M	128M	256M	128M	256M	256M
TAG0	dirty	dirty	dirty	dirty	dirty	dirty	dirty	dirty	dirty	dirty	dirty	dirty	dirty	dirty	dirty
TAG1	A17	A17	A17	A17	A17	A17	**A24**	**A25**	**A27**	**A24**	**A25**	**A27**	**A24**	**A25**	**A27**
TAG2	A18	A18	A18	A18	A18	A18	A18	A18	A18	**A25**	**A26**	-	**A25**	**A26**	-
TAG3	A19	A19	A19	A19	A19	A19	A19	A19	A19	A19	A19	A19	**A26**	**A27**	-
TAG4	A20	A20	A20	A20	A20	A20	A20	A20	A20	A20	A20	A20	A20	A20	A20
TAG5	A21	A21	A21	A21	A21	A21	A21	A21	A21	A21	A21	A21	A21	A21	A21
TAG6	A22	A22	A22	A22	A22	A22	A22	A22	A22	A22	A22	A22	A22	A22	A22
TAG7	**A16**	A23	A23	A23	A23	A23	A23	A23	A23	A23	A23	A23	A23	A23	A23
TAG8	-	**A16**	A24	-	A24	A24	-	A24	A24	-	A24	A24	-	A24	A24
TAG9	-	-	A25	-	-	A25	-	-	A25	-	-	A25	-	-	A25
TAG10	-	-	**A16**	-	-	A26	-	-	A26	-	-	A26	-	-	A26
Compare to 0	23-31	24-31	26-31	24-31	25-31	27-31	25-31	26-31	28-31	26-31	27-31	28-31	27-31	28-31	28-31

5.9.2. External Connections

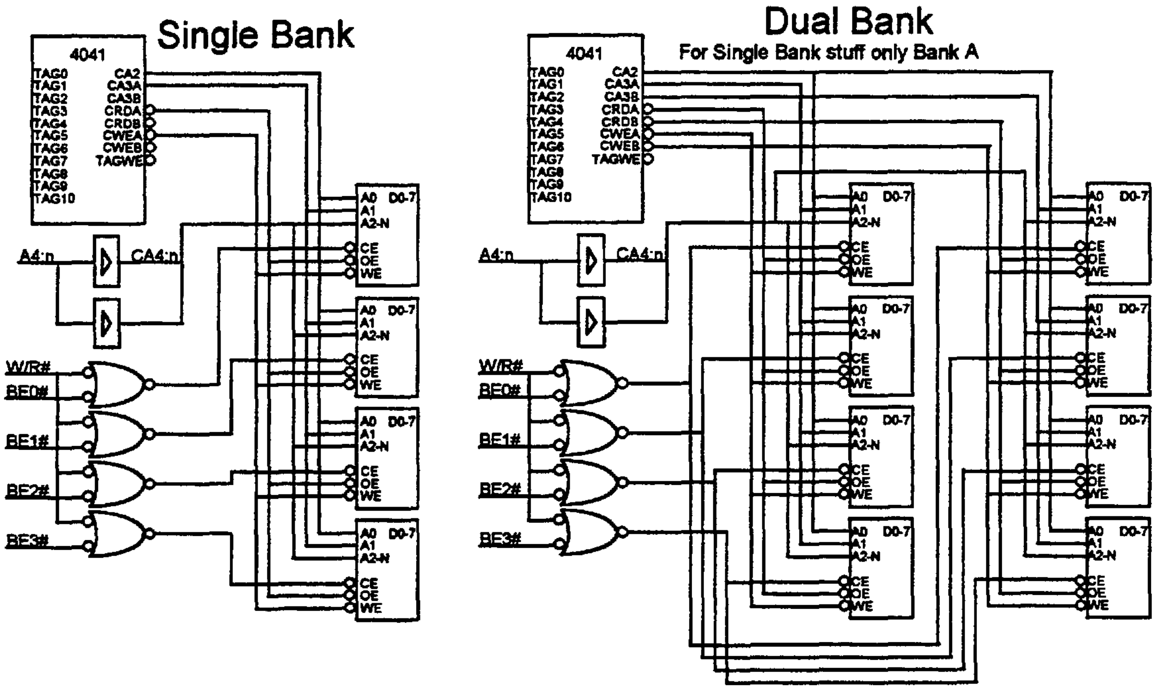
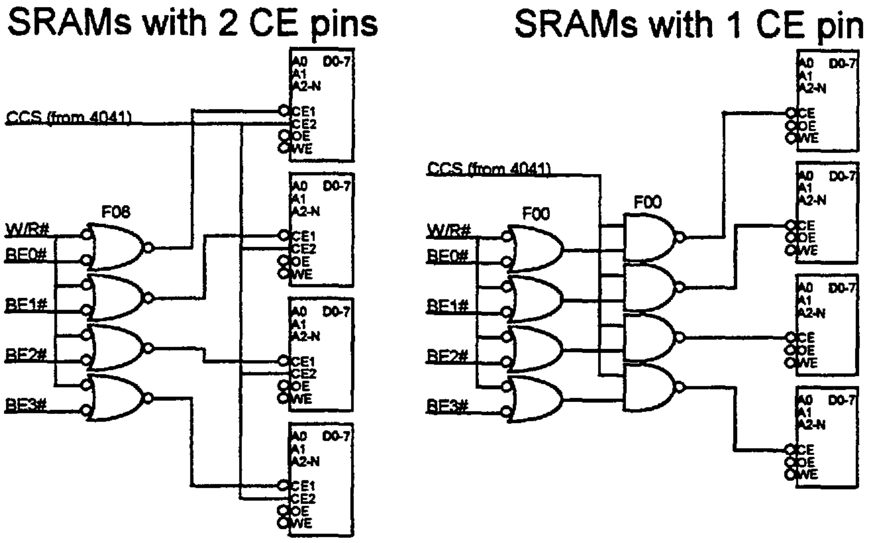


Figure 5.2: Cache Data SRAM Connections



CCS is a multifunction pin which may be either active high or active low.

Figure 5.3: Cache RAM Low Power Option Connections

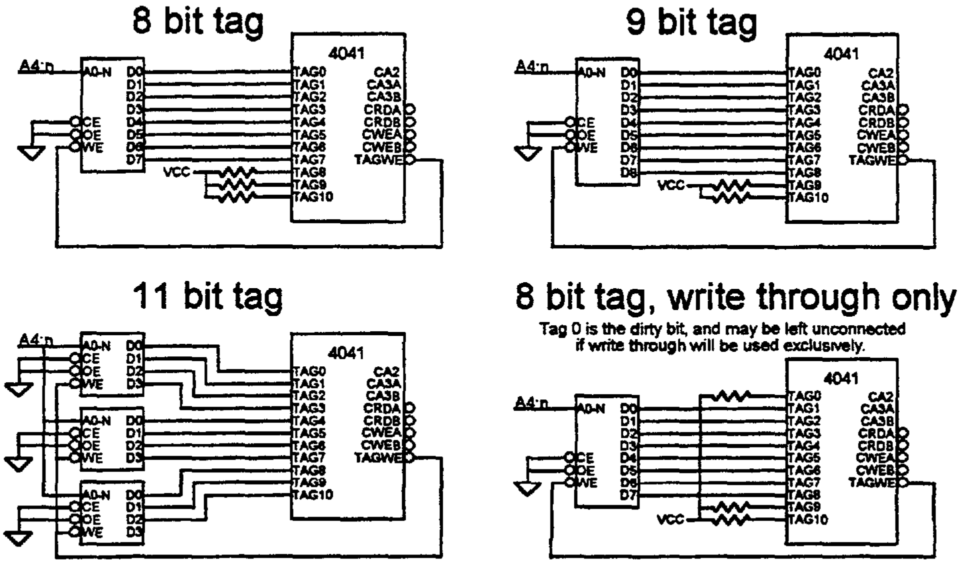


Figure 5.4: Cache Tag RAM Connections

5.9.3. SRAM Requirements

The 4041 cache controller uses standard off the shelf SRAMs for both tag and data.

Either 1 or two banks of **Data RAMs** are supported. If two banks are installed they are word interleaved, which puts less timing constraints on the RAM speed (allowing slower RAMs to be used, or a faster cache mode to be used).

For a single bank of data RAMs, CA2 and CA3 are supplied by the 4041 and are switched rapidly during a burst. These address bits are the most time critical for doing a -1-1-1 burst with a single bank.

When two banks of SRAMs are installed, A2 is encoded on the CRD# and CWE# signals, and each bank is supplied with a separate CA3 signal, which has staggered timing to allow maximum SRAM access time. (CS4021 operates this way, also.)

Table 5.11: Data SRAM Configurations

	Data RAMs, single bank		Data RAMs (dual bank)	
Cache Size	Qty	Type	Qty	Type
64K bytes	8	16Kx4	**8**	**8Kx8**
128K bytes	**4**	**32Kx8**	16	8Kx8 or 16Kx4
256K bytes	8	64Kx4	**8**	**32Kx8**
512K bytes	**4**	**128Kx8**	16	64Kx4
1M	8	256Kx4	**8**	**128Kx8**

Configurations in **BOLD** are the most common

Table 5.12: TAG SRAM Configurations

		8 bit Tag			9 bit tag			11 bit Tag		
Cache Size	Tag Depth	Qty	Type	cacheability	Qty	Type	cacheability	Qty	Type	cacheability
64K bytes	4K	2	4Kx4	8M	-	-	16M	3	4Kx4	64M
128K bytes	8K	1	8Kx8	16M	1	8Kx9	32M	2	8Kx8	128M
256K bytes	16K	2	16Kx4	32M	-	-	64M	3	16Kx4	256M
512K bytes	32K	1	32Kx8	64M	1	32Kx9	128M	2	32Kx8	512M *
1M byte	64k	2	64kx4	128M	-	-	-	3	64Kx4	1G *

* Maximum addressability of the 4041 is 256M for local DRAM.
64K, 256K, & 1M 9 bit tag is blank because there are no common 9 bit RAMs in this size. They are valid options.
x8 & x9 RAMs may be used for 64K, 256K, and 1M caches, but half of the chip (1 address line) is not used.
Unused tag bits MUST BE PULLED UP. 10K is sufficient.

The following table is a guideline for what SRAM speeds are required at various frequencies and timing modes. Additional tradeoffs of performance vs. SRAM speed are also possible.

Table 5.13: Suggested Cache Timing Modes and RAM Speeds

Frequency	# banks	Read Timing	Write Timing	Data RAM	Tag RAM	Comments
33MHz	Single	2-1-1-1	2-1-1-1	20nS	20nS	read timing is tight
	Single	2-2-2-2	2-1-1-1	25nS	20nS	read timing is loose
	Dual	2-1-1-1	2-1-1-1	25nS	25nS	
40MHz	Single	2-2-2-2	3-2-2-2	25nS	20nS	
	Single	3-2-2-2	3-2-2-2	20nS	25nS	
	Dual	2-1-1-1	2-1-1-1	20nS	15nS	
	Dual	2-1-1-1	3-2-2-2	20nS	20nS	
50MHz	Single	3-2-2-2	3-2-2-2	25nS	20nS	
	Dual	2-1-1-1	3-2-2-2	15nS	12nS	
	Dual	3-2-2-2	3-2-2-2	25nS	20nS	

The tag RAM may be 8, 9, or 11 bits wide. The tag width affects only the cacheable address range of the DRAM in the secondary cache. A wider tag is required for a small cache to achieve the same cacheable area. Cacheability is a cost/feature issue which is determined by the customer.

5.9.4. Cacheability

Only local DRAM is cacheable in either the 486 or secondary cache. Shadow RAM (RAM at 640-1M when present) is optionally cached. The programmable memory decodes may also make areas non-cached. If the total local DRAM size exceeds the cacheable range for the selected L2 cache size and tag width, DRAM above the cacheable range is automatically non-cacheable in L2 (high-order CPU address bits must be zero to get a cache hit or line fill). The entire DRAM is still cacheable in CPU cache (L1).

The 4041 controls L1 cacheability by driving KEN# low (cacheable) or leaving it high (non-cacheable) as needed. (The default is high.) The following Index bits affect KEN# usage and L1 cacheability:

Index 18h, bit 2 (shadow RAM) and bit 7 (write protection).
Indexes 33h and 37h, bits 0 and 1 (memory decodes).
Index 94h, bit 3 (SMM).

Other bits can affect cacheability indirectly by enabling or disabling DRAM. As noted above, local DRAM is usually always cacheable in both L1 and L2 unless specifically made non-cacheable, and memory accesses outside local DRAM are never cacheable in L2 but might be cacheable in L1 if specifically enabled using Indexes 33h or 37h (see Section 5.5.2.1).

Note that KEN# controls the L1 cache line fill decision, NOT the L1 cache hit/miss decision. Cacheability is controlled by not allowing line fills. Once information has been put into the cache and not invalidated by flush or EADS#, hits and writebacks can occur regardless of whether or not the memory area of origin may have been changed to non-cacheable status after the information was cached.

Likewise, L2 cacheability is controlled in the line fill decision, NOT in the cache hit/miss decision. (Write protected shadow RAM results in L2 cache writes being inhibited during attempts to write to the protected area.) As with L1 cache, information already in the L2 cache can result in cache hits and writebacks even if the memory area of origin has been changed to non-cacheable status.

The following Index bits affect L2 cacheability apart from other bits that enable or disable local DRAM:

Index 18h bit 3 (shadow RAM).
Index 22h, bits 7 and 6 (see below).

Index 22h bits 7:6 determine whether the cache is an instruction cache, a data cache, or both. The standard configuration is caching both. The other options allow performance tests to be performed. Restricting the cache to code or data affects only cache line fills, which occur only during read misses. For instance, if the cache is set for data only and a read miss occurs during a code fetch, it is treated as non-cacheable and the cache line is not replaced. If the existing line contains dirty data, it is not written back, since it is not being replaced.

Read and write hits are not affected by these bits, since they must continue to occur for proper cache coherency. For instance: The cache is set for data only and write back. A line fill occurs on a data read, followed by a write to the line, causing the cache line to become dirty. If a subsequent code fetch occurs to this same location, it must be read from the cache since that is the only valid copy. The same memory areas will be used for both code and data (Code generally gets loaded into the memory from which it is executed with prior Data write cycles).

The following table lists when line fills occur for each setting of the code/data configuration bits. Line fills only occur on memory reads to cacheable areas.

Index 22h Config Bits	Cache Type	Line fills occur when:
00	Unified	All memory reads
01	Instruction Cache	D/C# = 0
10	Data Cache	D/C# = 1

The code/data configuration bits may be changed on the fly without affecting cache coherency. These bits do not affect caching in the L1 (CPU) cache. Best system performance normally should result from caching both instructions and data.

5.9.5. Write Protection

Shadow RAM may be write protected, which causes a problem in the 486 cache. It is not possible to write protect the 486 cache. Several situations which occur in ATs require shadowed BIOSes to be write protected, such as:

1. Some programs search for ROMs by writing a location and reading it back to see if the data changed. Write protection properly simulates a ROM in this case.

2. Some adapter boards (very few) have write only registers memory mapped in the BIOS areas.

3. Some programs do a floppy disk verify (checking the CRC) by aiming the DMA address at the ROM. If it's not write protected, it is overwritten.

Items 1 and 2 pose a problem with the 486 cache. #3 is not a problem since the CPU is not doing the writing (an invalidate will occur to the 486 cache). There are several solutions, with various degrees of success and performance.

Solution A: Don't write protect the BIOS.

Advantages:	Highest performance.
Disadvantages:	Does NOT handle the 3 problems above.
4041 CHIPSet	Supported. Will not write protect in DRAM or either cache.

Solution B: Cache only in secondary cache. Still burst to the 486.

Advantages:	It works 100%.
Disadvantages:	BIOS runs slower.
4041 CHIPSet	Not Supported.

Solution C: Drive EADS# on writes to write protected areas:

Advantage:	High performance (almost as fast as Solution A).
Disadvantages:	Not a 100% solution. A write/read combination could occur before the EADS#. In practice this has not been a problem.
4041 CHIPSet	Supported. Recommended solution.

On CPUs with a write back cache Solution C causes a problem if the BIOS is cached in write back mode. The writes will never be seen on the bus, so the EADS# will not be generated. To get around this, the WB / WT# pin (a multifunction pin) goes to the write through mode for write protected areas during cache line fills.

5.9.5.1. Changing a DRAM Write Protect Bit

Changing a DRAM area from write enabled to write disabled can be a problem for an L1 writeback cache. There is no problem with L2 cache (WB or WT) or with L1 cache if it is write-through. Index 18h bit 6 can be set to '1' to force all writes to shadow RAM to be write-thrrough in L1 cache.

With writeback caches, memory reads and writes can result in updated data being loaded into the cache while the memory area of origin is enabled for writing. Updated data may still be present in the cache at the time that the memory area is changed to write disabled. Sometime later, the updated data may need to be written back out to memory after the memory (shadow RAM) has become write disabled. This is not a problem for L2 cache because the writeback operation is initiated and controlled by the 4041, which performs the L2 writeback regardless of whether or not the DRAM has become write protected. DRAM write protection prevents line fills, cache writes, and normal DRAM writes, but does not prevent L2 writeback operations when needed.

For L1 cache in the CPU, however, there is no way for the 4041 to know that a CPU writeback operation is occurring, since the CPU initiates and controls the cycle. The writeback will fail if the memory area has become write protected. To avoid this problem, the system should set Index 18h bit 6 to '1' to force writes to shadow RAM to be write-through rather than write-back in L1 cache. Making L1 cache write-through causes all writes to appear on the CPU pins, allowing the 4041 to perform the DRAM write while the DRAM is still write enabled, or to assert EADS# (Index 18h bit 7 set to '1') if the DRAM has become write protected.

5.9.6. Cache RAM Power-Down

Many cache RAMs go into a power-down mode when their chip selects are inactive. Since cache RAMs consume a significant amount of power, it may be advantageous to power them down between bus cycles, since the 486 CPU only uses the bus about 50% of the time (DX2 uses the bus a higher percentage of the time).

The cache controller provides a chip select pin for this function, which may be optionally used. When cache RAMs with only 1 chip select are used, the chip select must be gated with the byte enable and W/R# logic.

The chip select goes low asynchronously with ADS#, and stays low until the cache controller no longer requires use the cache RAMs. For a cycle where the cache is involved (a hit or line fill) the chip select will go high at the end of the final T state. For ISA bus or VL-Bus cycles it will go high when the cache determines that the cycle is a miss and is non-cacheable. For back to back cache cycles the chip select will be high for only a short time.

5.9.7. Cache Coherency

Coherency must be maintained in the primary and secondary caches when VL masters, ISA masters, and DMA cycles occur. Cache Coherency is handled in several ways:

Secondary cache: Coherency is maintained by always writing the data in the cache on write hits, regardless of the current master. There are no valid bits in the tag, so every cache location must be valid at all times after initialization. Coherency on reads is maintained by always reading the data out of the cache on read hits, to prevent stale data from being read from the DRAM. This is required because the secondary cache is write back.

486 Primary cache. Coherency is maintained by driving EADS# low for all local master, ISA master, and DMA memory writes (VL-Masters are responsible for driving it when they own the bus). This invalidates the cache line in the 486. The address is already provided on the local bus. A24:31 are driven low when ISA masters and the DMA controller have the bus (A28-30 have pull downs at the 486 since they do not go to the 4041).

486 Primary Write Back cache: Any access to DRAM must snoop the primary cache before accessing either the secondary cache or the DRAM. This is done by driving EADS# low and waiting for the HITM# signal before returning any READYs from the DRAM or cache controller. If dirty data is found in the cache, the current local bus cycle is backed off (by removing either the VL-master or the 4041 itself from the bus) and the CPU is taken out of hold and allowed to write back. The local bus cycle is then reissued. The line in the CPU is invalidated.

5.9.8. Cache Operation

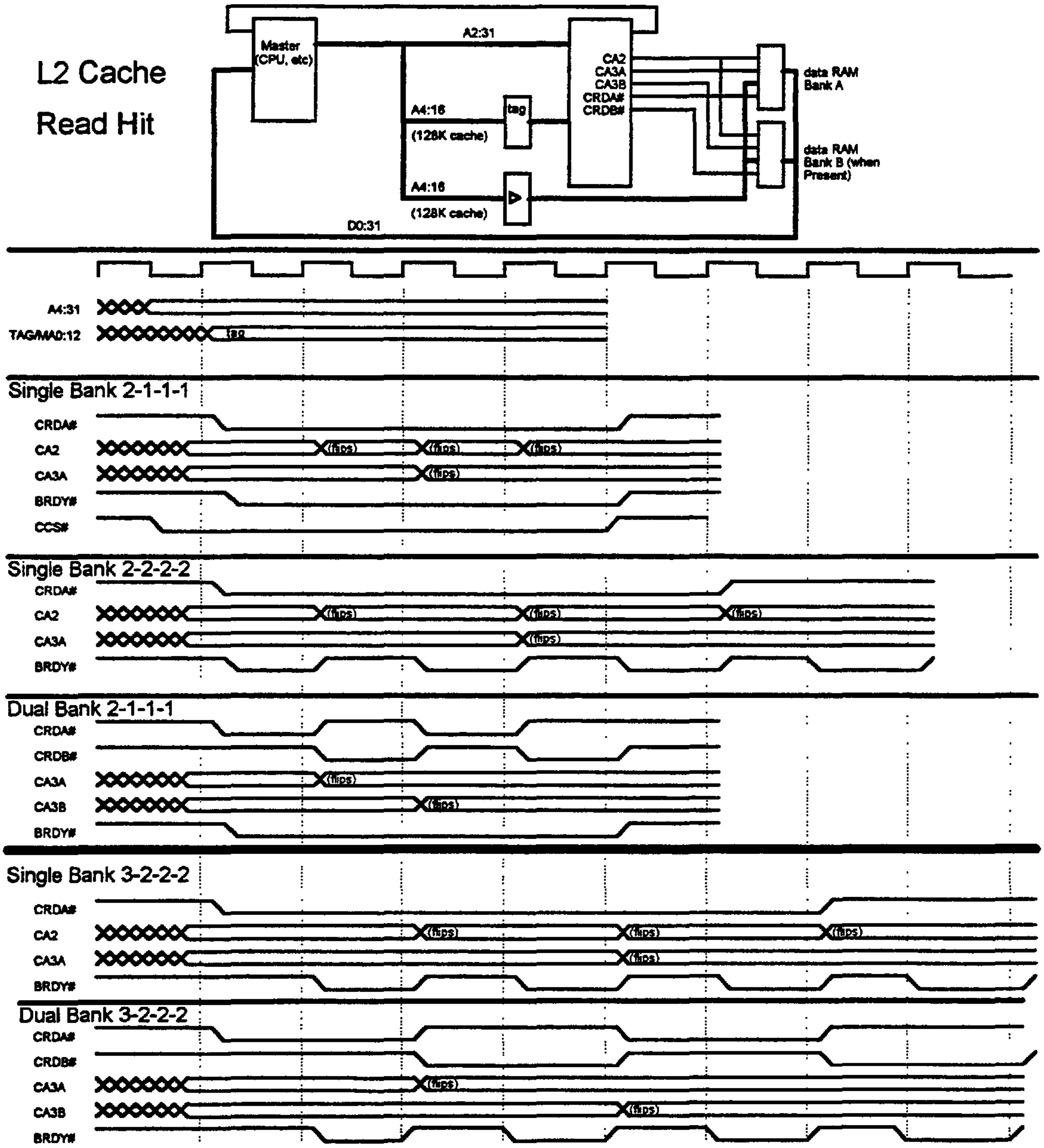


Figure 5.5: L2 Cache Read Hit

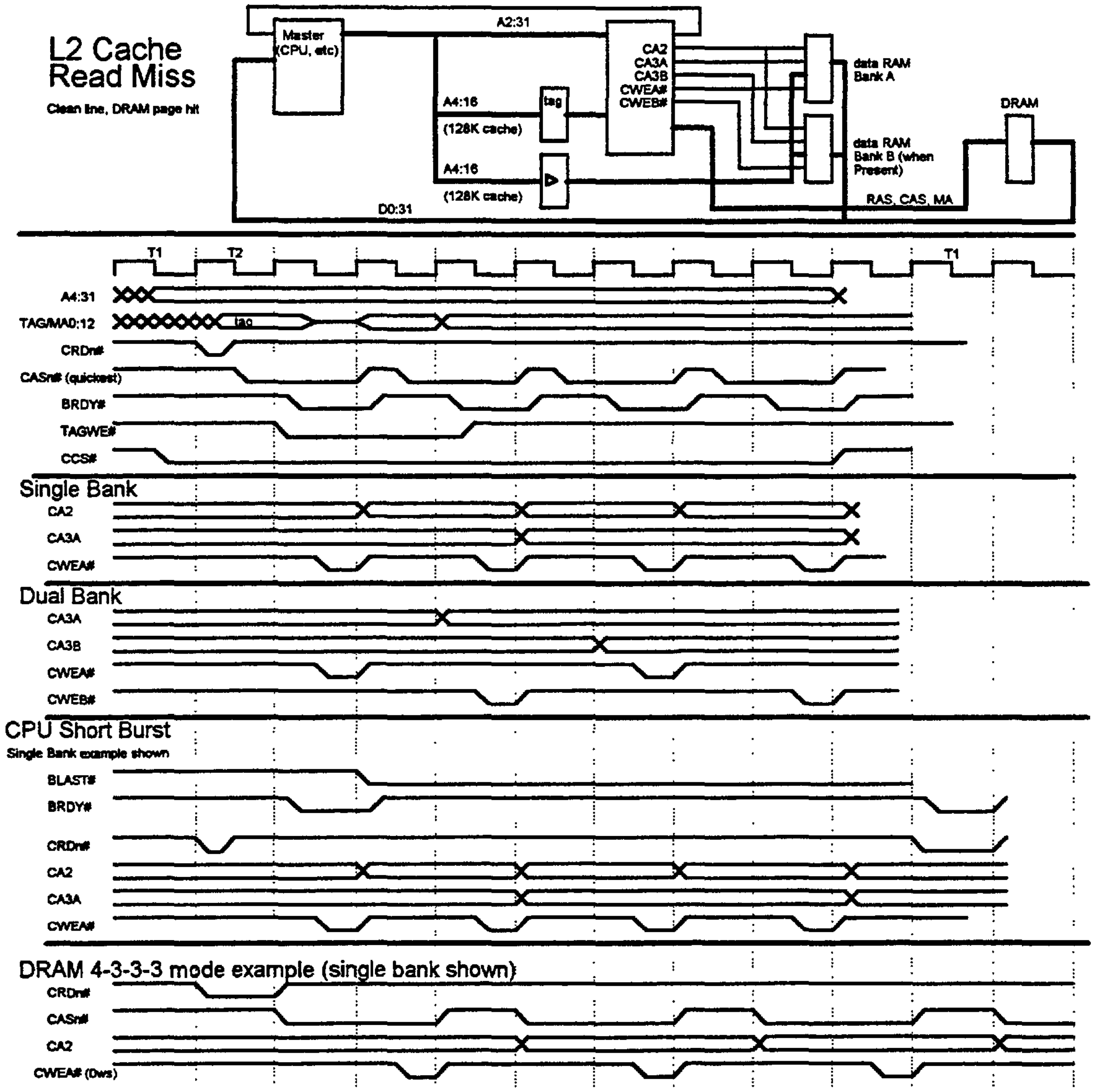


Figure 5.6: L2 Cache Read Miss

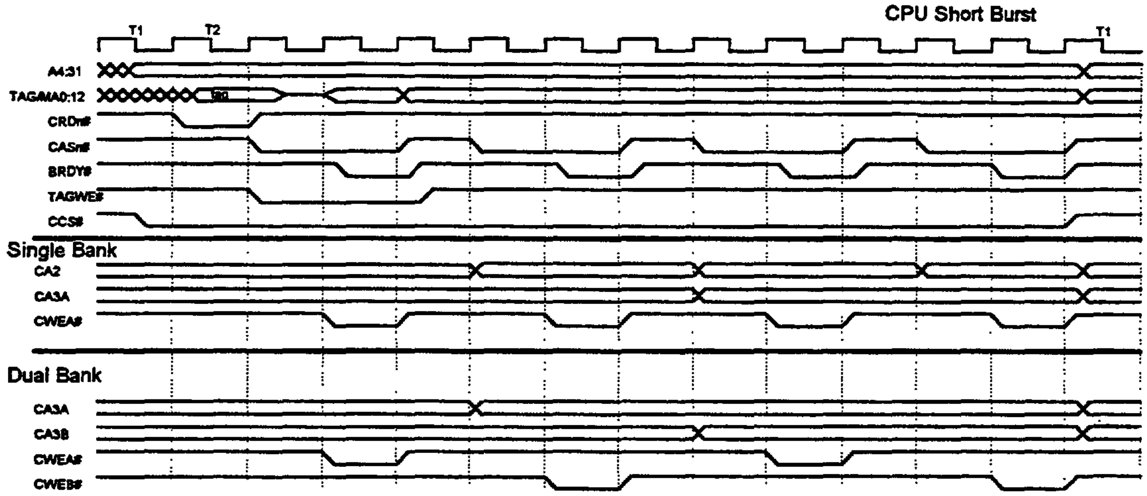


Figure 5.7: L2 Cache Read Miss, -2-2-2 Cache Writes

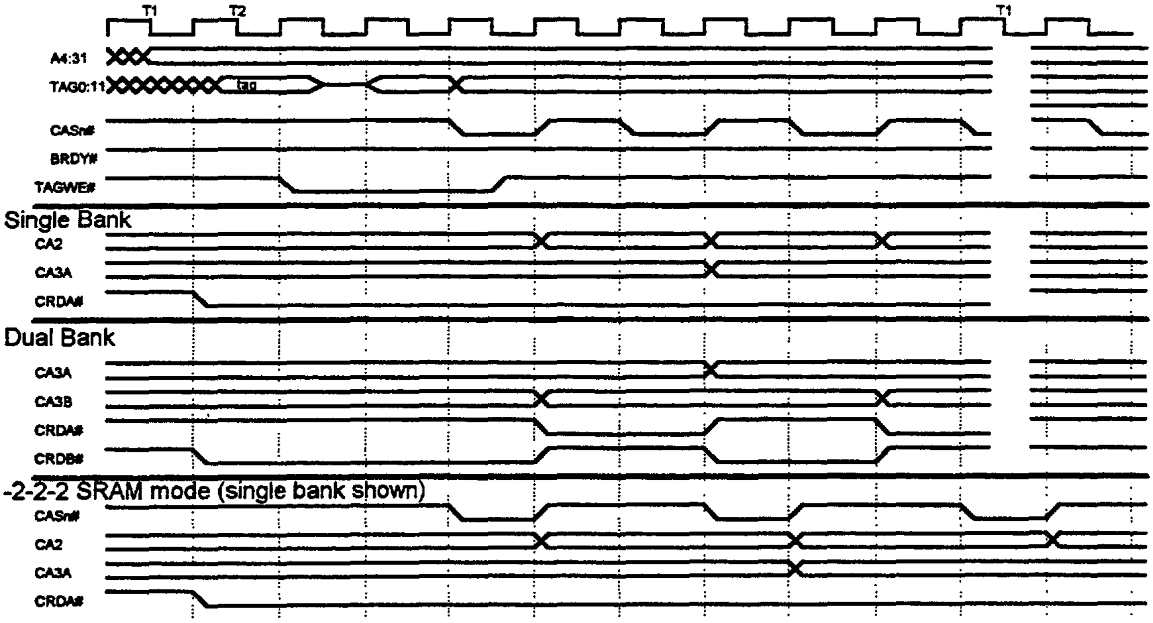
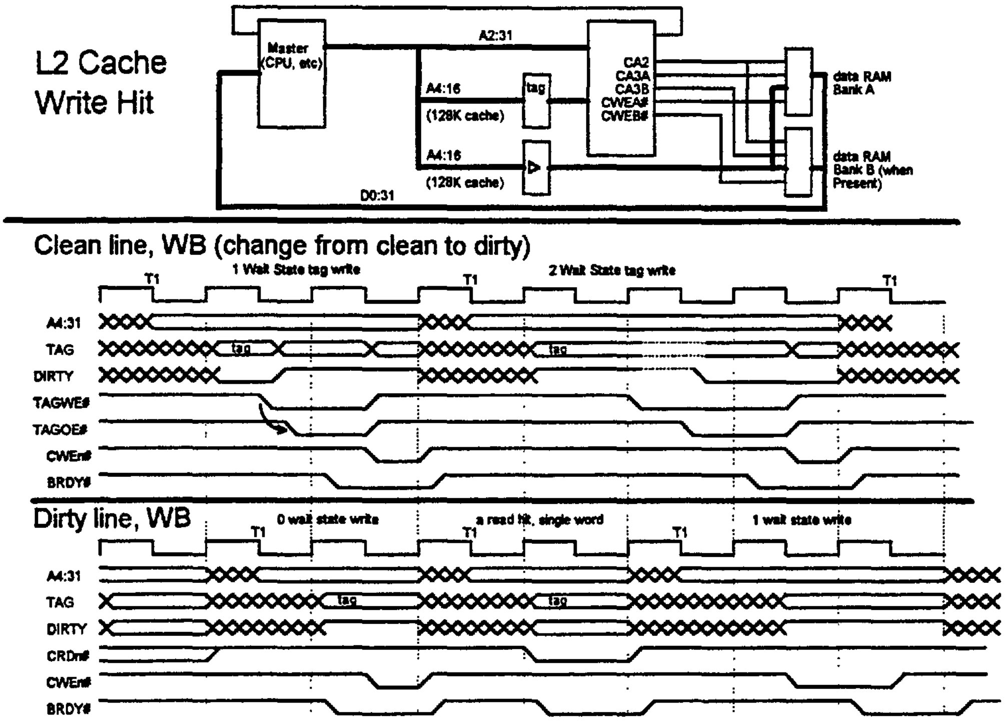


Figure 5.8: L2 Cache Write Back



Note: The 1WS option for clean-to-dirty tag write is not currently implemented. It is included for general illustration only. The clean-to-dirty tag write is always 2WS.

Figure 5.9: L2 Cache Write Hit

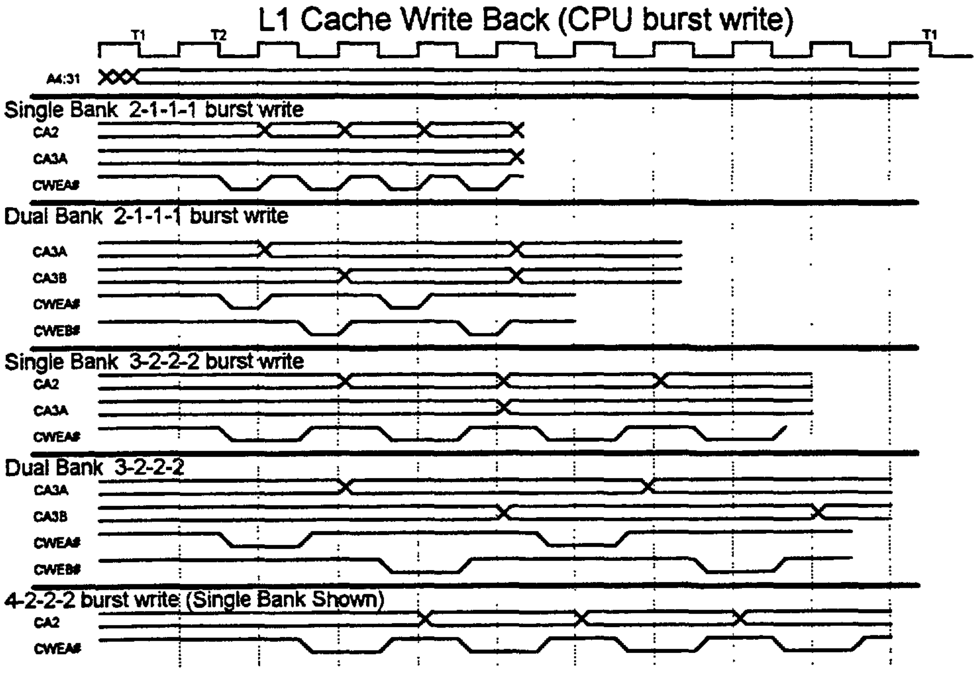

L1 Cache Write Back (CPU burst write)
T1
T2
T1
A4:31
Single Bank 2-1-1-1 burst write
CA2
CA3A
CWEA#
Dual Bank 2-1-1-1 burst write
CA3A
CA3B
CWEA#
CWEB#
Single Bank 3-2-2-2 burst write
CA2
CA3A
CWEA#
Dual Bank 3-2-2-2
CA3A
CA3B
CWEA#
CWEB#
4-2-2-2 burst write (Single Bank Shown)
CA2
CWEA#

Figure 5.10: L1 Cache Write Back

5.9.9. Cache Mode and Initialization

Cache "enabled" means that the cache is operating normally, performing line fills and cache hits as needed. Before enabling the L2 cache, software must first initialize it. "Cache initialize" means filling the entire cache with valid cache lines (data and tags) and making sure the cache contents remain valid at the time the cache is finally enabled. Initialization is accomplished by setting the cache to "Initialize" mode and reading a block of DRAM equal in size to the sum of L1 and L2 cache sizes. Reading a 640KB block of DRAM is sufficient to initialize all cache sizes except 1MB.

Disabling a writeback cache also requires special care. A WT cache can be disabled anytime, but a WB cache that has been in the "Enabled" condition may contain the only valid copy of certain DRAM data. The data in the cache must be flushed, i.e., written back out to DRAM, before disabling the cache. This can be accomplished by reading a block of DRAM equal in size to TWICE the sum of L1 and L2 cache sizes. Reading a 640KB block of DRAM is sufficient to flush all cache sizes except 512KB and 1MB.

The secondary cache may be switched from "Disabled" to "Initialize mode" or vice versa at any time. Cache initializing and flushing is explained in more detail below.

To facilitate initialization of the cache and disabling of the cache, the following cache control bits have been defined.

WRMODE Write Mode. Selects the write-hit policy.

0	Write through
1	Write back

ENCACHE Enable Cache.

0	Disable cache
1	Enable cache

INITCACHE Initialize cache. All read cycles are forced to be read misses. All cacheable memory read cycles update the cache data and tag as normal. No castouts are performed. Write hits are handled as write through. ENCACHE should be a 0 when this mode is being used.

0	Normal operation
1	Initialize cache mode

FRZCDIR Freeze cache directory

0	Normal cache operation
1	No directory update.

No updates on cache miss. No castouts occur either. Write hits still update the cache and dirty bit. Read hits come from cache. This mode emulates direct static RAM. It may be used in either write through or write back mode.

Table 5.14: Cache Modes.

encache	frzcdir	initcache	wrmode	Mode
0	0	0	X	Cache off (default).
0	0	1	X	Initialize cache. Writes forced write thru.
0	1	0	X	Cache off
0	1	1	X	Nothing useful.
1	0	0	0	Normal Cache operation, write through
1	0	0	1	Normal Cache operation, write back
1	0	1	X	Invalid
1	1	0	0	Directory Frozen, write through.
1	1	0	1	Directory Frozen, write back.
1	1	1	X	Invalid.

Initializing The Cache: The tag RAM will come up with random data, including the dirty bit. If the cache were enabled at this point, interesting things would happen. Most of them bad. The cache contains no valid bits. Once the cache is initialized, it is always valid. The cache is initialized by allowing the normal read-miss process to fill the cache with valid data. This is done by reading a block of memory which is the size of the cache. Read-hits are disabled to prevent an uninitialized tag from accidentally indicating a hit, and to force the read cycle to fill the cache with valid data. Castouts are disabled to prevent invalid data from being cast out into the DRAM during initialization.

The INITCACHE bit, when set overrides the FRZCDIR bit, which it should be set to '0'. It also overrides the WRMODE bit, which may be set either way, and may be set to the desired mode throughout the procedure.

Initialize Cache Algorithm.
ENCACHE=0, FRZCDIR=0, INITCACHE=0 to start. WRMODE=X
Set INITCACHE=1.
Read a block of cacheable memory which is at least as large as the cache size.
At the same time, set ENCACHE=1, INITCACHE=0, and WRMODE to desired mode.

Note: Before setting the INITCACHE bit to '1', Index 21h must be programmed to a value that correctly matches the physical cache configuration. Failure to do this will result in an invalid initialization of the cache contents and may also cause a software crash if any software is being executed from cacheable DRAM during the initialization process. (The reason for the potential software crash is that the 4041 relies on a cache read following each line fill to provide the read data requested by the CPU, including any code fetches from cacheable DRAM.)

Disabling the Cache: Disabling a write back cache, if not handled carefully, can result in an "Output and pull rug" instruction if there is dirty data in the cache. To avoid this, a mechanism is provided to dump the dirty data back into the DRAMs. This involves reading a cacheable memory block twice the size of the cache. Twice the size is required to make sure each location gets a read-miss, which will cause a cast-out if the dirty bit is set. The cache is then disabled. No writes should occur during this process.

Disabling The Cache: Write Through.
(ENCACHE=1, FRZCDIR=0, INITCACHE=0, WRMODE=1 to start (normal write through mode))
Set ENCACHE=0.

Disabling the cache: Write Back.
(ENCACHE=1, FRZCDIR=0, INITCACHE=0, WRMODE=1 to start (normal write back mode))
Read a block of cacheable memory which is at least TWICE as large as the cache size. No cacheable write cycles should be done during this time until after the NEXT step.
Set ENCACHE=0.

Switching Between Write Back to Write Through. To switch from write through to write back, nothing special need be done. Just flip the bit. To switch from write back to write through, the dirty data must be cast out. A block of cacheable memory twice the size of the cache must be read, with the WRMODE bit set to a 0.

Switching from Write Through to Write Back.
(ENCACHE=1, FRZCDIR=0, INITCACHE=0, WRMODE=0 to start (normal write back mode))
Set WRMODE=1.

Switching from Write Back to Write Through.
(ENCACHE=1, FRZCDIR=0, INITCACHE=0, WRMODE=1 to start (normal write back mode))
Read a block of cacheable memory which is at least TWICE as large as the cache size. No cacheable write cycles should be done during this time until after the NEXT step.
Set WRMODE=0.

Freezing the Cache Directory. While the cache is operating in write back or write through mode, the directory can be frozen to prevent new data from going to the cache. Writes will occur as normal, including updating the dirty bit if necessary. Read misses will simply not update the cache data or tag RAMs. This bit may be turned on and off at any time except when initializing the cache. A possible use is for preventing large data moves from thrashing the cache, or trying to cheat on a benchmark.

Freezing and Unfreezing the Cache Directory.
Flip the FRZCDIR bit as needed.

5.9.10. Tag & Data SRAM Testing

The 4041 has test modes for both the cache tag and data RAMs. The purpose of these modes is for the BIOS software to determine the size of the cache, and test its functionality before enabling it.

Testing the data RAMs basically makes them appear as static RAM within the test window. The CPU reads and writes to them as if they are just a static RAM bank. Testing the tag is a little more involved since the CPU does not have direct access to the tag data. The tag data is stored in a register within the 4041, which the CPU reads and writes through I/O ports 22 and 23.

There are two choices for the test window. One option places it in the bottom megabyte of the address space, allowing it to be used in REAL mode. The other places it at the 1Meg point, and must be used in protected mode. Register 22 bit 3 determines which mode is used. The addresses are as follows:

Table 5.15: Cache Test Window Location

Reg 22 bit 3	Cache Size bits	Test Window address range
0	64K, 128K, 256K	40000-7FFFF (256K to 512K area)
0	512K	20000-9FFFF (128K to 640K area)
1	All	100000-17FFFF (1M to 2M area)

The only effect of the cache size bits in either test mode is to determine the window size when the window is in the bottom meg. This allows more available DRAM when testing a cache that is 256K or smaller. The 512K test window is required for tag sizing, to determine whether the cache is 256K or 512K, however.

Note that the CPU addresses go directly to the tag and data RAMs (through an F244 for the data RAMs). The test mode does no mapping of any kind. When there is a 512K test window in the bottom meg, the address mapping is shifted a little, as follows:

Table 5.16: 512K Cache Test Mode Mapping When In The Lower Meg

CPU address	Physical SRAM address
80000-9FFFF	00000-1FFFF
20000-7FFFF	20000-7FFFF

5.9.10.1. Data SRAM Testing

When Data SRAM Testing mode is enabled, the cache appears as static RAM within the test window. The tag comparator is disabled (by virtue of the cache being disabled) and hits are forced whenever a memory cycle is done within the test window. All timing is the same as normal cache hits, as programmed in the configuration registers. Writes have the timing of a Write Hit Dirty. The DRAM controller will see these cycles as cache hits, and will be activated. All cycles outside the test window will act like a non-cacheable area. No line fills, etc.

The tag test registers, Indexes 23 & 24, are actually separate registers for reading and writing. After writing index 23, reading it back will NOT give the data just written, but will give the value stored in the register during the last read from the test window with tag testing enabled.

The following occurs when Data SRAM test mode is enabled:

Reads within the test window will read the SRAMs and be at the programmed timing mode.
Writes within the test window will write the SRAMs and be at the programmed timing mode.
The tag is ignored.
No castouts occur.
Memory accesses outside the test window go to DRAM, the AT bus, etc. and are not cached.
No line fills occur
No TAG writes occur
The test window decode is substituted for the tag hit signal.
The DRAM sees test window accesses as cache hits.
Writes will not go to DRAM regardless of WRMODE.
The cache size bits and tag width bits are ignored, except to determine the window size, as indicated in the above table.
The KEN# signal is not affected by this mode.

The cache mode bits must be set as follows:

ENCACHE = 0 INITCACHE = 0 FRZCDIR = 0 WRMODE=X

<u>To size the Data SRAM:</u>

Set the test window to the desired location. If it is in the lower 1M, set the cache size to 512K to get the full 512K byte test window size. If a 1M cache is to be tested, the window must be at 1M.
Set the cache mode bits as indicated above.
Disable the 486 internal cache to avoid getting fooled (or use Page Cache Disable).
Set the Data SRAM Test mode ON.
Test for single or dual bank as follows:
Set the Single/Dual bank bit to Dual Bank.
If single bank is installed, bytes 4:7 and C:F of each cache line will not work (the bus will float on reads).
If single bank is detected, switch the Single/Dual bank bit to Single bank.

Test the size of the SRAM as summarized below. (This is fairly complex because in dual bank mode the programmed cache size affects which CPU address bit is sent to CA2. See note in CA2 section above.) Stop on the first case that works:
Set the cache size to 1M and verify that the RAM does not repeat when changing A16, 17, 18, & 19.
Set the cache size to 512K and verify that the RAM does not repeat when changing A16, 17, & 18.
Set the cache size to 256K and verify that RAM does not repeat when changing A16 & 17.
Set the cache size to 128K and verify that RAM does not repeat when changing A16.
If none of the above works, assume the cache data size is 64K.
All lower address bits should be checked to verify SRAM integrity.

<u>To test the Data SRAM:</u>

Set the test window to the desired location. If it is in the lower 1M, set the cache size to 512K to get the full 512K byte test window size. A 256K window may be used by programming any other cache size if it is already known that the cache is less than 512K.
Set the cache mode bits as indicated above.
Disable the 486 internal cache to avoid getting fooled.
Set the Data SRAM Test mode ON.
Do a RAM test within the test window.

5.9.10.2. Tag Testing

Tag testing is a little more complicated than data RAM testing. Like the data RAM test mode, tag testing involves the CPU reading and writing to the test window. The CPU cannot directly read or write the tag RAM, however. Registers at index 23 and 24 hold the data for tag testing.

On a CPU read from the test window, the tag data is written into index registers 23 & 24 at the end of T2. This is a 0WS read, and the CPU received garbage data. The CPU must read indexes 23 and 24 to get the data.

On a CPU write cycle to the test window, data from index 23 and 24 gets written to the tag RAMs. The timing is identical to that of a "Write Hit Clean". It will be either 1 or 2 wait states depending on the "Tag Write Timing" bit (index 20 bit 0). Before doing the write to the test window, the CPU writes indexes 23 and 24 with the desired data.

Note that there is one tag location for every 16 bytes of data. The Tag RAMs get A4 as the lowest address bit. An access to any byte within the 16 byte, word, or dword within the 16 byte tag line will read or write the tag RAM. Consecutive tag RAM locations should be accessed by incrementing the CPU address by 16 bytes (10H).

The cache mode bits must be set as indicated for the Data SRAM test mode. The data RAMs are disabled during this time.

The following occurs when Tag test mode is enabled:

Reads within the test window will latch the tag data at the end of T2. The cycle will be 0WS. The CPU will read garbage data. RDY# will be returned, not BRDY#.
Writes within the test window will write the test data to the tag with the same timing as a Write Hit Dirty (1 or 2 wait states)
The tag comparator is disabled.
No castouts occur.
Memory accesses outside the test window go to DRAM, the AT bus, etc. and are not cached.
No line fills occur
The test window decode is substituted for the tag hit signal.
The DRAM sees test window accesses as cache hits.
Writes will not go to DRAM regardless of WRMODE.
The cache size bits and tag width bits are ignored, except to determine the window size, as indicated in the above table.
The KEN# signal is not effected by this mode.

To size the Tag RAM:

Set the test window to the desired location. If it is in the lower 1M, set the cache size to 512K to get the full 512K byte test window size. To test for a 1M cache size, the windows must be at 1M.
Set the cache mode bits as indicated above.
Disable the 486 internal cache to avoid getting fooled.
Set the Tag Test mode ON.
Test to see if and where the SRAM repeats itself, incrementing by 10H for each tag location, reading and writing the tag as indicated below:

To test the RAM:
Set the test window to the desired location. If it is in the lower 1M, set the cache size to 512K to get the full 512K byte test window size. A 256K window may be used by programming any other cache size if it is already known that the cache is less than 512K.
Set the cache mode bits as indicated above.
Disable the 486 internal cache to avoid getting fooled.
Set the Tag Test mode ON.
Do a RAM test within the test window, reading and writing the tag as indicated below:

To write a tag location:

Write the desired data to Indexes 23 & 24.
Write to the desired memory location in the test window.

To read a tag location:

Read the desired memory location in the test window. Ignore the data.
Read the data from Indexes 23 & 24.

5.10. DRAM Controller

Features:

Up to 8 banks of DRAMs (4 double bank SIMMs, etc.)
Page mode and Page interleave.
256K, 1M, 4M, and 16M deep DRAMs supported.
Direct drive RAS.
Direct drive CAS, DWE, and MA for up to 2 banks.
Hidden refresh with RAS staggering.

Optional DRAM data buffers.
CPU and alternate master timing modes may be set separately.
Support for CPUs with a write back cache
Support for SMM memory separate from user space.

Timing selections:

Burst Reads: 3-2-2-2, 4-3-3-3, or 5-4-4-4 timing modes
Single writes: 1 or 2 wait state.
Burst Writes: 3-2-2-2, or 4-3-3-3.
RAS to CAS timing: 2 or 3 T states (1.5 or 2.5 for 3-2-2-2 burst reads).
RAS pulse width for refresh: 3 or 4 T states.
RAS precharge: 2 or 3 clocks

The DRAM controller is designed to support 4 double bank 36 bit SIMMs for a total of 8 banks. It will also support 9 bit SIMMs, and any combination of both. There are 4 DRAM blocks, each of which may contain 1 or 2 banks.

5.10.1. Block Decodes

Up to 8 banks of DRAMs are allowed. There are 8 RAS pins, one for each bank, and 4 CAS pins, one for each byte, shared by all banks. The banks are grouped together into 4 DRAM "blocks".

Table 5.17 RAS and CAS usage

Block #	RAS used when one bank installed	RASes used when two banks installed	CASes used:
Block 0	RAS0#	RAS0# & RAS4#	CAS0-3#
Block 1	RAS1#	RAS1# & RAS5#	CAS0-3#
Block 2	RAS2#	RAS2# & RAS6#	CAS0-3#
Block 3	RAS3#	RAS3# & RAS7#	CAS0-3#

Each DRAM block has the following programmable bits:

Starting Address (A27-20)
DRAM depth (none, 256K, 1M, 4M, or 16M)
Number of banks installed (1 or 2)
Interleave bit

Starting Address. The Starting address provides the decode. A27 through A20 may be programmed. As the block size gets bigger (as determined by the DRAM depth and a combination of the interleave bit and the number of banks) successive lower bits in the starting address are ignored. The DRAM block must be placed on a block size boundary, as shown in the following table:

Table 5.18: DRAM Block Starting Address

DRAM Depth	Interleave and number of banks	Size of Bank	Address bits programmed	Placement Boundary size
256K	non interleaved single bank	1M byte	A20-27	1M
256K	interleaved bank OR dual bank	2M byte	A21-27	2M
1M	non interleaved	4M byte	A22-27	4M
1M	interleaved	8M byte	A23-27	8M
4M	non interleaved	16M byte	A24-27	16M
4M	interleaved	32M byte	A25-27	32M
16M	non interleaved	64M byte	A26-27	64M
16M	interleaved	128M byte	A27	128M

For example, if bank 2 contains 256K x 32 DRAM and bank 3 contains 1M x 32 DRAM and banks 0 and 1 are empty, the starting address for bank 3 should be 0 and the starting address for bank 2 should be 4MB (4xxxxxh). A starting address of 1xxxxxh for a 1M x 32 bank would be invalid (decoded as 0xxxxxh). To avoid holes or overlaps in the bank decoding, largest banks should map lowest in the address space.

Interleave bit. The interleave bit is used for single-bank blocks only (RAS0-3). It causes an even numbered bank (RAS0# or RAS2#) to map at even DRAM pages only, with twice the total block size. An odd numbered bank (RAS1# or RAS3#) maps at odd DRAM pages only, also with twice the total block size. Thus, if two banks are both the same size and have the same starting address, but one is odd and the other is even, turning on both interleave bits allows two-way page-interleaving to occur between the two banks. Interleaving is discussed in more detail in the next two sections.

If a block contains two banks (e.g., RAS0# and RAS4#), the two banks are page-interleaved automatically and the interleave bit for that block should be zero.

DRAM depth. The DRAM depth bits define the type of DRAMs installed. This is a 3 bit field, in which the 000 setting disables the bank. The following are the allowable configurations:

Table 5.19: DRAM Size Options

Prog bits	DRAM depth	DRAM types	non-int block size	Interleaved block size
000	disabled		0K	0K
001	256K	256Kx1, 256Kx4, 256Kx16	1M	2M
010	1M	1Mx1, 1Mx4, 1Mx16	4M	8M
011	4M	4Mx1, 4Mx4	16M	32M
100	16M	16Mx1	64M	128M

5.10.1.1. Of Page Mode, Page Interleaving, and Single-RAS Active

This CHIPSet employs Page Mode, and Page Interleaving techniques. These are two different techniques, which may be used independently or together.

Page Mode is always used in this CHIPSet for CPU accesses, both for bursts and between bursts. Page mode means keeping RAS low while reading or writing multiple words within a DRAM page by providing only a new column address and toggling CAS. A DRAM page is defined as a set of DRAM locations that can be addressed by changing column address bits only. The locations in a page will be contiguous if the lowest CPU address bits map to the column address, as is the case with the 4041. Address muxing is discussed further in Section 5.10.2.

Page Interleaving involves interleaving DRAM banks on a DRAM page boundary. This increases the percentage of time that a page miss cycle will hit the opposite bank of DRAMs, hiding the RAS precharge of one bank while accessing the opposite bank.

Single-RAS Active refers to allowing only one RAS# signal to be active at a time. With added CAS# pins (not supported by the 4041), more than one RAS# potentially could be active at the same time, allowing DRAM bank switching to occur without changing the state of the RAS# signals. Instead of just four CAS# signals, a fully general four-bank multi-RAS active architecture would require 16 CAS# signals, four for each bank, and the performance benefit would be minimal in systems that have both a CPU cache and a secondary cache. To conserve pins, the 4041 uses Single-RAS active only.

5.10.1.2. Controlling Page Mode and Page Interleaving

Page Mode is automatic. Page mode is always used for CPU and Local Master initiated cycles. **Page Interleaving** is controlled by the DRAM configuration bits, as discussed below.

The 4041 can perform two-way interleaving. Two banks may be interleaved if they are the same size (depth). If two banks are installed in a block, they are automatically interleaved by virtue of the 2 banks bit being set (the two banks in a block are required to be the same depth). Two single bank blocks may be interleaved if they are the same size, and one is an even numbered block and the other is an odd numbered block. This is done by setting their starting addresses to be the same and setting both interleave bits. Interleaving and non-interleaving may be mixed, and multiple sets of banks may be interleaved. For example:

Example 1: Banks 0 & 1 have 256K deep parts and be interleaved with each other.
Bank 2 may have 1M parts, not interleaved with anything.
Bank 3 may have 4M parts, not interleaved with anything.

Example 2: Banks 0 & 3 have 256K deep parts and are interleaved with each other
Banks 1 & 2 have 1M parts and are interleaved with each other.

Example 3: Block 0 has a double bank 256K SIMM (using RAS0# & RAS4#) which is interleaved with itself.
Blocks 1 & 2 have single bank 1M SIMMs (using RAS1# & RAS2# respectively) which are interleaved with each other.
Block 3 has a single bank 256K SIMM which is not interleaved with anything.

The following bank combinations may be interleaved:

	0 & 1	0 & 3
1 & 2	2 & 3	
	0 & 4	1 & 5
2 & 6	3 & 7	

The interleave scheme used in the 4041 is designed to be very versatile for the user. When a block is interleaved, its block decode is doubled in size. The interleave bit (either A11 or A12) is included in the bank decode and decoded to either a 0 or a 1 depending on the bank number. This makes the bank show up in every other page (2K or 4K). Table 5.20 shows what CPU address bits are used for the DRAM row address, Column address, bank decode, and the interleave bit.

Two banks are interleaved when they both have their block starting addresses pointing to the same place and one decodes the interleave bit to a 0 and the other one decodes it to a 1.

When both banks of a block are installed they are automatically interleaved. The interleave bit decoding was chosen such that the two banks will always decode the interleave bit in opposite directions. When a block is programmed for two banks the interleave bit is ignored, since interleaving is automatic. There is also only one starting address for the block, so all that need be done is programming the block for two banks.

Two banks in different blocks may be interleaved by programming the starting addresses to be the same and setting the interleave bit for both blocks. The DRAM depth must be the same to avoid getting a memory map which is full of holes. It is required that one of the blocks be even numbered and one be odd numbered so that one will decode even pages (interleave bit decoded to a 0) and one will decode the odd pages (interleave bit decoded to a 1). If both are even the two banks will sit on top of each other on the even pages and the odd pages will be empty, and likewise if they are both odd. A double bank block may not be interleaved with another block since it is already interleaved with itself.

The algorithm below can be used to auto configure for interleaving. A double bank block will automatically interleave itself prior to this algorithm.

```
if block 0 & 1 are the same size & both have only 1 bank installed.
                    Program blocks 0 & 1 to the same address and set both interleave bits.
endif
if block 2 & 3 are the same size & both have only 1 bank installed.
                    Program blocks 2 & 3 to the same address and set both interleave bits.
endif
if blocks 0 & 3 are the same size, both have only 1 bank installed, and neither have been interleaved above
                              Program blocks 0 & 3 to the same address and set both interleave bits.
endif
if blocks 1 & 2 are the same size, both have only 1 bank installed, and neither have been interleaved above
                              Program blocks 1 & 2 to the same address and set both interleave bits.
endif
```

5.10.2. Address Muxing

Table 5.20: CPU Address Assignments For Interleaving and Non-Interleaving

CPU Address	256K no intrl	256K interleaved	1M no intrl	1M- interleaved	4M no intrl	4M interleaved	16M no intrl	16M interleaved
2	Col 0	Col 0	Col 0	Col 0	Col 0	Col 0	Col 0	Col 0
3	Col 1	Col 1	Col 1	Col 1	Col 1	Col 1	Col 1	Col 1
4	Col 2	Col 2	Col 2	Col 2	Col 2	Col 2	Col 2	Col 2
5	Col 3	Col 3	Col 3	Col 3	Col 3	Col 3	Col 3	Col 3
6	Col 4	Col 4	Col 4	Col 4	Col 4	Col 4	Col 4	Col 4
7	Col 5	Col 5	Col 5	Col 5	Col 5	Col 5	Col 5	Col 5
8	Col 6	Col 6	Col 6	Col 6	Col 6	Col 6	Col 6	Col 6
9	Col 7	Col 7	Col 7	Col 7	Col 7	Col 7	Col 7	Col 7
10	Col 8	Col 8	Col 8	Col 8	Col 8	Col 8	Col 8	Col 8
11	Row 0	Interleave	Col 9	Col 9	Col 9	Col 9	Col 9	Col 9
12	Row 1	Row 1	Row 1	Interleave	Row 1	Interleave	Row 1	Interleave
13	Row 2	Row 2	Row 2	Row 2	Row 2	Row 2	Row 2	Row 2
14	Row 3	Row 3	Row 3	Row 3	Row 3	Row 3	Row 3	Row 3
15	Row 4	Row 4	Row 4	Row 4	Row 4	Row 4	Row 4	Row 4
16	Row 5	Row 5	Row 5	Row 5	Row 5	Row 5	Row 5	Row 5
17	Row 6	Row 6	Row 6	Row 6	Row 6	Row 6	Row 6	Row 6
18	Row 7	Row 7	Row 7	Row 7	Row 7	Row 7	Row 7	Row 7
19	Row 8	Row 8	Row 8	Row 8	Row 8	Row 8	Row 8	Row 8
20	decode	Row 0	Row 0	Row 0	Row 0	Row 0	Row 0	Row 0
21	decode	decode	Row 9	Row 9	Row 9	Row 9	Row 9	Row 9
22	decode	decode	decode	Row 1	Row 10	Row 1	Row 10	Row 1
23	decode	decode	decode	decode	Row11 and Col 10	Row11 and Col 10	Col 10	Col 10
24	decode	decode	decode	decode	decode	Row 10	Row 11	Row 10
25	decode	decode	decode	decode	decode	decode	Row 12 & Col 11	Row 12 & Col 11
26	decode	decode	decode	decode	decode	decode	decode	Row 11

Col n Indicates that the address bit appears on this column address bit.
Row n Indicates that the address bit appears on this row address bit.
decode Indicates that this bit is included as part of the bank decode as programmed in the starting address register.
Interleave Indicates that this bit is included as part of the bank decode, and is decoded as a 0 for even numbered banks and a 1 for odd numbered banks.

Table 5.21: Address Multiplexing

MA	Column address (all sizes)	Row address choices	Row Address assignment for: 256K non-int	256K interlv	1 or 4M non-int	1 or 4M interlv	16M non-int	16M interleave
0	2	11 or 20	11	20	20	20	20	20
1	3	12 or 22	12	12	12	22	12	22
2	4	13	13	13	13	13	13	13
3	5	14	14	14	14	14	14	14
4	6	15	15	15	15	15	15	15
5	7	16	16	16	16	16	16	16
6	8	17	17	17	17	17	17	17
7	9	18	18	18	18	18	18	18
8	10	19	19	19	19	19	19	19
9	11	20 or 21	20	20	21	21	21	21
10	23	22 or 24	x	x	22	24	22	24
11	25	23, 24 or 26	-	-	23	23	24	26
12	-	25	-	-			25	25
Interlv bit			-	11	-	12	-	12
Page Size			2K	2K	4K	4K	4K	4K

5.10.3. Timing Modes

The timing modes are divided into the following parameters:

Burst Read timing (3-2-2-2, 4-3-3-3, or 5-4-4-4)
RAS to CAS timing (1.5 or 2.5(2.0) T states)
Single write timing (1 or 2 WS)
Burst Write timing (3-2-2-2, or 4-3-3-3)
RAS Precharge (2 or 3)
Refresh RAS length (3 or 4)

The burst read, burst write, and single write timings may be set separately for CPU cycles and alternate masters. Alternate masters include VL bus masters, ISA masters, and DMA cycles. This function allows alternate masters to run at a slower timing mode. This may be useful for the following reasons:

The 4041 must generate parity for DRAM write cycles for the alternate master (the CPU parity is used for CPU cycles).

The CPU is a more controlled part on the board. Its timing may be able to be assumed better than the worst case VL master might be. The CPU may also be located physically closer to the DRAMs and the 4041 than the worst case VL master.

The system designer may optimize for the last bit of CPU performance without having to worry about VL-Master timings, and what the end user may install later.

The three Burst read modes are shown below.

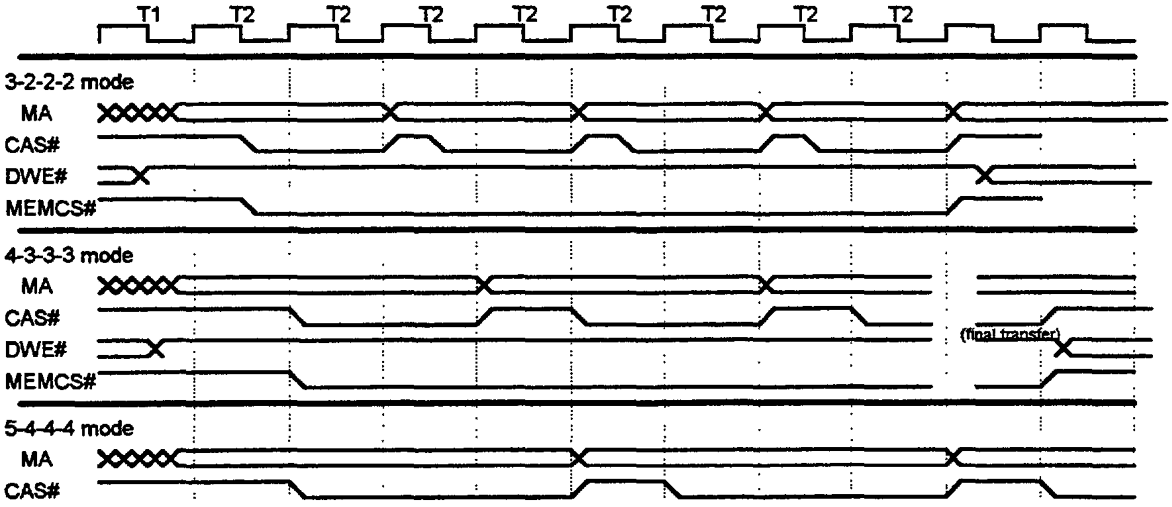


Figure 5.11: Burst Read Timing Modes

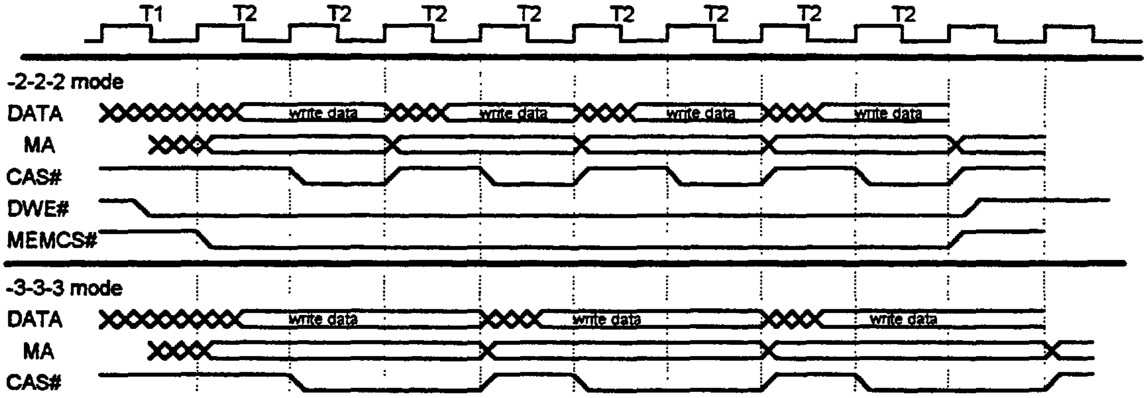


Figure 5.12: Burst Write Timing Modes

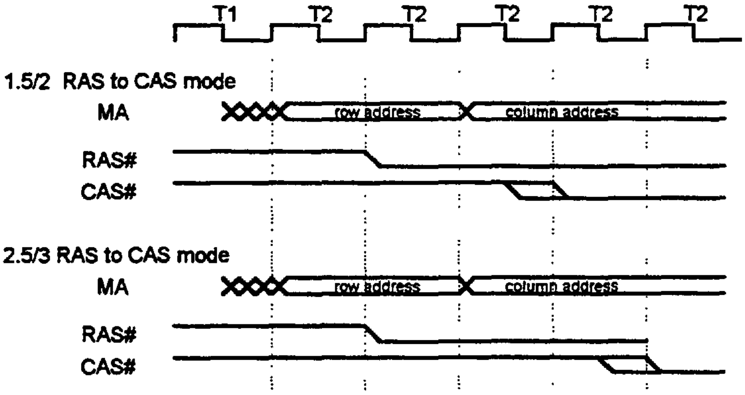


Figure 5.13: RAS to CAS Timing Modes

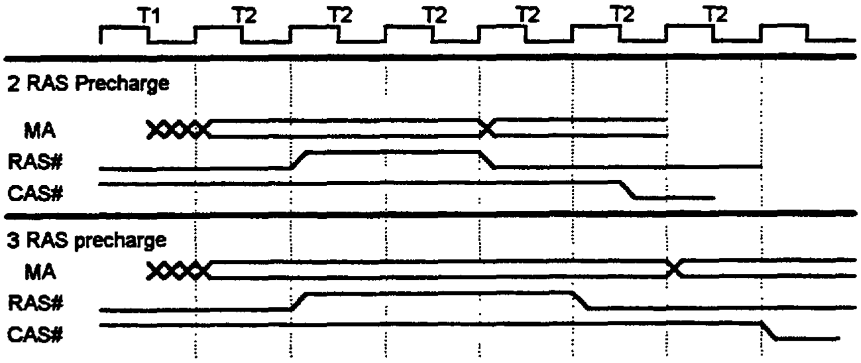


Figure 5.14: RAS Precharge Timing Modes

5.10.4. DRAM Refresh

Refresh is always hidden, regardless of who owns the bus. The CPU or alternate master is held in wait states as needed to allow refresh to occur, but HOLD-HLDA protocol is never used for refresh. DRAM refresh is always CAS-before-RAS (no refresh address required on the MA bus). To reduce power surges during DRAM refresh, RAS# signals are staggered as shown in the table below.

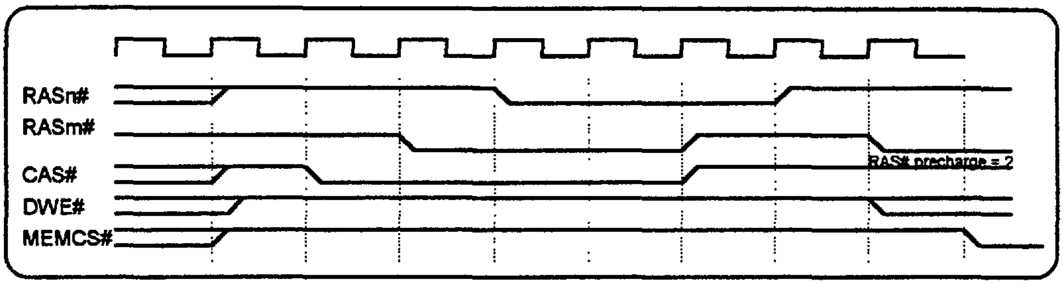


Figure 5.15: Refresh cycle

Table 5.22: Staggered Refresh Set Assignment

Set #	RASes	Goes first if:
1	0, 2, 5, 7	No RASes are active or a RAS in set 2 is active.
2	1, 3, 4, 6	A RAS in set 1 is active

Refresh requests are received from the 4045 through the control link. Refresh request pulses are received from the control link decoder in the 4041, and sent to the ISA state machine and the DRAM controller. The ISA state machine arbitrates the refresh request with the ISA bus controller for ISA refresh, sending an acknowledge back across the link when the bus is available (assuming ISA refresh is enabled).

DRAM refresh is done separately from ISA refresh. The DRAM controller arbitrates the refresh request with internal DRAM activity, and performs the refresh cycle when it is able. This may or may not coincide with the ISA refresh, and is generally much shorter than the ISA refresh. Since the MA bus is not required for refresh, the DRAM refresh may occur during an ISA bus access (which uses the MA bus as SD8:15).

5.10.5. DRAM Parity

Parity is generated by the CPU when it has control, and by the 4041 at all other times, including parity for VL bus masters, ISA Masters, DMA, and secondary cache castout cycles. Parity is generated asynchronously from the unlatched CPU data bus. Since the CPU generates even parity, the 84041 generates and expects even parity also. Parity is on a byte basis. Even parity means that a data value of 'FFh' has '0' for the parity bit.

Parity is checked by the 4041. It latches the data on the same clock edge as the CPU (or current master) and checks the parity one clock later.

5.10.6. Alternate Master Accesses (VL or ISA Master, or DMA)

When a VL master accesses the DRAM, it is handled the same as if the CPU is accessing it except that the 4041 generates parity, and there is a separate timing mode register which is used which allows a more relaxed timing mode to be used.

5.10.7. Programming the Timing Modes.

The following table gives the suggested programming for 60 and 70nS DRAMs at 33, 40 and 50MHz. The DRAM specs used were a composite of the more commonly available DRAMs. Many DRAMs are faster on many parameters than the composite. A few are slower. The 60nS DRAMs assume 15nS tCAC and the 70nS DRAMs assume 20nS tCAC. Other DRAM speeds, such as 50 and 80 are supported. In many cases the 50nS parts are not much better than 60nS parts for page mode cycles. 80nS parts may require some parameters to be relaxed slightly.

The table is reasonably conservative. Some of the parameters may be set more aggressively (some are noted with an "*"). Often EXTREMELY aggressive timing parameters work very well. If extremely aggressive timing is used, it is suggested that a more conservative mode be in register 1E, which is used by VL masters (it is also used for DMA and ISA masters). Since there will be a wide variety of VL-Masters some are likely to be close to the edge of the spec (or violate it completely). CPUs are often much faster than spec and are shipped with the system, which may allow a more aggressive timing mode. The ability of the user to install an upgrade CPU, which may not have better-than-worst-case timing should be kept in mind.

Table 5.23: Suggested DRAM Timing Modes

		33MHz				40MHz				50MHz			
	Index	non-cache		cache		non-cache		cache		non-cache		cache	
Parameter	10 bit(s)	60	70	60	70	60	70	60	70	60	70	60	70
Burst Reads	1:0	0	1*	1*	1*	1	1	1	1	2	2	2	2
Burst Writes	6	0	0	0	0	1*	1*	1	1	1	1	1	1
Single Write	4	0	0	0	0	0	0	0	0	1	1	1	1
RAS to CAS	2	0	0	0	0	0	0	0	0	0	0	0	0
RAS Prech	5	0	0	0	0	0	1	0	1	1	1	1	1
RAS low on ref	7	0	0	0	0	0	0	0	0	0	1	0	1
Total register	all	00	01	01	01	41	61	41	61	71	F1	71	F1

* Conservative setting. Next faster setting is slightly aggressive.

The RAS to CAS delay would be set to a 1 for 80nS DRAMs at 50MHz, and also at 33MHz with 3-2-2-2 mode.

Parameter	0	1	2	3
Burst Reads	3-2-2-2	4-3-3-3	5-4-4-4	(Reserved)
Burst Writes	3-2-2-2	4-3-3-3	-	-
Single Writes	3 (1WS)	4 (2WS)	-	-
RAS to CAS	1.5/2 clocks	2.5/3 clocks	-	-
RAS Precharge	2 clocks	3 clocks	-	-
RAS Low on Refresh	3	4	-	-

5.10.8. Automatic DRAM Sizing & Setup

The BIOS normally should auto-size the DRAM and support any possible DRAM configuration. It should not require any DRAM to be installed in block 0 or any other particular block. The BIOS should assume that there are four double-bank 72-pin SIMM sockets and that the user may install any valid DRAM type into any socket in any combination.

Each block can be checked individually, leaving other blocks disabled during the test. The block to be tested can be programmed as 16M deep, non-interleaved, single-bank. The various possible physical DRAM sizes will then function as described below:

256K deep (1MB total) — page size will be 2KB (800h). Leave A11 always zero (alternate pages missing). A21-26 are ignored, causing pages to repeat at 2MB intervals (0020 0000h).

1M deep (4MB total) — no missing pages. A22-26 are ignored, causing pages to repeat at 4MB intervals (0040 0000h).

4M deep (16MB total) — page size is 4KB (1000h). Leave A23 and A24 always zero. A25-26 are ignored, causing pages to repeat at 32MB intervals (0200 0000h).

16M deep (64MB total) — no missing or repeated pages over a full 64MB range (0400 0000h).

To test for a second bank in a block, program the block for the correct DRAM size as previously detected, and set the block for two banks. If alternate DRAM pages are missing, only one bank is installed in the block. The page size is 2KB (800h) for 256K deep DRAM, or 4KB (1000h) for all other DRAM sizes.

After sizing each block, enable interleaving between banks where possible. Two banks may be interleaved if the are single-bank blocks of the same size, and one is even (RAS0# or RAS2#) while the other is odd (RAS1# or RAS3#).

Finally, program the block starting addresses so that the largest blocks map lowest in the address space. If two banks are interleaved, they should map lower in the address space than a single bank of the same DRAM type. Interleaving doubles the effective block size for a given DRAM size. For example, a block containing two 4M deep DRAM banks (or two single-bank 4M deep blocks interleaved together) should map lower than a single 4M deep DRAM bank, but higher than a single 16M deep DRAM bank.

5.11. ISA Bus

5.11.1. CPU or VL Master Accesses to the ISA Bus

Bus cycles which do not get claimed by a local slave or the DRAM controller go to the ISA bus by default. Secondary cache has the first opportunity to claim a cycle. Next comes local DRAM, then VL slaves (via LDEV#), and finally ROM connected to ROMCS#. If the cycle is not claimed by any of these, it goes to the ISA bus. I/O cycles not claimed by LDEV# go to the ISA bus, except that I/O reads from on-chip or XD-bus peripherals will have the data busses steered as needed, blocking SD-bus data.

The ISA bus runs off of a clock which is derived from CLKIN. It should normally be set at about 8MHz, but may be set faster if all of the peripherals can handle the faster cycles. See the Clock section above for programming the ISA bus clock rate.

As noted in the Pin Descriptions and Figure 1, the MA2:9 lines are used for XD8:15 as well as the DRAM address, depending on whether the cycle is an ISA bus cycle or DRAM access.

There are two registers which control the timing of the ISA bus cycles. They are normally set to the default values, but may be adjusted, especially if the ISA bus clock rate is increased. There are three adjustments:

> **Command Delays.** These are adjustable separately for I/O, 8 bit memory, and 16 bit memory to be 0, 1, 2, or 3. 0 command delay activates the command on the falling edge of ALE. Each command delay added delays the activation of the command by one-half of a BUSCLK, or 62.5nS with an 8MHz bus. Command delays cut into the command active time. They do not extend the end of the command. That must be done by the wait state control, or by the ISA bus slave itself by pulling IOCHRDY. The normal settings are 0 for 16 bit memory and 1 for I/O cycles and 8 bit memory.
>
> **Wait States.** This sets the standard number of wait states for bus cycles. Wait states may be added by the slave by pulling IOCHRDY, or reduced by pulling 0WS#. The wait states are set separately for 8 bit and 16 bit slaves. 8 bit slave cycles are settable to 2, 3, 4, or 5 while 16 bit slave cycles are settable to 0, 1, 2, or 3. Zero wait states would be a two clock TS-TC cycle. With 0 command delays, that would be a 1 clock command. Each wait state adds a full BUSCLK. The normal values are 4 wait states for 8 bit slaves and 1 wait state for 16 bit slaves.
>
> **Address Hold Time.** This setting allows RDY# to the processor to be delayed by 1 extra CPU clock after the ISA bus command goes inactive. It will provide 1 T state of additional address hold time, as well as a longer period between ISA bus commands. It is normally disabled.

5.11.2. DMA or ISA Master Accesses to the ISA Bus

The 4041 performs byte swapping for these cases, and controls the XD buffers to properly steer the data when the slave is on the XD bus.

5.11.3. DMA or ISA Master Accesses to DRAM or VL Slaves

When DGNT# goes active (for DMA or ISA master cycles) the 4041 becomes an ISA bus slave, by floating the ISA commands, and a local bus master, by driving the local bus control signals. When a DMA or ISA master cycle attempts to access a device on the local bus, the 4041 performs the control signal translation and data steering for the cycle. Local Slaves and DRAM are handled identically.

5.12. Fast IDE

The 4041 Fast IDE provides much higher speed access to the IDE controller data ports than the standard ISA bus implementation, thus boosting disk performance.

Features:

Fast accesses to IDE drive data port
32 bit I/O cycles supported
Up to 8 drives supported (up to 4 IDE addresses)
Programmable I/O read length, I/O write length, command inactive, and address setup time
Each drive may select either of 2 programmed timing sets
May coexist with other IDE controllers at different addresses, either local bus or ISA
When enabled, it may disable an existing controller on the ISA bus
Transparently supports the 3F7 register sharing with the floppy, wherever the floppy is located

The 4041 IDE controller has a direct local bus interface to speed up access to the IDE drive. Drive accesses are speeded up by three means:

Eliminating the overhead of an ISA bus access
Allowing 32 bit I/O cycles to reduce CPU overhead (converted to two 16 bit IDE cycles)
Shortening the command timing to the drive

The first two will speed up any drive. The final one is very drive dependent.

5.12.1. Connections and Signal Generation

The following signals are specific to the fast IDE controller:

IDEIOR#	I/O read strobe
IDEIOW#	I/O write strobe
IDECS0#	IDE Chip Select 0. Command and data. 1F0-1F7 at primary IDE address.
IDECS1#	IDE Chip Select 1. Control. 3F6-3F7 at primary IDE address.
IDEEN#	IDE buffer enable.
P3F7D7	Port 3F7 bit 7. Disk Change from the floppy.

The following signals have special significance during IDE accesses:

XA0:1	IDEA0 and 1. Normally forced low.
SBHE#	IDEA2. Normally forced low.
IOCHRDY	IDE ready. Has special timing for IDE data transfers.
IOCS16#	IDE IOCS16#. May be forced active internally.

The following signals are also used by the IDE interface:

SYSRESET	IDE reset (inverted)
IRQ14	IDE interrupt

5.12.1.1. IDEIOR#, IDEIOW#, & IDEEN# Generation

IDEIOR# and IDEIOW# are the OR of the ISA IOR# and IOW# and the local bus IDE state machine signals. The ISA IOR# and IOW# are sometimes generated internally by the 4041 (CPU or VL master) and sometimes generated externally (DMA and ISA master).

The IDE state machine generates the IDEIOR# and IDEIOW# only for data port accesses (1F0 for the primary IDE address). The ISA state machine generates them for all other cycles.

IDEEN# is generated from the ISA IOR# and IOW# signals when they are accessing the IDE devices, and from the local bus state machine when it is accessing them.

5.12.1.2. IDECS0#, IDECS1#, XA0:1, and SBHE# Generation

IDECS0#, IDECS1#, XA0:1, and SBHE# are all forced to a default state when not in use. This default state is that of an access to the data port (1F0 on the primary IDE address) and provides the setup time for IDE data port reads. This allows the IDE access to start earlier in the bus cycle. The default states are high for IDECS1# and low for all of the others.

When a cycle goes to the ISA bus the signals revert to their normal values. IDECS0# and IDECS1# are address decodes and XA0:1 and SBHE# come from the ISA state machine. When the DMA controller or ISA master has control of the bus the IDECS0# and IDECS1# are address decodes and XA0:1 and SBHE# are sourced externally throughout the entire cycles.

SBHE# is a special case. When this function is enabled (the normal case) SBHE# becomes A2 during IDE cycles. When an IDE I/O port is accessed, A2 is substituted for SBHE#. This occurs only for IDE addresses which are enabled for the 4041 IDE controller, and only when the CPU or VL master has the bus. This function may be disabled and A2 supplied from the CPU A2. This provides less IDE address setup, but preserves the function of SBHE#.

5.12.1.3. P3F7D7

This is an input only bit to the 4041. During I/O reads from location 03F7 and 377 this bit is used for D7. This is to allow the floppy disk to provide this bit while the hard disk is providing bits 0:6. This is a shared I/O port on the original IBM AT disk controller card. It is now gaining almost as much fame and folklore as the infamous GATEA20 signal.

Externally this pin is usually connected to SD7. It may also be connected directly to the floppy's disk change pin.

5.12.1.4. IOCS16# and IOCHRDY Sampling

IOCS16# is not sampled by the fast IDE state machine. It is always assumed active for the IDE data port. If an 8 bit drive is used (assuming any exist in the first place), that drive number should be set to generate ISA cycles for the data port. Drives 0, 1, 2, or 3 may be set this way. Drives 4, 5, 6, and 7 may not.

Since all of the chip selects are tied together for the 4 IDE connectors, all 4 will be generating IOCS16# at the same time. To select the proper one, an external mux is used. This is not necessary if only 1 IDE connector is implemented. If it is desired that only 16 bit drives be supported it is not necessary even for multiple IDE connectors. IOCS16# need not even be hooked up in this case.

IOCHRDY is sampled by the fast IDE state machine. It is sampled 1 SCLK before the command is to go inactive. If it is low, the command will remain low until 1 SCLK after it is sampled high. The 1 clock delay is necessary for double clocking to prevent a metastable condition.

5.12.2. Cycle Description

5.12.2.1. Data Port

The data port accesses are the ones which are sped up and optimized for. A special state machine is used for these accesses which runs off of the CPU local bus.

The default signal states are set in anticipation of these accesses. This assures adequate address setup time to the command strobes. These default signal states are in effect when BOTH of the following are true:

ATEN is inactive (ATEN indicates an active ISA bus cycle)
DGNT# is high (indicating the CPU or VL master has the bus)

Table 5.24: Default signal states

Signal	Default
XA0, XA1, SBHE# (IDEA2)	0
IDECS0#	0
IDECS1#	1

The command for the data port may be generated as early as the end of the first T2, and most of the time this is the case. Several Timing Parameters are programmable. These are:

Table 5.25: Programmable Timing Parameters

Timing Parameter	# of choices	Selections	Per Drive Programming
Address Setup Time	4	1,2,3 or 4 clocks	Separate A and B programming
Read Command Length	16	1,2,3.....15,16 clocks	Separate A and B programming
Write Command Length	16	1,2,3.....15,16 clocks	Separate A and B programming
Command Recovery	16	1,2,3,4,5,6,7,8,10,12,14, 16,18,20,22,24 clocks	Separate A and B programming
Earliest Command Start	2	End of first or second T2	One global choice.
Write Data Hold	2	1 or 2 clocks	One global choice

Most of the parameters may be selected differently for different drives. Older drives cannot be run as fast as newer drives in general, and some newer drives are faster than others. The 4041 allows two sets of timing parameters to be specified and allows each of the 8 drives to select which one it uses. In a typical system which will allow for only 2 drives, this allows each drive to be programmed separately. Drives 0, 1, 2, and 3 may also be programmed to skip the local bus accesses and access the data port from the ISA bus. This would be required if the drive only supported 8 bit data accesses for some reason. IOCS16# will be sampled by the ISA bus. It is ignored, and assumed active for the data port by the fast cycle logic.

The internal logic determines which drive is being accessed according to the following table.

Table 5.26: Determine Which Drive Is Active

Drive #	Data Port Being Accessed	Drive Select Port & Value
Drive 0	1F0	Port 1F6 Bit 4=0
Drive 1	1F0	Port 1F6 Bit 4=1
Drive 2	170	Port 176 Bit 4=0
Drive 3	170	Port 176 Bit 4=1
Drive 4	5F0	Port 5F6 Bit 4=0
Drive 5	5F0	Port 5F6 Bit 4=1
Drive 6	570	Port 576 Bit 4=0
Drive 7	570	Port 576 Bit 4=1

Three parameters control when the command goes active: The T state start bit, which sets the earliest start at the end of either the first T2 or the second T2, the Address Setup selection, which sets the minimum number of T states that the address signals are setup (forced to their "default" states) before the command may go active, and the Command Recovery, which sets the minimum high time of the command. The Address Setup and command recovery parameters may be set differently for different drives.

Normally even if the address setup is set to a large number (4 is the maximum) the command may go active at the end if the first or second T2. This is because the default states of the address signals are generally present for many T states before the IDE access. An internal counter constantly monitors how long the default states have been held, and is checked at the start of the IDE data port access. The only time when the commands must wait for the address setup is following an ISA bus cycle (where the ISA address is present until near the end of the local bus cycle) and when the IDE cycle follows a DMA/ISA master cycle (where DGNT# was high). Even at these times the time-out will likely be achieved.

The Command Recovery Time is also constantly monitored by a counter. An active IDE command resets the counter, and it starts counting when both are inactive. A new command will not be issued until the select command recovery time has been met. The command recovery time only delays a data port access.

The Command Recovery Time is also used between commands on a 32 bit I/O operation.

The Read and Write command pulse widths are fairly self explanatory. These parameters are only used for data port accesses.

5.12.2.2. Command/Control Registers

All I/O ports of the IDE drives other than the data port are accessed by standard ISA cycles. The only difference is the steering of the external data bus drivers. The IOR# and IOW# ISA commands are copied onto the IDEIOR# and IDEIOW# pins. IDEEN# goes active as described below. On read cycles, SDIR0 and SDIR1 do not go low, since the data will be driven from the IDE buffers.

Port 3F7 is a special case. Here D0:6 come from the IDE controller, and D7 comes from the floppy controller. To accomplish this, a special input pin is used to provide bit 7. Normally this pin is connected to SD7. The difference is that SDEN# goes high to disable the SD bus buffers, and D7 is internally taken from the P3F7D7 pin instead of XD7. The IDE controller will provide data on XD0:7. XD0:6 will be used, and XD7 ignored. The P3F7D7 pin is an input only, as this port is only an input. If a write occurs to 3F7 it occurs as normal, and SDEN# & IDEEN# will both be active.

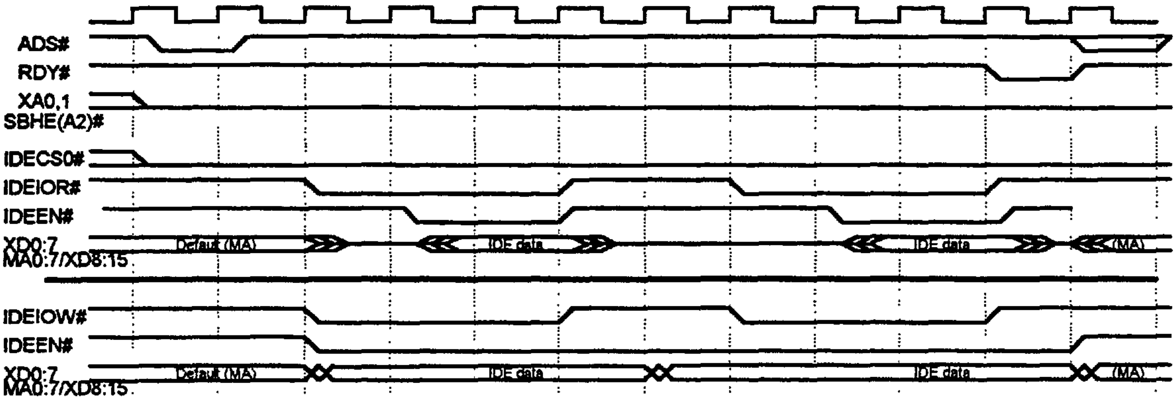


Figure 5.16: Basic Fast IDE Timing.

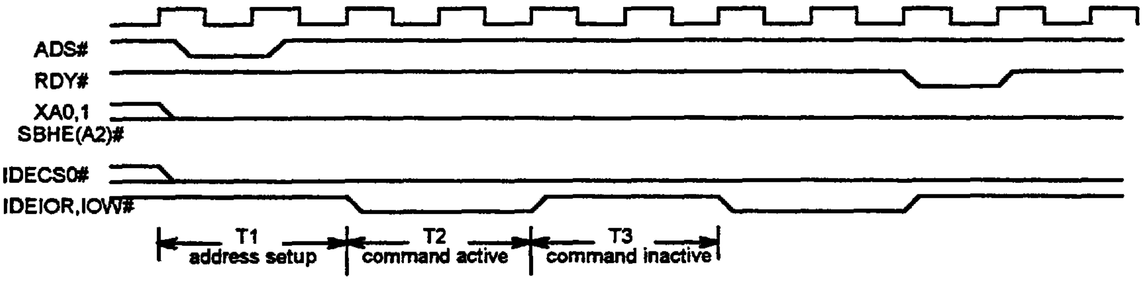


Figure 5.17: Fast IDE Programmable Parameters

Table 5.27: Fast IDE Register Set

Index	D7	D6	D5	D4	D3	D2	D1	D0
40								
41	forceDefault	UseSBHE	datahold	tstatestart	ideen76	ideen54	ideen32	ideen10
42	drv3sped1	drv3sped0	drv2sped1	drv2sped0	drv1sped1	drv1sped0	drv0sped1	drv0sped0
43	–	drv7sped	–	drv6sped	–	drv5sped	–	drv4sped
44	wrA3	wrA2	wrA1	wrA0	rdA3	rdA2	rdA1	rdA0
45			asuA0	asuA0	inacA3	inacA2	inacA1	inacA0
46	wrB3	wrB2	wrB1	wrB0	rdB3	rdB2	rdB1	rdB0
47			asuB0	asuB0	inacB3	inacB2	inacB1	inacB0

Table 5.28: Drive Speed Selection

Timing Selection	Drives 0,1,2,3 speed1:0 bits	Drives 4,5,6,7 speed bit
Timing A	00	0
Timing B	01	1
Reserved (timing C)	10	-
ISA access	11	-

Table 5.29: Register Settings for IDE Timing

Programmed Value	read pulse active (clocks)	Write pulse active (clocks)	Command inactive (clocks)	Address setup (clocks)
0	16	16	24	4
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	-
5	5	5	5	-
6	6	6	6	-
7	7	7	7	-
8	8	8	8	-
9	9	9	10	-
A	10	10	12	-
B	11	11	14	-
C	12	12	16	-
D	13	13	18	-
E	14	14	20	-
F	15	15	22	-

Table 5.30: Programming for Standard IDE Modes

		Mode 0				Mode 1				Mode 2				Mode 3			
	margin	nS	33	40	50	nS	33	40	50	nS	33	40	50	nS	33	40	50
clock nS			30	25	20		30	25	20		30	25	20		30	25	20
Address setup	10	70	3	4	4	50	2	3	3	30	2	2	2	30	2	2	2
Command Pulse	5	165	6	7	9	125	5	6	7	100	4	5	6	80	3	4	5
Cycle time		600				383				240				-			
(Command high)	0	435	15	18	22	258	9	11	13	140	5	6	7	70	3	3	4
Write Data Hold	0	30	1	2	2	20	1	1	1	15	1	1	1	10	1	1	1

5.12.3. Software Considerations

Ideally the fast IDE driver should be a standard part of the BIOS. The default programming should set up the timing modes for the "Mode 0" above in both Timing A and B. Set drive 0 up for Timing A and all others to Timing B.

It is possible for the IDE drives to be connected to either the 4041's fast IDE, to a 3rd party VL bus controller, or to the ISA bus. For IDE connector, the 4041 IDE enable bit should be initially disabled. The software should then look for an IDE drive. If one is present, the drive is hooked to either the VL or ISA busses. It doesn't matter which. If it is not present the fast IDE enable bit should be turned on and the drive checked for again.

Newer drives can be queried as to their speed and capacity (as well as other information). This should be used as an auto setup, and the timing information taken from there. Generally there will be two connectors installed. For this mode, the best setup is as follows:

Drive 0: Timing Mode A
Drive 1: Timing Mode B
Drive 2: ISA Timing
Drive 3: ISA Timing

Drives 2 and 3, if used, are generally CD ROMs or tape drives.

Table 5.30.1: Typical Timing Parameters

Index	Reg Name	Value	Notes
40			
41	Control, Enables	C3	4 drive system (2 connectors)
42	Drive 3:0 timing	F4	Drv 3,2, ISA Drv 1: Timing B, Drv 0: Timing A
43	Drive 7:4 timing	0F	Drives 7,6,5,4: Timing B
44	Timing A pulse width	See table above	Set to Drive 0 mode
45	Timing A high time, add setup	See table above	Set to Drive 0 mode
46	Timing B pulse width	See table above	Set to Drive 1 mode
47	Timing B high time, add setup	See table above	Set to Drive 1 mode

The user should be given the choice of setting timing A and B to Mode 0, 1, 2, or 3 or custom, and the choice of which drive gets which timing mode.

The minimum start bit should be able to be set to a 0 always.

The write data hold time should be set as follows (see the timing table above):

33MHz:	Always 0
40 & 50MHz:	If any Mode 0 timing drive is attached and is using Timing A or Timing B, set to 1, else 0.

5.13. Multifunction Pins

Several pins on the 4041 may take on multiple functions. This section describes each function and which pins may be programmed for what functions.

5.13.1. CPU Functions

WB / WT#. Output. This pin is used by CPUs with L1 write back caches. It is sampled at the same time as the first RDY# or BRDY# of a CPU cache line fill. It is stored in the TAG information in the CPU for future write cycles to that cache line. WB / WT# must be valid at the end of the first T2 for 0 wait state L2 cache hits. The default state is high (Write Back). It may be forced low by either Programmable Memory decode being set to write through mode, or by the BIOS area being set to Write Through.

WPROT#. Output. Write Protect. This is an input for some future CPUs. When this signal is sampled low with the first RDY# or BRDY# of a cache line fill, the line is marked as Write Protected in the tag. The 4041 drives this signal asynchronously with the Write Protect decode. It must be asynchronous to be valid by the end of the first T2. It is driven for all memory cycles, regardless of read or write (the CPU samples it on writes).

5.13.2. Cache and DRAM Functions

CACHECS. Output. Cache Chip Select. Active high. Used to power-down the cache RAMs between cycles. The generation of this signal is described in the cache controller section.

MEMCS#. Output. Memory Chip Select. This signal controls the optional data bus buffers for the DRAMs. The generation of this signal is described in the DRAM controller section.

MA12. DRAM address bit A12. Used for 16M deep DRAMs with 13/11 addressing (16Mx4).

5.13.3. VL-Bus Functions

LDEV1#. Input.
LDEV2#. Input. Additional LDEV# inputs. Logically ORed internally with LDEV0#, which is a dedicated pin. An external OR function (physical AND gate) may be used instead to perform the same function.

5.13.4. SMM and Power Management Functions

EXT0. Input. External Event 0 for the power management logic. Used in various ways to cause system events, etc. May also generate an SMI. See the Power management section for details.
EXT1. Input. External Event 1. See above.
CLKSPEED. Clock Speed select. This pin goes high to select the slow clock. Used for an external synthesizer.
14.3MHz. 14.31818MHz input. May optionally be used for the power management time base.

5.13.5. Chip Selects and I/Os, & Misc

GPIO. I/O. General Purpose I/O bits. These may be an input or an output, controlled individually.
IOCS0#. I/O chip select 0. Output of the Programmable I/O decode 0. May be a chip select or a strobe, and active for reads and/or writes. See the Programmable I/O decode section for details.
IOCS1#. I/O chip select 1.
MEMCS0#. Memory chip select 0. Active when Programmable Memory decode 0 is active. It is a chip select only, and is active for both read and write cycles. It will be valid regardless of the destination bus of the cycle.
MEMCS1#. Memory chip select 1. See above.
IRQ12. Mouse interrupt. Normally sent across the link, it may be an output pin directly.
KBINHIBIT. 8042 keyboard controller inhibit (keylock). When low, the 8042 will not accept keyboard input.
IOCSIPC#. I/O Chip select for the SIPC. This is a straight decode of A10:15. The pin is low when all are low. Used as the upper address decode for the 4045 SIPC.
ISAEN. AT bus enable. Active when an ISA bus cycle is active, or when DGNT# is low (indicating that the a DMA or ISA master is in progress). This pin is high if either ATEN is active or DGNT# is active. This pin is active high since the power up state of the multifunction pin is high, and the buffers must be enabled at power up to read the ROM.
XDIO#. XD bus I/O. When this pin is low, I/O cycles will be directed to the XD bus instead of the SD bus. It has no effect for memory cycles or VL cycles. It is intended for future use for on-board devices residing on the XD bus.

5.13.6. Pin Selection

Table 5.31: Multifunction Pin Function Programming

Pin Number		Function 1 (00)	Function 2 (01)	Function 3 (10)	Function 4 (11)
LDEV1#	In only	GPin	LDEV1#	14MHz	(Reserved)
LDEV2$	In only	GPin	LDEV2#	KBINHIBIT	EXT1
GPA	In only	GPin	14MHz (I)	XDIO#	EXT0
GPB	Out only	WB / WT#	ISAEN	IRQ12	GPout
GPC	Out only	WPROT#	CACHECS#	IOCS0#	GPout
GPD	Out only	DRAMCS#	ISAEN	IOCS1#	GPout
RAS6#	Out only	RAS6#	-	MEMCS0#	GPout
RAS7#	Out only	RAS7#	IOCSIPC#	MEMCS1#	GPout
FLUSH#	Out only	FLUSH# (O)	(Reserved)	IOCS0#	GPout
STPCLK#	Out only	STPCLK# (O)	CLKSPEED (O)		
MCLK	OC or in	MCLK (OC)	GPin		
MDATA	OC or in	MDATA (OC)	GPin		

Table 5.32: Multifunction Pin Programming Registers

Function	D7	D6	D5	D4	D3	D2	D1	D0
GP select	gpBsel1	gpBsel0	GPAsel1	gpAsel0	ldev1sel1	ldev1sel0	ldev0sel1	ldev0sel0
GP select	gpDsel1	gpDsel0	gpCsel1	gpCsel0	ras6sel1	ras7sel0	ras6sel1	ras6sel0
GPout data	flushsel1	flushsel0	FLUSH#	GPD	GPC	GPB	RAS7#	RAS6#
GPin data	(Reserved)		MDATA	MCLK	-	GPA	LDEV1#	LDEV0#

The two "GP select" registers select the function of 8 of the multifunction pins. Each is a 2 bit field. An additional 2 bits in the "GPout data" select the function of the FLUSH# pin. The STPCLK# function is selected in the power management section. The MCLK and MDATA functions are selected by the mouse enable bit, and the HITM# function is controlled in the CPU register.

The "GPout data" supplies the output bits when a multifunction pin is programmed as a general purpose output. Bits 7 & 6 select the FLUSH function.

The "GPin data" reads the current level of the pins listed. These pins may be read regardless of the function selected for that pin.

5.14. Power Management

Power management consists of the following features:

Ability to select the speed of the system clock under software or hardware control
Ability to restart I/O instructions.

Ability to detect inactivity in the system using two activity timers, and then:
- Generate an SMI (either timer)
- Switch to the slow clock (Timer A)
- Stop the CPU clock (Timer B)

Ability to detect a HALT cycle in the system and either:
- Generate an SMI
- Stop the CPU clock.

Ability to detect a wake up event and:
- Generate an SMI (either event).
- Switch to full speed mode (Wake A event)
- Restart the clock (Wake B event).

Ability to have a separate memory area for SMM code and data.
Ability to access all system resources from within SMM.

Ability to redirect CPU soft resets to SMM.

5.14.1. Power Management Techniques

CS4041 supports control of system power through the following chipset output signals:

CPU clock. The CPU clock can be slowed or stopped, either by software command (Index 8Eh) or automatically due to system inactivity (see Timers A and B below). In addition, execution of a HALT instruction can cause the CPU clock to be stopped, either unconditionally or only if running slow (Index 8Dh). While running slow, the clock can switch back to full speed automatically via a "Wake A" event (Indexes 82h and 83h) or by software command (Index 8Eh). While stopped, the clock can start running again automatically in response to a "Wake B" event (Index 85h). Timer A and Wake A are associated with clock slowing, while Timer B and Wake B are associated with clock stopping and restarting. Index 8Ch determines the CPU clock speed for slow mode. CPU clock slowing or stopping can be done instantaneously or using #STPCLK protocol with the CPU, or using the STPCLK# output to control an external clock generator chip. The clock switching protocol is selected via Index 8Dh.

CPU SMI pin. Any one of a wide range of selectable hardware events can cause a System Management Interrupt (SMI). SMM software (System Management Mode) can then decide what further action to take, if any. An SMI can also be triggered directly by a software command (Index 8Eh). Indexes 92h and 93h determine which events can cause an SMI. Indexes 90h and 91h indicate which enabled SMI events are currently pending. Software can selectively clear pending event indicators by writing to Indexes 90h and 91h. One possible SMI event is I/O restart. Index 95h determines which I/O ranges cause an I/O restart SMI. Index 94h allows SMM operation to be adjusted according to the CPU type. As explained in Section 5.6, the 4041 can set aside DRAM space specifically for SMM (Indexes 1Ch and 1Dh). The SMIACT# / SMIADS# signal from the CPU is used to ascertain whether or not SMM is currently in effect.

General Purpose Outputs. A total of six output pins are potentially usable as general purpose programmable outputs, under direct software control. Depending on the specific system objectives, some or all of these signals could be used to switch power on or off to various parts of the system. The pins available for general purpose outputs are GPB, GPC, GPD, RAS6#, RAS7#, and FLUSH#. All six pins also have other possible functions, as programmed by Indexes 3Ch and 3Dh (also 3Eh bits 7:6). When used as general purpose outputs, Index 3Eh bits 0:5 define the output pin states.

Note: All six general purpose outputs default to the high state immediately following system reset (SYSRESET#), but will not necessarily remain high unless appropriately programmed by the BIOS. Several of the signals default to DRAM related functions and should be programmed by the BIOS as needed prior to performing the first DRAM access. GPC defaults to WPROT#, which will go low with the first ROM access in the 0Fxxxxh range. If GPC needs to remain high in the intended system implementation, BIOS should reprogram it as IOCS0# before executing the initial far jump to the 0Fxxxxh area. This can be accomplished either by placing the necessary instructions in the FFFFFFFxh area prior to the far jump, or by executing a NEAR jump to the GPC instructions, followed by the far jump.

CS4041 also supports system power management through the following input signals:

EXT0 input. This is a multifunction pin (GPA) whose function is controlled by Index 3Ch. If used as EXT0, this input can be programmed to cause an SMI and/or affect CPU clock speed. EXT0 can be edge or level triggered (Index 8Fh). Alternatively, this pin can be used as a general-purpose input, directly readable by software via Index 3Fh.

EXT1 input. This is a multifunction pin (LDEV2#) whose function is controlled by Index 3Ch. If used as EXT1, this input can be programmed to cause an SMI and/or affect stopping and restarting of the CPU clock. EXT1 can be edge or level triggered (Index 8Fh). Alternatively, this pin can be used as a general-purpose input, directly readable by software via Index 3Fh.

Other General Purpose inputs. LDEV1, MCLK, and MDATA can all be used as general-purpose inputs, directly readable by software via Index 3Fh. These are all multifunction pins whose functions are selected by Index 3Ch.

The 4041 provides two programmable timers that can be used to detect system inactivity and either generate an SMI or automatically slow or stop the CPU clock. The timers are clocked at a base rate determined by Index 8Ch bits 0:3, intended to be set as close to 1 µs as possible.

Timer A can detect system inactivity and automatically switch the CPU clock to slow mode and/or generate an SMI. The timer is initialized by software and can be re-initialized to the same timeout value by any of a large set of possible hardware events as selected by Indexes 80h and 81h ("Event A"). If none of the enabled events occur within the programmed timeout interval, Timer A times out and causes an SMI and/or CPU clock speed reduction. Another set of possible events, "Wake A" (Indexes 82h and 83h), can automatically restore the CPU clock to full speed and/or generate an SMI. The timeout interval is programmed via Indexes 88h (prescaler) and 89h. For highest resolution in Index 89h, the prescaler should be set to the fastest rate that allows the desired timeout interval to be programmed in Index 89h.

Timer B can detect system inactivity and automatically stop the CPU clock or cause an SMI. The timer is initialized by software and can be re-initialized to the same timeout value by any of a large number of possible hardware events as selected by Index 84h ("Event B"). If none of the enabled events occur within the programmed timeout interval, Timer B times out and causes an SMI or CPU clock stop. Another set of possible events, "Wake B" (Index 85h), can automatically restart the CPU clock to whatever speed was last selected, and optionally cause an SMI as well. The timeout interval is programmed via Indexes 8Ah (prescaler) and 8Bh. For highest resolution in Index 8Bh, the prescaler should be set to the fastest rate that allows the desired timeout interval to be programmed in Index 8Bh.

Either timer can be used as either a system inactivity detector or periodic SMI generator or both. Whenever a timer times out, it automatically restarts at the programmed value and continues counting. Each timer will produce timeout events repetitively until disabled by software via Index 88h (Timer A) or 8Ah (Timer B). One possible usage of the two timers is to let Timer A detect long-term system inactivity (e.g., 15 minutes) and let Timer B assist in stopping the CPU clock again after each IRQ0 Timer Tick interrupt (usually 18 times per second, with 1 ms needed by software for interrupt processing). In that usage of Timer B, some kind of power management software, either SMI-based or non-SMI, would need to disable Timer B in response to new system activity other than Timer Tick interrupts.

5.14.1.1. Power Management for Non-SMI CPUs

For non-SMI CPUs, the power management is done in hardware. This basically consists of running at full speed or reduced speed, and possibly stopping the CPU clock. Initialization software would setup the following:

Activity timer lengths (amount of idle time before switching to the slower clock).
What events reset each activity timer.
What events resume to full power.
What events restart the clock when stopped.
Frequency of the slow clock.
Whether to stop the CPU clock on a HALT.

The hardware will switch to the slower clock when the timer times out, indicating an idle system.
The hardware will switch back to full speed when a wake up event occurs.
While the slow clock is running the hardware may optionally stop the CPU clock when a HALT is executed.

Typical events which reset the activity timer:
Keyboard I/O port access
Floppy or hard disk I/O port access.
Any interrupt acknowledge other than the system timer.
Any alternate master or DMA cycle.

Typical wake up events:
Any interrupt acknowledge other than the system timer.

A multifunction pin, STPCLK#, may be programmed to indicate the speed mode (full or slow). This may be used to switch an external clock generator PLL instead of using the internal clock divider. Since the PLL can switch slowly, this will allow non-SMI CPUs with internal PLLs to be frequency switched.

```
Algorithm:
          Go to slow speed:
                    TimerA times out (no system events occur)
          Stop Clock
                    CPU executes a HALT while in slow speed mode.
          Restart clock
                    Any INTR or NMI or external source
          Return to full speed:
                    Wake up event (INTA other than IRQ0, push button, NMI, master/DMA access)
```

Programmable options:

TimerA:

Interval
What resets it

Timer B:

Interval
What resets it

Stop clock options

Do not stop clock
Pull STPCLK pin
Stop the actual CPU clock
Pull STPCLK pin and stop the CPU clock.

5.14.1.2. Power Management for SMI CPUs

For SMI capable CPUs, greater control of power management is possible. This includes multiple levels of power control, better control of stopping the CPU clock, and the ability to shut down other devices, including video, under software control.

The activity timer causes an SMI when the system is idle. The SMM code then chooses the appropriate clock frequency to switch to. It may also put other devices in a low power mode, if desired. It may then setup the activity timer to trigger at a different interval, or a different set of events to reset it to go into a further low power mode.

A wake up event also causes an SMI. The SMM code then picks the level of system performance according to the event which caused the wake up.

If the SMM code stops the CPU clock (with the STPCLK function) a wake up event, including an INTR will restart the clock and issue the SMI.

5.14.2. SMI Sources

The SMI sources are as follows:

TimerA time-out
TimerB time-out
Wake up EventA
Wake up EventB
I/O Restart
Halt bus cycle
CPU restart request.
External SMI source (either of two pins)
Software SMI

Two index registers indicate which sources have requested an SMI. The MSB of register A indicates whether any bits in register B are set. Multiple bits may be set if several sources happen at once. Bits may continue to be set while in SMM. Writing a 1 to the register at any bit position (other than the MSB of register A) will clear that bit.

Two more index registers enable the sources of SMI interrupts. The bit encoding is the same as that of the SMI status registers except that the MSB of register A is a global enable for SMI. If SMI# is still low upon exiting from SMM, the global enable should be toggled before exiting SMM to guarantee that the SMI# pin toggles. This need not be done if index 90h reads as 00, since this indicates that the SMI# pin is inactive (high).

Table 5.33: SMM Status and Enable Registers

Function	D7	D6	D5	D4	D3	D2	D1	D0
SMI status A	next register	-	Halt	I/O Restart	WakeB	WakeA	TimerB	TimerA
SMI status B	-	-	-	-	External1	External0	CPU restart	Software smi
SMI enable A	global enable	-	Halt	I/O Restart	WakeB	WakeA	TimerB	TimerA
SMI Enable B	SMI Mode	-	-	-	External1	External0	CPU restart	Software smi

5.14.3. SMI Timing Modes

The 4041 has two timing modes for SMI#: Intel and Cyrix.

In Intel mode, SMI# is always an output, and is the OR of the sources.

In Cyrix mode, it is an I/O pin. The Cyrix CPU begins driving SMI# low after detecting it low, then drives it back high again at the end of the SMM routine. The 4041 algorithm in Cyrix mode is:

Normally drive SMI# high.
To signal an SMI, drive it low for 4 clock cycles, then float it.
On the clock edge where it is detected high, begin driving it high.
It may be driven back low again in the following clock cycle.

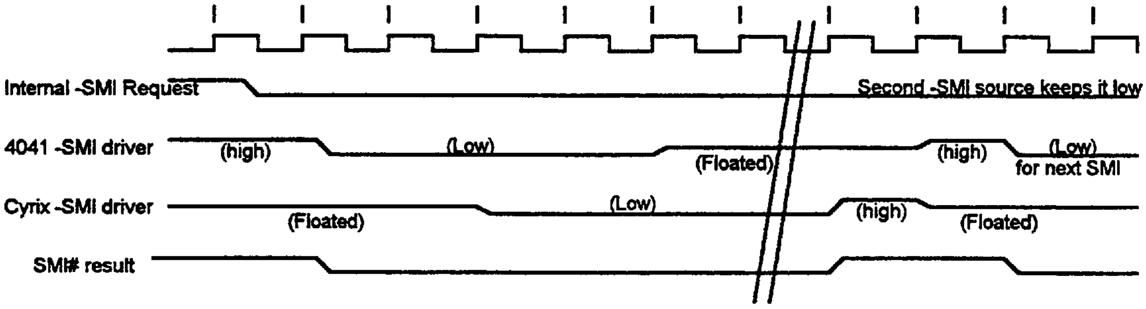


Figure 5.18: Cyrix SMI# Timing

5.14.4. Activity Monitor Timers

There are two activity monitor timers. Each is an 8 bit counter which is restarted by the occurrence of any selected event. EventA restarts TimerA, while EventB restarts TimerB. When the timer times out, it will either perform a hardware function or cause an SMI. It will then automatically restart at the programmed value. It will perform its function (generate an SMI or change the CPU clock) each time it times out.

Table 5.34: Activity Timer Selectable Functions

Timer	Purpose	Hardware Function	Software Function
TimerA	Detect long term inactivity	Switch to slow clock mode	Generate an SMI
TimerB	Detect short term inactivity or Periodic SMI Timer	Stop the CPU clock	Generate an SMI

TimerA is the main activity timer, used to go into "Green" mode. TimerB has several potential uses. One is for a shorter period time-out for going in and out of Stop Clock mode between IRQ0 interrupts. Another is to make a periodic SMI timer.

Time Base Selection (Index 8Ch). An approximately 1MHz clock is generated internally from the CLKIN pin, which is first divided by 2 if 2x clock mode is selected. A 14.3 MHz clock source may also be used, which is provided on either of two multifunction pins. Index 8Ch bit 3 selects between CLKIN (0) or 14 MHz (1). Index 8C bits 2:0 should be set as shown in Table 5.34.1.

Table 5.34.1: Time Base Selection

8Ch Bit 3	CPU Frequency	8Ch Bits 2:0	Divider	Resulting Period
0	25 MHz	000	24	0.960 μs
0	33.3 MHz	001	32	0.960 μs
0	40 MHz	010	40	1.000 μs
0	50 MHz	011	48	0.960 μs
1	14.31818 MHz Input*	100	14	0.978 μs*

* The external frequency applied to the 14.3 MHz input pin need not be 14.3 MHz. For other frequencies, the divider should be selected to achieve a period as close to 1 μs as possible.

After selecting the time base, the slow clock divider should be set to produce the desired slow clock frequency. Since many CPUs will not work below 8 MHz, this will be the most common mode. Suggested dividers are listed in Table 5.35.

Table 5.35: CPU Slow Clock Programming

CPU Frequency	8Ch Bits 6:4	Divider	Resulting Slow Clock
25 MHz	001	2	12.5 MHz
33.3 MHz	010	4	8.33 MHz
40 MHz	010	4	10 MHz
50 MHz	011	6	8.33 MHz

In Tables 5.34.1 and 5.35, "CPU Frequency" refers to the CLKIN frequency in 1X clock mode, or CLKIN/2 in 2X mode. The register settings are not affected by the 1X/2X clock mode.

Prescalers. The 1μs (approx.) time base is further divided by 64, then by 16 several times to produce the periods below. Each timer may independently select one of these clock periods.

Table 5.36: SMM Timer Resolutions and Max Time-Outs

Mode	Divider	Resolution (approx.).	Max time-out (approx.)
0	off	--	(Timer disabled -- no timeout)
1	64	64uS	16mS (test mode)
2	1024	1mS	256mS
3	16,384	16mS	4 seconds
4	262,144	256mS	64 seconds
5	4,194,304	4 seconds	1024 seconds (17 minutes)
6	67,108,864	64 seconds	16384 seconds (4.6 hours)

When an "Event" occurs the timer restarts to it's initial programmed value. "EventA" resets TimerA, while "EventB" resets TimerB. The values may be read at any time by reading the 8 bit register. The initial value may be read by turning off the timer, which is done by setting the count rate to 0. This both stops the clock and resets the value. When reading the count while the timer is active, it is possible to read it while it is de-incrementing. Reading it multiple times until the same value is read twice in a row is advisable to guarantee an accurate value.

The timers and associated dividers are clocked approximately once every 16 uS if the proper time base dividers are set. Loading the timers with a new value also occurs only on these clock edges. With the synchronization required, after writing a new timer value it will take between 8 and 24 uS before the new value is loaded into the timer. Reading the timer before this occurs will give the old value. It is also possible for the timer to time out and cause an SMI or clock slowdown up to 24 uS after the timer is reset or an event occurs.

The timers are reset on any combination of the following:

INTR or NMI signal to the CPU.
Interrupts (IRQ0, IRQ1, IRQ3,4,5,7, IRQ6,14, IRQ9,13, IRQ,8,10,11,12,15)
I/O port accesses (Floppy, IDE1, COM1&2, COM3&4, LPT1,2,3, KB, programmable1,2)
Video Memory access.
HLDA going high (alternate master active).
An external pin going active.

See the Events Detection section below on how these events are actually detected, and what I/O ports, etc. are used.

5.14.5. Wake Up Events

There are two programmable wake up events. WakeA is normally used to switch back to full speed mode. WakeB is normally used to start the CPU clock after it has been stopped. Each may also generate an SMI.

5.14.6. Events Detection

There are 4 "Events" which do various functions. Two of them reset the activity timers while the other two wake up the system from slow or stopped clock. The functions are:

Table 5.37: Event Functions

Event Name	Function	Additional Function
EventA	Reset TimerA	
EventB	Reset TimerB	
Wakeup A	Switch back to full speed	Generate an SMI
Wakeup B	Restart the CPU clock if stopped	Generate an SMI

EventA and Wakeup A have individual control for almost every possible event. They are programmable separately for maximum flexibility. EventB and Wakeup B require far less flexibility. They provide only a few possible key event selections, but may also select EventA or Wakeup A as one of their sources, effectively taking the "A" selection and adding to it.

EventB will generally use EventA, and add IRQ0 to it. If used, it generally times the active period of the timer tick interrupt.

Wakeup B restarts the CPU clock. INTR, NMI, and external events are all that are generally required. A stopped clock CPU is not able to access any I/O ports or generate any INTA cycles (for individual IRQ detection by the 4041) or assert HLDA.

The following explains how each event is detected.

<u>INTR.</u> INTR is generated by the 4045, but it is monitored by the 4041 on a dedicated 4041 input pin. In the 4041, INTR is used solely for event detection. It is generally used as a Wakeup B event, which re-enables a stopped clock. When this pin is high it causes an event, and continues to do so as long as it is high.

NMI. NMI is generated by the 4041. It causes an event on a low-to-high edge.

IRQs. The IRQ pins do not enter the 4041. Instead, the 4041 detects specific IRQs via the interrupt acknowledge cycle. This simplifies the software greatly, since it does not have to know which interrupts are enabled in the 8259 logic in the 4045. Interrupt acknowledges are generated by the 4041 and sent across the link to the 4045. The 4045 returns the vector on the second INTA. The event logic will look at the second INTA (where A2 is 0) and pick up the vector from XD0:7 when INTA# goes back high. The vectors returned by the two 8259s are programmable, although DOS systems always program them the same way (IRQ0:7 produce vectors 08:0Fh and IRQ8:15 produce vectors 70:77h). The only thing this logic must do when it receives a vector is figure out which interrupt controller it came from. The location of the second interrupt controller is automatically captured in register 87h. The 4041 compares this to the upper 5 bits of the vector. If it is equal it is IRQ8:15, otherwise it is IRQ0:7. Normally the value is programmed to 0111 0xxx when running under DOS, but the value automatically tracks the value written into the interrupt controller. This allows the value in 87h always to match the value currently being used by software. Windows, for example, changes the interrupt vector base in INTC2 to 58h, then changes it back to 70h upon return to DOS.

Alternate Master. This event is detected by HLDA going high. This will occur when any alternate master owns the bus, including VL Masters, ISA Masters, and DMA cycles. It does not go active for refresh cycles.

External Pins. Two external pins may be used to cause events. These are multifunction pins, which may be used for many functions. The are programmable as active high or low, and level or edge detected to cause an event. Two bits program the mode for each.

Table 5.38: External Pin Event Modes In Index Register 8Fh

External Pin Mode	Function	Event Duration
00	Active Low level Triggered.	Event Active while low
01	Active High level Triggered.	Event Active while high
10	High to low edge Triggered.	Event pulse on high to low edge
11	Low to high edge Triggered	Event pulse on low to high edge

EventA. EventA causes Timer A to restart. EventA is the final OR of all of the individual events which are currently enabled. It may be used as an event input for EventB. This would mean that EventB would occur anytime EventA occurs PLUS any additional events selected for EventB (the individual selections for EventB are more limited than for EventA).

I/O port Events. There are many standard peripherals at known locations. These are decoded with fixed decodes. Some of these are enabled in groups. For instance all of the COM and Parallel ports are enabled together, while a separate register determines which addresses to look at. The addresses are given in the tables below. The two programmable I/O port ranges may also be selected as an event. See the Programmable I/O decode section for details on programming this.

Memory Access Events. Memory accesses are less common as system events. The CS4041 allows accesses to VGA memory to be system events. Within the VGA event, the A0000 and/or B0000 area may be selected.

Table 5.39: Event Fixed I/O Address Ranges

Function	I/O Address Range
Floppy Disk	3F0:3F1, 3F3:3F5*
IDE Controller(s)	See IDE table
COM Ports	See COM table
Parallel Ports	See Parallel Port Table
Keyboard Controller	60 and 64
VGA	3B0:3BB, 3C0:3DF

* I/O port 3F2h is not included for the floppy because it is accessed often by the timer tick interrupt to shut off the motor. Including it would falsely indicate floppy activity.

Table 5.40: Selectable Ranges for Specific Events

IDE Controller Events		COM Ports Events		Parallel Ports Events	
Function	I/O Address	Function	I/O Address	Function	I/O Address
IDE1	1F0:1F7, 3F6	COM1	3F8:3FF	LPT1	3BC:3BE
IDE2	170:177, 376	COM2	2F8:2FF	LPT2	378:37F
IDE3		COM3	3E8:3EF	LPT3	278:27F
IDE4		COM4	2E8:2EF		

Table 5.41: Power Management Configuration Registers

index	function	D7	D6	D5	D4	D3	D2	D1	D0
80	EventA	NMI	INTR	IRQ8,10:12,15	IRQ9,13	IRQ6,14	irq3,4,5,7	IRQ1	IRQ0
81	EventA	prog IO 1	prog IO 0	master	Ext0	video mem	kb	com/lpt	disk
82	WakeupA	NMI	INTR	IRQ8,10:12,15	IRQ9,13	IRQ6,14	irq3,4,5,7	IRQ1	IRQ0
83	WakeupA	prog IO 1	prog IO 0	master	Ext0	video mem	kb	com/lpt	disk
84	EventB	EventA	prog IO 0		keyboard	video mem	INTR	Ext 1	IRQ0
85	WakeupB	wakeA event	Ext 1	master		SMI	NMI	INTR	-
86	select which		IDE2	IDE1	floppy		lpt1,2,3	com3&4	com1&2
87	INTA base	intabase7	intabase6	intabase5	intabase4	intabase3	-	-	-
88	timerA cntrl	WakeAfast			TimerASlow		rateA2	rateA1	rateA0
89	timerA	tmrA7	tmrA6	tmrA5	tmrA4	tmrA3	tmrA2	tmrA1	tmrA0
8A	timerB cntl				Stop clk		rateB2	rateB1	rateB0
8B	timerB	tmrB7	tmrB6	tmrB5	tmrB4	tmrB3	tmrB2	tmrB1	tmrB0
8C	Time Base	stpclk# pin	slowclk2	slowclk1	sloclk0	cksel	div2	div1	div0
8D	Stop Clock	waitstopack	plldelay2	plldelay1	plldelay0	stpclk4slow	stopmode	StopHaltnslo	StopOnHalt
8E	Commands					GenSMI	Stpclk	GoFast	GoSlow
8F	EXT pins					EXT1 1	EXT1 0	EXT0 1	EXT0 0
90	SMI statusA	next register	-	Halt	I/O Restart	WakeupB	WakeupA	TimerB	TimerA
91	SMI statusB	-	-	-	-	External1	External0	CPU restart	Software smi
92	smi enableA	global enable		Halt	I/O Restart	WakeupB	WakeupA	TimerB	TimerA
93	smi EnableB	smi mode				External1	External0	CPU restart	Software smi
94	smm modes				SoftResRedir	DisKen	ForceA20m	flushOnSmm	smiact pin
95	I/O restart	prog IO 1	prog IO 0	KB	VGA	floppy/HD	LPT(378)	com2	com1
96	Shadow70	port70d7	port70d6	port70d5	port70d4	port70d3	port70d2	port70d1	port70d0

5.14.7. I/O Restart

I/O restart allows CPUs with SMM capability to trap specific I/O operations and later restart them. This may be used to power-down peripherals, then re-initialize them when an access attempt is made. It may also be used to emulate a device, but this is a less common usage.

I/O cycles are detected on the local bus. If a cycle is to be restarted, it does not generate an ISA cycle. Instead, the internal LDEV# is pulled, and SMI# is issued. RDY# is returned to the CPU 4 SCLKs after SMI# is generated.

Several fixed I/O ranges may be selected to issue restarts as well as the two programmable ranges. The fixed ranges are:

COM1or COM2 (3F8:3FF, 2F8:2FF)
Parallel port at 378:37F
Floppy and Hard disk controllers (1F0:1F7 & 3F0:3F7, 170:177 & 370:377)
The VGA I/O (3B0:3BB, 3C0:3DF)
The keyboard. (60 & 64)
The two programmable ranges.

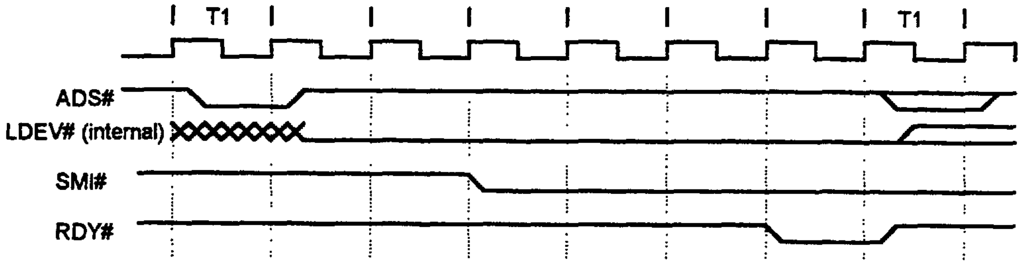


Figure 5.19: I/O Restart Timing

5.14.8. Power Management Clock Changing

There are many CPU types and modes for slowing and stopping the clock. Consequently, there are many ways for this CHIPSet to slow down and stop the clock. The CS4041 is fairly flexible with the switching and stopping, allowing individual functions to be selected based on the CPU and mode being used, rather than setting the CPU type and mode in a register. An endless variation of CPUs and modes seems inevitable. The individual functions will be described, followed by what CPUs and CPU modes require what.

When to Slow Down the Clock

TimerA	(Hardware Power Management)
Software command	(Software Power Management)

Either method may be used to slow the clock down. A TimerA time-out is optional, and normally used with a non-SMM CPU, or when SMM is not to be used for this function. TimerA may be programmed to time-out after a specified amount of system inactivity. The Software Command would normally be used from within SMM, but is available at any time.

When to Speed Up the Clock

WakeA	(Hardware Power Management)
Software command	(Software Power Management)

WakeA goes active on any selected event. In the software power management mode this would generate an SMI which would switch the clock.

How to slow down the Clock

Internal Divider
Signal to external synthesizer

If the internal divider is used, it is preset to a slow down frequency and the hardware or software command switches to that divider. For standard 486 CPUs, a synthesizer may be used, which will change the clock frequency gradually enough to meet the requirements of the CPU's VCO. The 4041 then simply provides an output pin to switch to slow mode. The internal divider is not used.

When to stop the Clock

Halt bus cycle
Halt bus cycle only while slow clock is running
Software command

The clock may be set to stop when a HALT CPU bus cycle is detected. It may also be set to only do this when the slow clock is running. This would allow a very low power 'green' mode without affecting normal full speed operation. The system would wake up on each timer tick and the clock is then stopped by executing a HALT at the end of the timer tick routine or in higher-level power management routine that originally executed the first HALT. A software command may also be used, but the HALT method will probably be used in SMM mode instead.

When to Restart the Clock

WakeB

The clock can only be restarted by a WakeB event. Normally an interrupt. Since software can't run when the CPU clock is stopped, software obviously cannot restart the clock.

How to Stop the Clock

Pull STPCLK pin only
Pull STPCLK pin and gate off the CPU clock

Two modes are available for clock stoppage. The first is to simply pull the STPCLK pin but leave the clock output running. For an S series CPU this allows the clock to be restarted without having to wait for the VCO to stabilize. The other method is to pull STPCLK and also gate off the clock. This is a lower power mode, but takes longer to restart if the CPU contains a VCO (some do not, and may be restarted immediately). The CPU clock can also be stopped immediately, without STPCLK#, if the CPU can tolerate it. Most CPU types either cannot tolerate a stopped clock at all or require STPCLK# protocol.

Protocol for Changing the Clock Frequency

Assert STPCLK before changing frequency: yes/no
Wait for STPACK before changing the clock: yes/no
Wait for x amount of time before releasing STPCLK: Yes/no & delay

There are three protocol adjustments which can be made for changing the CPU clock frequency. They depend on the requirements of the CPU in use.

The first is whether to assert STPCLK before changing the frequency. This is required for a CPU with a VCO (Intel S series). It is not required for Cyrix or AMD, allowing clock switching to be instantaneous.

The second is whether to wait for STPACK (a special bus cycle) before changing the clock speed. Again, this is required for a CPU with a VCO but not for others. Not all CPUs will give a STPCLK acknowledge. For Cyrix CPUs, the SUSPA# signal may be used to indicate a stop clock acknowledge. This is required in Cyrix CPUs that have a PLL. There is no need to wait for the SUSPA# signal on Cyrix CPUs without a PLL.

The third is how long to wait before releasing STPCLK after changing the frequency. Intel S series requires 1mS. Non-VCO CPUs do not require any time.

Table 5.42: PLL Stabilization Delay Time

Mode	Delay
000	No delay
001	32uS
010	64uS
011	125uS
100	250uS
101	500uS
110	1mS
111	2mS

Standard 486 CPU (SX, DX & DX2)

A standard 486 contains a PLL which requires the clock frequency to be changed slowly while it is operating. There is no STPCLK# function or SMM capabilities. The minimum frequency is 8MHz. Two methods of power control may be used:

Use a clock synthesizer to change frequencies. This requires a synthesizer chip which is capable of switching the clock frequency slowly enough to satisfy the 486. Most do. A multifunction pin is used to control the frequency. The internal clock divider is not used. Hardware clock switching is used.

Reset the 486 while changing frequencies. This method uses the STPCLK# output of the 4041 to reset the 486 while the clock is being switched. STPCLK# is inverted and ORed with CPURESET. The CPU will be reset before switching clock frequencies, and taken out of reset 1mS after the switch. The reset vector must be capable of figuring out that this is the cause of the reset.

Intel S Series

The S series has a STPCLK# function and SMM. It has a PLL, and requires the use of STPCLK#, with a 1mS delay between changing the frequency and removing STPCLK#. Either hardware or software power management may be used. The 4041 should wait for the Stop Clock Acknowledge bus cycle.

If 2x clock mode is used the clock may be changed instantaneously without STPCLK#. It is not recommended to use this because it will not work with DX2s or P24Ts.

AMD DX2. Same as Intel DX2. It has a PLL.

AMD S Series

These are basically the same as Intel, with PLL.

Cyrix S & S2 CPUs

The Cyrix parts have the same basic functionality as the S series. The "PLL" is static, and may have the frequency changed at any time. Future Cyrix CPUs may contain a PLL and require using the SUSPA# signal when switching the clock.

5.15. Internal Keyboard/Mouse Controller

The 4041 contains a state-machine implementation of a PS/2 compatible keyboard/mouse controller, eliminating the need for an external 8042 microcontroller device. The 4041 implementation is entirely state-machine and gating-tree based; there is no internal microcontroller, microcode, or ROM of any kind. Even the translation of keyboard scan codes to standard PS/2 keycodes is done with a gating tree rather than a ROM. As compared to a general-purpose 8042 microcontroller, the 4041 implements the subset of 8042 features needed for a PS/2 compatible keyboard and mouse controller. (The internal controller can be disabled, if desired, to allow an external 8042 or equivalent to be used.)

Host CPU software communicates with the keyboard/mouse controller via two I/O addresses ("host CPU" refers to the main system CPU, 486-family):

64h -- status (read), or host command (write)
60h -- data read/write (8-bit)

Reading from 64h. Controller status can be read at any time from I/O address 64h. The status following system reset is 00h. Controller status bit assignments are as follows:

7	(Not used; read as 0)
6	Timeout during transmission to or from keyboard or mouse
5	Mouse Data Available in Controller Output Buffer
4	Inhibit Switch (KBINH# signal), '1' = high
3	Command/Data (1/0) last written to 64h/60h
2	System Flag (1 usually means a switch from Protected to Real Mode)
1	Controller Input Buffer Full (last write to 60h/64h still pending)
0	Controller Output Buffer Full (data available to read at port 60h)

Controller Input Buffer. The controller has a single 8-bit Input Buffer that receives all writes to either 60h or 64h. Status bit 1 indicates whether or not the previous write is still being processed, and bit 3 indicates whether the write was a host command (64h) or data (60h).

Controller Output Buffer. The controller also has a single 8-bit Output Buffer that holds all data for reading by the host CPU at address 60h. Status bit 0 indicates whether or not there is data available to read. If the host CPU isn't expecting a response to a previous host command, the data read from 60h will be either keyboard data or mouse data, as indicated by status bit 5. Reading from 60h also clears any pending mouse or keyboard interrupt. Address 60h should not be read unless status bit 0 is '1'.

Writing to 60h. I/O address 60h should not be written unless status bit 1 is '0'. If the Controller isn't expecting data associated with a previous host command, data written to I/O address 60h is transmitted to the keyboard as a keyboard command. (Commands can be transmitted to the mouse via host command code D4h, described below.)

Writing to 64h. Host commands should not be written unless status bit 1 is '0'. In addition, host commands that generate a read data response should not be issued until the keyboard and mouse have been disabled and any data already in the Controller Output Buffer has been read. Otherwise, data from the keyboard or mouse could be overwritten by the command response.

Interrupts. IRQ1 can be enabled as the keyboard interrupt, and IRQ12 can be enabled as the mouse interrupt. Both interrupts are latched in the 4045 until cleared by a read from I/O address 60h. Interrupt information is communicated from the 4041 to the 4045 via the Control Link.

System Flag. Status bit 2 has no direct effect on the controller. It is written by the host CPU (via the 60h command described below), usually to indicate the reason for a CPU reset. On 80286-based systems, CPU reset was the only way to return from Protected Mode to Real Mode. 386 and 486 systems remain compatible with this method.

Keyboard Inhibit Input. The LDEV2# pin optionally can be programmed to function as a standard Keyboard Inhibit input (KBINH#) via Index 3Ch, bits 3:2. When the pin is so enabled and driven low, the internal keyboard controller discards any scan codes received from the keyboard. The keyboard is not prevented from continuing to send scan codes, and the host CPU can still send commands to the keyboard, but scan codes received from the keyboard are not given to the host CPU. The keyboard inhibit signal can be overridden by host CPU command 60h (see next section). The state of the KBINH# signal can be read at any time via port 64h. Mouse operation is not affected by KBINH#.

5.15.1. Host CPU Commands

The 4041 internal keyboard/mouse controller supports the host CPU commands listed below. Unless otherwise noted, all commands execute in the same ISA cycle in which they are issued.

20h	Read Command Byte
21-2Fh	Read RAM (one-byte simulated RAM)
60h	Write Command Byte
61-6Fh	Write RAM (one-byte simulated RAM)
A7h	Disable Mouse
A8h	Enable Mouse
A9h	Mouse Interface Test
AAh	Self Test
ABh	Keyboard Interface Test
ADh	Disable Keyboard
AEh	Enable Keyboard

C0h	Read Input Port
D0h	Read Output Port
D1h	Write Output Port
D4h	*Transmit to Mouse
E0h	Read Test Inputs (keyboard and mouse clocks)
F0-FFh	*Pulse Output Port

* D4h and F0-FEh may not finish executing in the same ISA cycle as issued.

(20h) Read Command Byte. The Command Byte is placed into the Controller Output Buffer for reading by the host CPU. For bit definitions, see 60h command below. (The "Command Byte" is an internal byte separate from the host CPU commands written to port 64h.)

(21-2Fh) Read RAM. A one-byte simulated "RAM" is provided for software compatibility. Any "Read RAM" command causes this one byte to be placed into the Controller Output Buffer for reading by the host CPU.

(60h) Write Command Byte. The next byte written to port 60h is placed into the Command Byte. On reset, the byte is set to 70h. Scan code conversion is always enabled. Bit definitions:

7	Reserved, write as 0.
6	Reserved
5	Disable Mouse
4	Disable Keyboard
3	Inhibit Override (ignore KBINH# input)
2	System Flag
1	Enable Mouse Interrupt
0	Enable Keyboard Interrupt

(61-6Fh) Write RAM. The internal keyboard/mouse controller contains a one-byte register that simulates the RAM found in the 8042. A single byte is sufficient for PS/2 compatibility. Any "Write RAM" command writes this one byte.

(A7h) Disable Mouse. This host command sets bit 5 in the Command Byte, causing the mouse clock to be driven continuously low to disable the mouse interface.

(A8h) Enable Mouse. This host command clears bit 5 in the Command Byte, allowing the mouse clock to return to the high state and be pulsed by the mouse if the mouse has new data to transmit.

(A9h) Mouse Interface Test. For PS/2 compatibility, a 00h test result is placed in the Controller Output Buffer for reading by the host CPU. No actual test of the mouse interface is performed.

(AAh) Self Test. For PS/2 compatibility, a 55h test result is placed in the Controller Output Buffer for reading by the host CPU. No actual self-test is performed beyond issuing the 55h result.

(ABh) Keyboard Interface Test. For PS/2 compatibility, a 00h test result is placed in the Controller Output Buffer for reading by the host CPU. No actual test of the keyboard interface is performed.

(ADh) Disable Keyboard. This host command sets bit 4 in the Command Byte, causing the keyboard clock to be driven continuously low to disable the keyboard interface.

(AEh) Enable Keyboard. This host command clears bit 4 in the keyboard/mouse command byte, allowing the keyboard clock to return to the high state and be driven by the keyboard if the keyboard has new data to transmit.

(C0h) Read Input Port. The contents of the Input Port are placed in the Controller Output Buffer for reading by the host CPU. The Input Port is a subset of the P10-P17 Input Port found in an 8042. Bit assignments:

7	Keyboard Inhibit input (LDEV2# pin; see Index 3Ch)
6:2	(Not used; read as 0)
1	Mouse data input
0	Keyboard data input

(D0h) Read Output Port. The contents of the Output Port are placed in the Controller's Output Buffer for reading by the host CPU. The Output Port is a subset of the P20-P27 Output Port found in an 8042. Bit assignments:

7:6	(Not used; read as 0)
5	Mouse interrupt, IRQ12 (latched in the 4045)
4	Keyboard interrupt, IRQ1 (latched in the 4045)
3:2	(Not used; always 0)
1	Gate A20 function, affects A20M- via the 4045
0	CPU Reset request, causes CPU reset (via the 4045) or SMI

If additional general-purpose I/O signals are needed, they can be implemented using GPIO pins on the 4041. See Index Registers 3C-3Fh (multifunction pins).

(D1h) Write Output Port. The next data byte written to port 60h is placed in the Output Port. See D0h command above for bit assignments.

(D4h) Transmit to Mouse. The next data byte written to port 60h is transmitted to the mouse. The execution time of this command depends on mouse timing.

(E0) Read Test Inputs. The Test Inputs are internal equivalents of the T0 and T1 input pins found on an 8042. Bit assignments are as follows:

7:2	(Not used, read as 0)
1	Mouse clock input
0	Keyboard clock input

(F0-FFh) Pulse Output Port. A '0' in bits 0-3 causes the corresponding Output Port bit to be pulsed low for 6 µs minimum. Pulsing bits 2 or 3 has no effect, since these Output Port bits aren't used. The most typical use of this feature is to pulse the CPU reset request line. An FFh command executes immediately (same ISA cycle as host command).

5.15.1.1. Serial Data and Clock

Data transmissions to and from the keyboard and mouse utilize an 11-bit serial transmission format consisting of a start bit, 8 data bits (LSB sent first), parity bit, and stop bit. Parity is odd, meaning that a data value of FFh has a 1 for the parity bit. A bit value of 1 corresponds to a high logic level, and 0 corresponds to low. The start bit is always a 0, and the stop bit is always a 1. The clock and data lines remain high between transmissions except as described below for higher level communication protocol.

Clocking during a transmission is always provided by the keyboard or mouse, not by the controller, regardless of whether the transmission is to or from the controller. When the keyboard and mouse are both idle and enabled for input to the host, the clock and data lines for each device are both high. To inhibit transmission, the clock is held continuously low for as long as transmission is to be inhibited. The clock and data lines have open collector drivers in the controller, keyboard and mouse. External pull-up resistors keep the signals in the logic high state when not being driven low.

The typical sequences of events for data transmission are summarized below.

Input to controller from device:

Device waits for its clock and data lines both to be high.

Device asserts the start bit (data line driven low).

Upon seeing the device's start bit, the controller drives the other device's clock line low to inhibit transmissions from the other device while the first device is transmitting.

The transmitting device pulses its clock line low 11 times, driving the next data bit onto the data line after each rising edge of the clock. Data transitions during input to the controller occur while the clock is high.

After the 11th clock pulse, the controller typically drives the clock low as long as needed to make the input data available for reading by the host CPU.

When the controller is ready to accept new input, the controller releases the clocks to both devices, allowing the clocks to return high.

Output from controller to device:

Controller drives both clock lines low, inhibiting transmissions from both the keyboard and the mouse.

If the controller needs to transmit to the keyboard, the controller holds the mouse clock low continuously throughout the transmission. (Vice versa for transmission to the mouse.)

After both clocks have been low for a period of time, the controller releases the clock to the receiving device. At about the same time, the controller begins driving the start bit (0) onto the data line.

Some time after detecting the controller's start bit, the receiving device pulses the clock line low 11 times. The controller drives the next data bit onto the data line shortly after each falling edge of the clock. Data transitions during output from the controller occur while the clock is low, usually very near the falling edge of the clock.

During the 11th clock pulse (while the clock is low and the stop bit is being sent), the device drives the data line low to signal the "line control" bit to the controller. The device releases the data line after releasing the clock at the end (rising edge) of the 11th clock pulse.

When the controller is ready to accept new input from the other device, it releases the other device's clock line.

Both the keyboard and the mouse use the same protocol, as described above. Clock timing during all transmissions is determined by the keyboard or mouse, typically around 40 us low and 40 us high.

The controller is largely transparent to the content of transmissions to or from the keyboard or mouse. The main processing done by the controller, other than serial-to-parallel conversion (or vice versa), is to translate keyboard scan codes as required by PS/2 standards.

5.15.1.1.1. Additional Considerations

External 8042. The default following reset is for the internal keyboard controller to be enabled (mouse disabled). See Index 39h. This is necessary with certain keyboards in systems that use the internal keyboard controller. These keyboards need the clock line held in the low state following system reset. If an external keyboard controller is used, the clock and data outputs from the 4041 need to be reprogrammed as inputs from the external controller. During the time before the BIOS can enable these pins as inputs, the 4041 will be driving them low at the same time that the external controller is trying to drive them high. To limit the current that can flow during this time, 1K ohm series resistors should be used between the 4041 KBCLK/KBDATA pins and the external controller's GATE A20 and CPU RESET pins.

Emulated Gate A20 and CPU Reset. The emulation option (Index 39h) for Gate A20 and CPU reset is most useful when using an external 8042, to provide a significant performance benefit over normal 8042 processing delays. The emulation options are also usable with the internal keyboard/mouse controller, providing a slight performance advantage over the internal controller. For highest system performance, the emulation option should always be enabled, regardless of whether the internal or external keyboard/mouse controller is being used.

5.16. Manufacturing Test Modes

The 4041 has two test modes to support board-level manufacturing test:

Putting all outputs in a high-impedance state (Hi-Z)
Checking each pin's soldering connection (AND tree)

The same basic procedure is used for activating either of these modes:

Drive SYSRESET high for at least 5 cycles of SCLK. (SCLK must be running or pulsed.)
While SYSRESET is high, drive TEST# low (4041 LIN pin).
While TEST# is low, drive SYSRESET low. Allow at least 3 cycles of SCLK from falling edge of TEST# to falling edge of SYSRESET.
Drive a test code onto XD7:0. Use code 01h for the "Hi-Z" test mode, or 07h for the connectivity test (AND tree).
Keep TEST# low and XD7:0 valid for at least 5 cycles of SCLK after SYSRESET goes low.
Drive TEST# high. The rising edge of TEST# latches the test code into the 4041. XD7:0 should remain valid for at least 20 ns after TEST# is driven high.

In the connectivity test mode, all pins except SYSRESET, TEST#, and ROMCS# become inputs to an AND-gate tree. The final output of the AND is ROMCS#. When all inputs are high, ROMCS# is high. When any pin is driven low, ROMCS# goes low.

To exit from test mode, pulse SYSRESET while TEST# (4041 LIN pin) remains high. Keep SYSRESET high for at least 5 cycles of SCLK.

6. 84045 Functional Description

6.1. SIPC Chip Overview

The 4045 is derived from the 4035, with a few features added to support the 4041. The 4045 features described in this document apply to the 4035 as well, except where otherwise noted. The 4045 is designed to replace the 4035 in CS4031 designs, as well as providing additional features useful in CS4041 designs. The term "SIPC" refers to both the 4045 and the 4035.

The SIPC is an 82C206 Integrated Peripheral Controller ("IPC") with additional logic added to reduce signal and/or chip count of the external logic. "SIPC" can be interpreted to mean "Superset IPC." The following features are added to the 206 to form the SIPC:

14MHz crystal oscillator circuit and divider
DMA clock divider from SCLK
32KHz crystal oscillator circuit
System reset logic
System arbitration logic including support for local bus masters
ISA bus hidden refresh logic
Performance control logic to emulate an 8MHz PC/AT
DMA controller address generation logic
SA17:19 buffering
A20M# generation
486 floating point error logic
Speaker interface
Control link to communicate with the 4041 or 4031

6.1.1. 4045 Added Features

The 4045 includes the following features not available in the 4035:

WBACK# signal. Accepts the WBACK# output signal from the 4041. This pin was reserved on the 4035.
Additional LREQ/LGNT pairs. 2nd and 3rd sets are available by changing the functions of other pins.
Arbitration Lock added. Prevents HOLD from going active during clock speed changes.
Misbehaved ISA master fix. HOLD will not go inactive until MASTER# is high.
A10:15 decode input pin. Allows 16 bit decode of internal IO. Shared with SA19 pin.
RTC IRQ output. Available on the IGNNE# pin when PWRGOOD is inactive.
RTC Password Protect. PS/2 style via Port 92h (same as in CS4021).
Ability to **disable the port 92 reset function** to allow redirection to SMI in the 4041.
DMA clock divider allows 14.3MHz clock to be used as a source.
Port 26/27. SMM configuration register access window.
Mouse Interrupt. The mouse interrupt is optionally sent to the 4045 across the control link.
SRESET arbitration mode. This mode controls the CPU reset and HOLD arbitration, allowing the CPU to be reset only when it is in HOLD.
IPC Reset Disable. A configuration bit allows the reset to the IPC core to be blocked. This is used to retain the IPC registers during a 0 volt suspend (suspend to disk).
ISA Refresh Disable. Allows ISA refresh to be disabled while DRAM refresh remains enabled, resulting in a small performance benefit and allowing a TTL buffer to be eliminated if ISA refresh isn't required.

6.1.2. Using 4045 in place of 4035

The 4045 is a pin-for-pin drop-in replacement for the 4035. The 4045 powers up as a 4035, allowing it to function exactly as a 4035. There are a few board and system level considerations to make sure that none of the 4045 extended features are accidentally enabled in a 4035 environment.

The DGNT# pin should have a pull-up resistor on it, since one of the superset options (SA17:19 redefinition) may be essential for ROM access at power up and is based on the state of the DGNT# pin when SYSRESET goes low. A 10K pull-up is sufficient.

Index register 09h bit 3 should be set to a 1 to assure that ISA refresh is enabled. For the 4035, this bit is described as "Reserved, write as 0."

Index registers 0Bh and 0Ch should be left in their default states. They power up to 0s, in which all functions are 4035 compatible. These registers do not exist in the 4035.

6.1.3. A quick design checklist.

The following system level functions are required for proper 4045 operation.

VCC must be supplied from a battery backed up circuit in order for the Real Time clock and CMOS RAM to operate when the power is off. There are no separate power pins for these. The circuit that switches between battery and main +5V must be capable of delivering normal operating currents to the 4045 during normal operation without excessive voltage drop. Refer to the Application Schematic Examples for a suggested circuit.

DGNT# must have either a pull-up or pull-down resistor in order to set the mode of the SA17:19 pins at reset. A pull-up selects SA17:19 while a pull-down selects the additional VL bus arbitration signals and the 16-bit I/O chip select input. DGNT# is floated while SYSRESET is high and sampled on the falling edge of SYSRESET.

A20M# should have a pull-up resistor. 10K is sufficient. After reset A20M# is the TEST input pin. A configuration register bit enables the A20M# output driver. The pull-up is required both to prevent the SIPC from going into a test mode and to allow the CPU to boot properly (A20M# held high).

LOUT should have a pull-up. LOUT is floated at reset because it connects to 4041 LIN, and 4041 LIN is the TEST pin on the 4031 and 4041. The same configuration register bit that enables A20M# also enables LOUT.

6.2. Clocks

Three clock frequencies are used in the SIPC:

14.31818MHz for the timer functions
SCLK for the CPU related functions and to generate DMACLK
32KHz for the real time clock

6.2.1. 14.31818 MHz clock

Crystal pins are provided for a 14.31818 MHz clock. 20pF capacitors should be included from each crystal pin to ground as well as a 10M resistor across the crystal pins. X2 should be buffered with a bus driver or inverter to form OSC for the ISA bus. A 33 ohm series resistor should be provided between the buffer and the bus, close to the buffer.

The 14.31818 MHz clock is divided by 12 internally to form the 1.19MHz clock used by the IPC timers.

To conserve battery power during standby mode, the 14 MHz clock does not run (14 MHz output is driven low continuously) when PWRGOOD is low. There is a delay of about 250 ms from the rising edge of PWRGOOD to the falling edge of SYSRESET to allow ample time for the 14 MHz oscillator to reach stable operation after PWRGOOD goes high (as well as to allow a CPU internal VCO to stabilize.)

6.2.2. SCLK

SCLK, the 1x CPU clock, is provided to the SIPC. SCLK is used for the arbitration logic and (unless a 14.3 MHz mode is selected) for generating the clock to the DMA controllers and ISA refresh logic.

6.2.3. DMA and Refresh clock generation.

The DMA controller and refresh clock (Index 0Ah) should be around 8 MHz. It is normally divided by 2 by the IPC core (see Index 01h) to obtain the 4MHz at which the DMA controllers normally operate. Refer to Index register 0Ah description for a list of available dividers and sources. The 14.3 MHz modes are available only in the 4045 and are recommended for use when SCLK frequency is changed for power reduction.

6.2.4. 32.768KHz clock

Crystal pins are provided for the 32.768KHz clock. This clock is operating at all times, running off of the battery when the power is off. It is used only by the real time clock portion of the IPC. The crystal frequency should be 32,768 Hz (32x1024).

6.3. Reset

The SIPC contains the reset logic for the system. It receives PWRGOOD and generates SYSRESET and CPURESET in response to PWRGOOD going high. It also receives soft reset requests over the control link or Port 92h to generate CPURESET.

PWRGOOD disables all outputs and gates off all inputs to the chip except for PSRSTB, the 32KHz oscillator pins, and of course PWRGOOD itself. Also, the 14.3MHz output is driven low rather than floated. When PWRGOOD goes high the outputs are enabled. SYSRESET and CPURESET are driven high, and remain high for approximately 8 million SCLKs (250 ms if SCLK is 33.3 MHz) to assure proper startup of the 14.31818 MHz oscillator, and to allow the 486 internal VCO to stabilize.

The 1x clock is expected to be in the same phase as the 486, which is high for phase 1 and low for phase 2. The resets will be generated on the rising edge of SCLK.

SYSRESET is generated based on the PWRGOOD circuit alone. CPURESET is generated based on PWRGOOD, but is also be generated for "soft resets". The following are the sources of soft resets:

Keyboard controller reset
CPU shutdown cycle
Port 92h bit 0 transitioning from a 0 to a 1.

Keyboard Reset and Shutdown are sent to the SIPC through the control link from the 4041. When the SIPC receives the request, it immediately passes the request on to the logic which arbitrates CPURESET with CPU HOLD.

Port 92h is contained in the SIPC. When bit 0 makes a 0 to 1 transition, a CPU reset is requested after a delay of about 16 cycles of the 4045 internal BUSCLK (Index 0Ah). This delay allows the CPU to execute a HALT instruction following the output to port 92h.

In the 4045 the port 92 reset may be disabled, allowing the 4041 to redirect it to SMI. The 4041 will also redirect the other sources of the CPU restart by not sending the link code.

The SIPC interlocks CPURESET and the CPU HOLD signal to prevent both from occurring simultaneously. This is required for a 386 CPU. If the CPU is currently in HOLD when the reset request occurs, the reset is delayed until after the HOLD is removed. Likewise if a hold request is issued during the CPU reset sequence, it is delayed until after the CPURESET is removed. The 4045 also contains an option to reset the CPU only when it IS IN HOLD. This is for the SL enhanced CPU, as described in the arbitration logic section.

CPURESET is 16 SCLKs long (minimum) for soft resets. The falling edge always occurs from the rising edge of SCLK.

DGNT# floats at power up (PWRGOOD low) and remains floated until SYSRESET goes low. On the falling edge of SYSRESET it is sampled internally to determine the function of SA17:19. A 10K pull-up resistor should be used to select SA17:19, or a pull-down to select the LREQ2# / LGNT2# and IOCS# functions.

PSRSTB is internally blocked when PWRGOOD is high. This allows the use of a very long time-constant RC circuit to generate PSRSTB, without any risk of a system malfunction if PSRSTB is held low too long after the BIOS begins executing. PSRSTB can be grounded at any time during normal system operation (PWRGOOD high) without any effect. If PSRSTB goes low at any time while PWRGOOD is low, the RTC "VRT" bit will be cleared to alert the BIOS that the system battery may be dead.

6.3.1. Inhibiting IPC Reset for 0V Suspend

The 4045 has an option to block reset to the IPC core following PWRGOOD rise, to support 0V suspend applications (suspend to disk). If config register 0C bit 7 is set to a 1, the IPC core will not be reset. This config bit is reset only when PWRGOOD and PSRSTB are both low, and will retain its value when the power is shut off, as long as the battery voltage is maintained for the 4045. The purpose of this option is to allow a suspend to disk function without having to reinitialize the IPC peripherals. Software should set this bit before shutting off the power when doing a suspend to disk. It should reset the bit after resuming, so that the IPC core will be reset following a normal power off.

The IPC core includes the DMA Controllers, Interrupt Controllers, Timers, RTC, and Configuration Register 01h. The reset inhibit does not prevent the clearing of other 4045 Configuration Registers (Indexes 08h through 0Ch) or loss of the port 22h index value when PWRGOOD goes low. Systems that implement the 0V suspend feature must save and restore the non-preserved Configuration Registers. If the software that decides to suspend is invoked by an SMI or other interrupt, care must be taken not to split a port 22h/23h access, which shouldn't be happening anyway if the system is idle and ready to suspend. Normally only the BIOS performs 22h/23h accesses and probably will inhibit interrupts when accessing ports 22h and 23h during normal system operation.

The 4045 reset inhibit feature is intended for applications where the 4041 and CPU are both powered down during suspend and must be reset when PWRGOOD returns to the high state. PWRGOOD must go low to put the 4045 into its lowest current standby mode. While PWRGOOD is low, the 4045 floats SYSRESET and CPURESET, which are pulled high by external pull-up resistors. When PWRGOOD goes high, the 4045 always keeps SYSRESET and CPURESET high for a fixed time interval (around 250 ms if SCLK is 33.3 MHz), regardless of whether or not the 4045 internal IPC reset has been inhibited.

If the CPU and 4041 are not powered down during suspend, then other power management features such as CPU clock control and SMI can be used to optimize system power consumption.

6.4. GATEA20

The SIPC generates A20M#. In response, the 486 CPU performs the actual gating of address bit A20. This gating is needed for software compatibility with the 8088 and 8086 at address FFFF:10h and above.

Within the SIPC itself, there are two sources for A20M#:

Control link codes from the 4031 or 4041
Port 92 bit 1.

The 4031 and 4041 OR together all of their sources for GATEA20, and send a code across the link whenever this signal changes. The main sources in the 4041 are the keyboard controller GATEA20 (external, internal, or emulated) and the SMM GATEA20 function.

The A20M# output pin is shared with the TEST# input. At power up this pin is floated and remains that way until index register 09 bit 4 is written to a 1. This bit disables the test input and begins driving A20M#. Until the bit is set to a 1, an external pull-up resistor keeps the pin high, preventing the SIPC from going into test mode, and holding A20M# high to the CPU to allow it to boot.

6.5. Arbitration

6.5.1. Arbitration Overview

The arbitration logic for the system is contained in the SIPC. It performs the following functions:

Arbitrates between the CPU, local bus masters, and the internal DMA controllers.
Arbitrates between 2 or 3 sets of VL masters (optional, in 4045 only)
Puts the CPU in HOLD to slow it down when performance control (DeTurbo) is enabled.
Arbitrates between CPU HOLD and CPURESET (in two different ways).
Arbitrates between the DMA controller and hidden refresh.
Preempts VL masters when a DMA cycle is pending.
Brings the CPU out of hold to perform a cache write back following a snoop.
Prevents the CPU from going into HOLD to allow for a speed switch (through a config bit).

The following signal pins are involved in arbitration:

HOLD and HLDA for the CPU
LREQ0# and LGNT0# for local bus masters
LREQ1# / LGNT1# and LREQ2# / LGNT2# if enabled.
DGNT# indicating the DMA controller has the bus.
WBACK# for CPU write back cache support.

6.5.2. VL Master Arbitration

In the 4045, VL master arbitration for three sets of LREQ/LGNT signals is similar to the arbitration performed with an external PAL in a 4035 system, but more elegant. (PAL implementations generally use a simple scheme in which VL masters keep pre-empting each other rapidly). When a VL master owns the bus and another VL master requests it, the arbitration logic will take the first LGNT# high, wait for the LREQ# to go high, then give the bus to the second master. In order to prevent VL masters from continuously preempting each other and dropping system performance, a master will not be preempted by another VL master until it has had the bus for 31 T states. This is enough to do several average memory accesses, but not long enough to cause latency problems. The 31 clocks start when the VL master gets control of the bus. The DMA controller will preempt immediately, without waiting for the clocks.

Rotating priority is used for selecting the VL master when more than one is requesting. The order is LREQ0#, LREQ1#, LREQ2#. The DMA controller has a higher priority than any VL master.

LREQ0# and LGNT0# have dedicated pins.

LREQ1# and LGNT1# share pins with the SLOW# and FLUSH# pins respectively. Register 0Bh bit 3 selects the function of these pins. The power-up default is SLOW# and FLUSH#, with Index 08h = '00h,' as well. The result is that the state of the SLOW# input has no effect, and the FLUSH# output is driven high. Performance control (Index 08h) is the only source for FLUSH# in the SIPC. If the FLUSH# pin is actually being used as LGNT1#, there is no conflict since the SIPC will be driving it high.

LREQ2# and LGNT2# share pins with SA17 and SA18 respectively. The functions of these pins must be set at reset time or the system may not boot up correctly, since proper SA17 function may be needed by the BIOS ROM. On the other hand, systems that use LREQ2# and LGNT2# would have several problems if these signals were driven with address bits. These pins must operate in their proper functions at power-up. To accomplish this, the DGNT# pin is sampled and internally latched at the end of SYSRESET to select the function. This pin is normally an output but is floated from when PWRGOOD is low until SYSRESET goes low. Putting a pull-down on DGNT# selects LREQ2# and LGNT2# (along with the IOCS# input on SA19) while putting a pull-up on DGNT# selects SA17:19.

Arbitration for more than one set of LREQ# / LGNT# signals adds 1 clock of overhead for synchronization for master switching. If only LREQ0# and LGNT0# are needed and sets 1 and 2 are disabled, this extra clock is eliminated, resulting in a slight arbitration performance advantage.

6.5.3. Main Arbitration Logic

The primary function of the arbitration logic is to arbitrate the system bus between the CPU, VL masters, and the DMA controller (which is also used by ISA masters to get the bus). There are several other functions it must perform, generally to prevent certain things from happening at the same time.

By default the CPU has the bus. The CPU is put in HOLD when any other master needs the bus. There is a two way arbitration between Local Bus Masters and the DMA controller The CPU might not get control of the bus between a local master access and a DMA cycle. The arbitration is performed using SCLK, and all signals are synchronous to it. The DMA controller HOLD is synchronized to SCLK before being used by the arbitration logic.

There is a fixed priority to bus activity. The priority is as follows:

1. DMA controller
2. Hidden Refresh
3. Local Master
4. CPU

Note that either a local master or the CPU have control of the bus when a hidden refresh occurs. Hidden refresh is listed above because it may not occur when the DMA controller has the bus, and it has a lower priority. For instance, if the DMA controller requests the bus and a hidden refresh request occurs before the CPU or local master gives up the bus, the hidden refresh will be delayed until after the DMA controller gives up the bus.

A DMA request will preempt a VL master using the standard VL bus protocol.

Other Arbitration functions, which are each described below, are:

Hidden Refresh arbitration with DMA.
CPU reset arbitration with HOLD
Performance control HOLD (DeTurbo Mode)
Arbitration Lock when the CPU VCO is not stable.
WBACK# pin removing the CPU HOLD.
Misbehaved ISA master fix (waiting for MASTER# to go high before giving control to the CPU or VL master)

6.5.3.1. Hidden Refresh arbitration

Hidden Refresh is arbitrated with the DMA controller, since they must be mutually exclusive. The hidden refresh may occur while the CPU or a local master is in control. Unlike DMA and local masters, hidden refresh does not require the CPU to go into HLDA state. Hidden refresh and an ISA bus cycle are further arbitrated in the 4041, since an ISA refresh cannot occur at the same time as another ISA cycle, and the 4041 is the ISA bus controller for CPU or local master cycles.

6.5.3.2. CPU reset arbitration

The CPU reset is arbitrated with the CPU HOLD signal. The CPU will not be reset while it is in hold, although this is generally not a problem for 486 CPUs.

The 4045 (unlike the 4035) has an option to reverse this function. When this mode is enabled, the CPU will not be reset until it IS IN HOLD. This mode is specifically for the SL enhanced CPUs, which take M / IO#, D / C#, and W / R# high as soon as they receive a RESET or SRESET. This will turn any cycle into a memory write, which could cause severe problems if the CPU happens to be doing a memory read or code fetch at the time of the reset (which some programs are known to do). When CPU reset is used to return the CPU from Protected Mode to Real Mode, as was necessary with software originally developed for 80286-based systems, it is essential not to corrupt the contents of DRAM or caches.

When a CPU reset is requested, a HOLD request will be generated. The reset will occur when both HOLD and HLDA are active. The rest of the arbitration logic will function as normal, which means that the DMA controller or VL master may be granted the bus, or already have the bus when the CPU is being reset. CPU cache snoop cycles can safely be ignored at this time since the CPU cache will be flushed and disabled upon coming out of RESET. CPUs with an L1 write back cache will handle the SRESET as an interrupt instead of resetting the CPU, and will continue to accept snoop requests.

This mode is controlled by register 0C bit 3. A 0 selects the original mode. A 1 selects the SL mode. System resets (initiated by PWRGOOD going high) will be sent to the CPU as normal, without arbitration.

6.5.3.3. VL bus preemption by DMA

The LREQ# and LGNT# signals support a preemptive protocol for the VL-Bus masters. If a DMA HOLD occurs while a VL-BUS master has the bus, LGNT# is driven back high, requesting that the local master give up the bus. The local master will take LREQ# back high to indicate that it has given up the bus.

6.5.3.4. Performance Control HOLD

The performance control HOLD request (Index 08h) is ORed with the other sources of HOLD, but does not enter into the arbitration. A DMA cycle or ISA master may gain access to the bus during a performance control HOLD without waiting for it to finish.

6.5.3.5. WBACK# Signal

The WBACK# will remove the CPU hold, without going through any arbitration. The LGNT# or DGNT# (which ever is active) will remain active at this time, even though HOLD and HLDA go inactive. A CPU with a write back cache will pull WBACK# low in the middle of a DMA, ISA Master, or local master bus cycle to allow the CPU to perform a write back cycle. Whatever master has control of the bus when the WBACK# signal is activated will retain control until after WBACK# goes inactive.

There are two timing modes for this. The first is the 4035 mode, which keeps HOLD low as long as WBACK# is low. This mode is not particularly useful, but is included for historical reasons. The WBACK# pin was actually listed as reserved on the 4035. The second mode keeps HOLD low for only 4 T-states, then takes it back high. This allows the CPU to come out of HOLD only long enough to do the write back cycle (it will not go back into HOLD until it is finished). The timing is shown below:

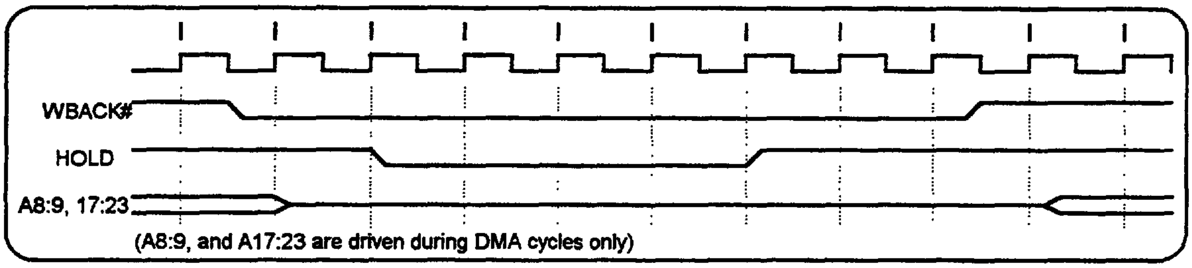


Figure 6.1 WBACK# Timing

If the DMA controller has the bus, the SIPC floats A8:9 and A17:23 when WBACK# is active (WBACK# is clocked internally first). WBACK# should be connected to the 4041 WBACK# output pin.

6.5.3.6. Arbitration Lock

This applies to the 4045 only. The arbitration lock is for the S series CPU. When the frequency is changed the CPU will not respond to EADS# cycles properly until the internal VCO has stabilized. Software will set the arbitration lock, switch the frequency, then release the lock after the VCO stabilizes.

When register 0B bit 0 is set to a 1, HOLD will not go high. If it is already high it will not be affected.

Software Caution: When register 0B bit 0 is a 1 and register 0C bit 3 is set to a 1, if a CPU restart is attempted by actually resetting the CPU, the CPU reset will NOT occur until one of these bits is set to a 0. The CPU reset will be waiting for the CPU to go into HOLD, but HOLD will be blocked. This situation should never occur in normal operation.

6.5.3.7. Misbehaved ISA master fix

Some ISA masters do not take MASTER# high until DACKn# goes high. The round trip time for this, and the fact that MASTER# is open collector (and rises slowly) can cause the A to SA/LA buffers to be still turned around when the CPU begins its next cycle. The 4045 (unlike the 4035) prevents the CPU or VL master from getting control until a minimum of 90 ns (3 T-states) after MASTER# goes high.

6.6. Performance Control (DeTurbo)

"DeTurbo" or Performance control refers to slowing down the system to approximately that of an 8MHz IBM AT to allow some old software, which cannot operate on a fast machine, to be used. This is becoming less and less of an issue as this software becomes increasingly out of date and infrequently used.

In the past, this was often accomplished by switching the clock to about 8MHz. With the advent of SL Enhanced CPUs and Green PCs we are again switching the clock, but this time for power savings. The performance control logic in the 4035 and 4045 puts the CPU in hold for a programmable percentage of time instead of switching the clock, leaving clock control entirely to the power management software.

Performance reduction via HOLD is implemented based on refresh. On every refresh request, the CPU is put in HOLD for a programmable period of time. Optionally, the FLUSH# pin is also pulled low during this time to make sure the CPU doesn't continue to execute out of its internal cache. The length of the HOLD pulse is selected to provide the desired degradation in performance. The intent is to match the speed of a 6 or 8 MHz AT in order to allow some games and copy protection programs to work properly. The FLUSH# and HOLD functions may be enabled separately, but generally they will be used together.

After the HOLD duration is set (Index 08h), the slow mode may be enabled and disabled by either of two ways: Index register access or the SLOW# input signal. This is an OR function, allowing either of them to slow the system down (they both should be disabled for full speed).

The clock generated for the IPC DMA controllers is used for the time base of the performance control. This clock is normally about 8 MHz (0.125 µs period). The corresponding maximum HOLD delay programmable via Index 08h is 15.9 µs (127 cycles of 0.125µs). However, to avoid possible system lockup, care must be taken not to program a delay greater than the time between refresh cycles, normally about 15 µs (18 cycles of 1.193 MHz). With a 486 CPU running at 25 MHz or more, Index 08h probably will need to set in the range of F0h to F8h (close to the refresh period) to reduce effective system performance to the 6 to 8 MHz range.

The HOLD is initiated each time the refresh request occurs from timer 1. Local bus master, DMA, ISA master, or refresh cycles may occur while the performance control HOLD is taking place, avoiding a DMA underrun or overrun.

Note that the FLUSH# pin on the CPU is also optionally driven by the 4041. If both the performance control and power management FLUSH# functions are required, a low-true OR function (physical AND gate) must be used externally. The 4041 is designed such that the FLUSH# pin is not required for SMM mode in most cases, so only the 4045 FLUSH# signal need be used.

6.7. Refresh

The SIPC performs the ISA bus refresh. It drives SA0:7, REFRESH#, and MEMR# directly.

When timer 1 produces a refresh request, it sets a "RefreshPending" flip-flop. The output of this flip-flop is then arbitrated with the internal HLDA for the DMA controllers. Internal HLDA for DMA will not go active when the hidden refresh sequence is taking place, and the hidden refresh sequence will not begin while internal HLDA for DMA is high.

When hidden refresh wins the arbitration, the refresh request is communicated to the 4041 via the Control Link. The 4041 arbitrates this with CPU or local master accesses to the ISA bus, and sends a Refresh ACK back to the SIPC. At this point the SIPC performs the ISA refresh (if enabled). When the ISA refresh cycle is complete, the SIPC signals this to the 4041 via the Control Link. Within the SIPC, the internal HLDA to the DMA controller may then go active if a DMA request is pending.

The following timing diagram shows a hidden refresh cycle. Hidden refresh wins the arbitration with DMA, since there is no pending DMA request. RefAck is the output of the Link Input logic of the SIPC, and indicates that 4041 has sent a Refresh Acknowledge over the link, allowing the refresh to proceed. During the hidden refresh sequence, DGNTIN# goes active (the DMA HLDA input to the DMA/Refresh arbiter). DGNT# is withheld until the hidden refresh is finished.

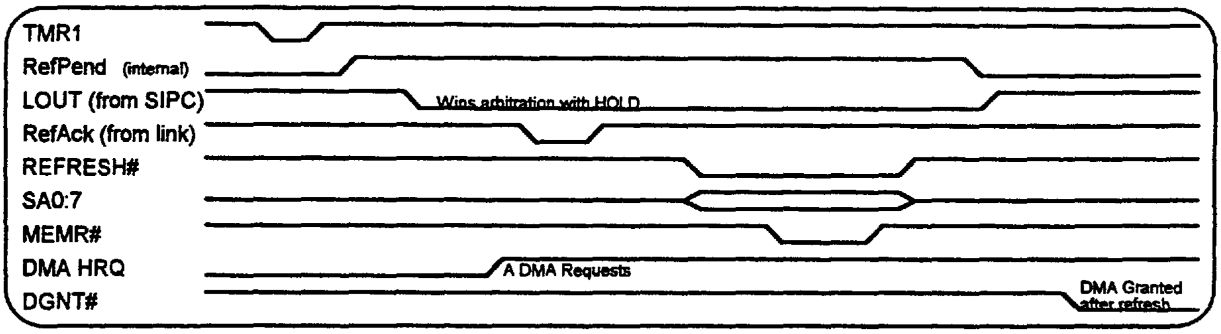


Figure 6.2 Hidden Refresh Timing

For master refresh, no arbitration need be done since the current master is requesting the refresh. When the ISA master pulls REFRESH# low, the SIPC performs the refresh by driving SA0:7 with the refresh and driving MEMR# and SMEMR#. The SIPC also pulls LOUT low for the duration of REFRESH# active to indicate to the Sprite chip that a refresh is occurring. In response to SIPC LOUT, the 4041 performs a refresh for local DRAMs.

The SIPC drives both MEMR# and SMEMR# low for refresh cycles, regardless of the address on the upper bits (for normal bus cycles SMEMR# is only driven when A20:23 are all low).

Two configuration bits control refresh:

> Register 09 bit 2 controls the Refresh Request signal from Timer 1 of the 8254 equivalent logic in the IPC core. This bit blocks the Refresh Request signal when a 0, which is the power up default. The main purpose of this is to avoid refreshes immediately after power up. The 8254 timer does not receive a reset of any kind, and continues to operate after a hardware reset. Software should set this bit to a 1 for normal operation. All refreshes will be disabled when this bit is a 0, including both DRAM and ISA refresh cycles.
>
> Register 09 bit 3 controls ISA refresh. This function exists in the 4045 only (not the 4035). ISA refresh is always enabled in the 4035. When this bit is a 0, ISA refreshes do not occur. All of the arbitration described above still occurs, since part of this arbitration is what signals the DRAM controller to do a refresh, and the 4041 does not know whether or not ISA refresh is enabled. When the refresh is granted by the 4031/4041, the 4045 immediately signals that the ISA refresh is complete.

The "normal operation" setting for both of these bits should be a 1.

6.8. ISA Bus

This section describes the generation and usage of the ISA bus control signals on the SIPC.

AEN

This signal is driven continuously high during DMA cycles, low at all other times (including ISA Master cycles). During DMA, AEN indicates that the address on the SA-bus is a memory address, not an I/O address, even though IOR# or IOW# is active.

MASTER#

This is an input. It forces AEN continuously low during ISA Master cycles.

DREQ0:7, DACK0:7#, TC

These come directly from the DMA logic in the IPC megacell. DREQ0:7 are inputs. DACK0:7# and TC are always outputs.

MEMR#, MEMW#

These signals are inputs except during DMA and refresh. They are used to generate SMEMR# and SMEMW#. During refresh (either SIPC-initiated or ISA Master-initiated), MEMR# is an output from the refresh logic. MEMW# remains an input during refresh. During DMA cycles, both signals are outputs from the SIPC.

SMEMR#, SMEMW#

Always outputs, even when an ISA master has the bus. They follow MEMR# and MEMW# respectively when the address is below 1Meg (A20-23 are all low). They are high if any of A20-23 are high. An exception to this is when REFRESH# is low, at which time SMEMR# follows MEMR# regardless of the address.

IOR#, IOW#

These signals are outputs during DMA, inputs at all other times.

6.9. IPC Functions

The 4035 and 4045 include the IPC megacell, which is basically the 82C206 chip. It includes many of the "Compatible" peripherals necessary to make an AT computer. The functions are as follows:

2 DMA controllers, 8237-compatible with page register extension.
2 Interrupt Controllers, 8259-compatible
1 Three-channel Timer, 8254-compatible
1 Real Time clock and with CMOS RAM, a superset of the Motorola 146818

6.9.1. DMA Controllers

The DMA controllers consist of two 8237 compatible blocks and a page register module, compatible with the 74LS612.

8237 DMA controllers are designed for 8-bit data transfer using a 16-bit address. PC/AT architecture allows 16-bit data transfers by left-shifting one controller's address by one bit. PC/AT architecture also allows a 24-bit address by using a 74LS612 dual-port RAM to provide the additional address bits.

One of the 8237s is used for 8 bit DMA, with its address lines mapped directly onto the system address bus. The other DMA controller is adapted to make it perform 16 bit DMA. This is done by shifting the address lines by one position, and forcing A0 to a 0. Software must shift the address by 1 bit before loading the DMA controller's registers. Each time the address in the DMA controller increments, it effectively increments the resulting system address by 2. The lower 4 address lines on an 8237 are used to select the register when programming the chip. Because they are shifted by one position in the 16 bit controller, the registers only show up on every other IO port. Because of this arrangement, there are several restrictions on DMA in all PC/AT compatible system:

Channels 0, 1, 2, and 3 are always used for 8 bit DMA.
Channels 5, 6, and 7 are always used for 16 bit DMA.
8 bit DMA must be within a 64KB memory range (only the 16 bit address increments) until software updates the corresponding page register.
16 bit DMA must be within a 128KB memory range (the address is shifted by 1 bit) until software updates the corresponding page register.

Each DMA controller has a HOLD output and an HLDA input, plus 4 DREQ inputs and 4 DACK# outputs (the DREQs and DACKs have programmable polarity, but the AT standard is active high for DREQ (for unknown reasons, although it's useful for testing), and active low for DACK. The 16 bit channel's HOLD and HLDA signals become the external HOLD and HLDA signals after processing by the arbitration logic. The 8 bit controller's HOLD and HLDA are connected to the 16 bit controller's DREQ0 and DACK0#. This channel, which would have been channel 4 otherwise, is put in "cascade mode." This means that the 8 bit controller goes through the 16 bit controller to get the bus, adding some overhead to its arbitration.

Cascade mode, which basically converts DREQ/DACK pair into a HOLD/HLDA protocol, is also used by ISA bus masters to get the bus. A DMA channel in this mode will request the bus in response to a DREQ, and take DACK# low when it gets the bus, but will not drive the address or control signals for the DMA cycle, expecting something else to do it (the ISA master or a external DMA controller). The DACK# stays active until the DREQ goes low.

The upper DMA addresses (A16:23 for the 8 bit controller and A17:23 for the 16 bit controller) are provided by a small dual port RAM, also known as the DMA Page Register array, which was implemented with a 74LS612 in the IBM PC/AT. One port is IO mapped and used to write and read the RAM. The other port has a separate set of internal address lines and outputs. The address lines are controlled by a complex combination of the DACK# signals which saved logic on the original PC/AT, but made the IO port ordering confusing. Only 8 of the 16 locations are used for DMA (one of which is output for REFRESH cycles). D7:0 map to A23:16 respectively, but bit D0 (A16) not used for 16 bit channels. The page register bits are not shifted.

DMA cycles are done by transferring the data directly from the IO device to the memory device. IOR# and MEMW# are activated together for memory writes, and MEMR# and IOW# are activated for memory reads. The data is not held in any intervening register.

For memory writes, IOR# goes low one DMA clock before the MEMW# since "extended write" mode is always used. This is necessary since the data must be stable from the IO device before CAS falls on the DRAMs, which is timed from the beginning of MEMW#. For memory reads, MEMR# is delayed by internal logic outside the 8237 to match the IOW# command. IO devices should latch the data on the rising edge of IOW# since it won't be there anywhere near the beginning. The MEMR# delay is presumably there to allow enough address setup time to the DRAM controller. RAS# is generated with MEMR# going low. The MEMR# delay is programmable in the IPC through Index 01h.

SIPC LOUT acts as a strobe to latch DMA address bits A10:16 into the 4041 from the XD bus. These address bits are not connected to the SIPC (to optimize the pin count). The SIPC drives SA0:7 directly during DMA, along with A8:9 (which are also used by the SIPC as inputs during I/O reads and writes to SIPC registers). Address generation during DMA is summarized below.

Driven directly by the SIPC:
SA0:7
A8:9
A17:23
SA17:19 (optional)

Driven by the 4041 or externally pulled low:
A10:16 Driven by the 4041 using the value previously latched from the XD bus by SIPC LOUT.
A24:27 Driven low (000) by the 4041.
A28:30 Pulled low (000) by external pull-down resistors. Not connected to either the SIPC or 4041.
A31 Driven low by the 4041.

Driven by external A:SA address buffers:
XA0:1 & A2:7 Driven by A:SA address buffers from SA0:7 (SIPC).
SA8:9 Driven by A:SA address buffers from A8:9 (SIPC).
SA10:16 Driven by A:SA address buffers from A10:16 (4041).
SA17:19 Driven by A:SA address buffers from A17:19 if not driven directly by SIPC.
LA17:23 Driven by A:SA address buffers from A17:23 (SIPC).

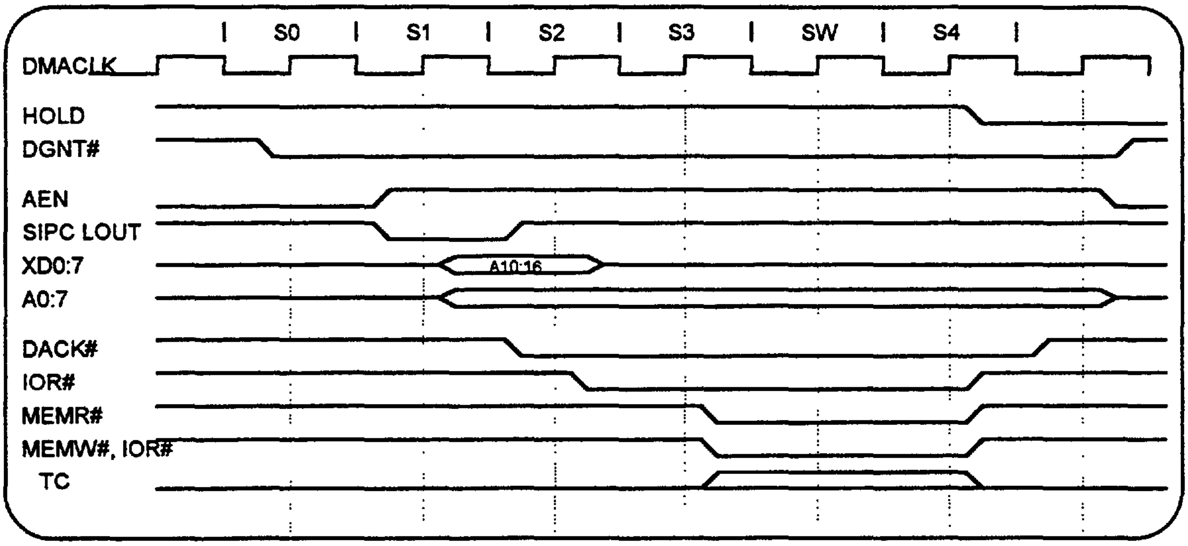


Figure 6.3 DMA Cycle Timing

6.9.2. Interrupt Controllers

The IPC contains two 8259-compatible interrupt controllers. The controller configured as the master is at IO ports 20h and 21h, and accepts interrupts IRQ0:7. The controller configured as the slave is at IO ports A0h and A1h and accepts interrupts IRQ8:15. What would be IRQ2 from the master controller actually receives the cascaded interrupt request from the slave controller. The diagram below shows the basic hookup.

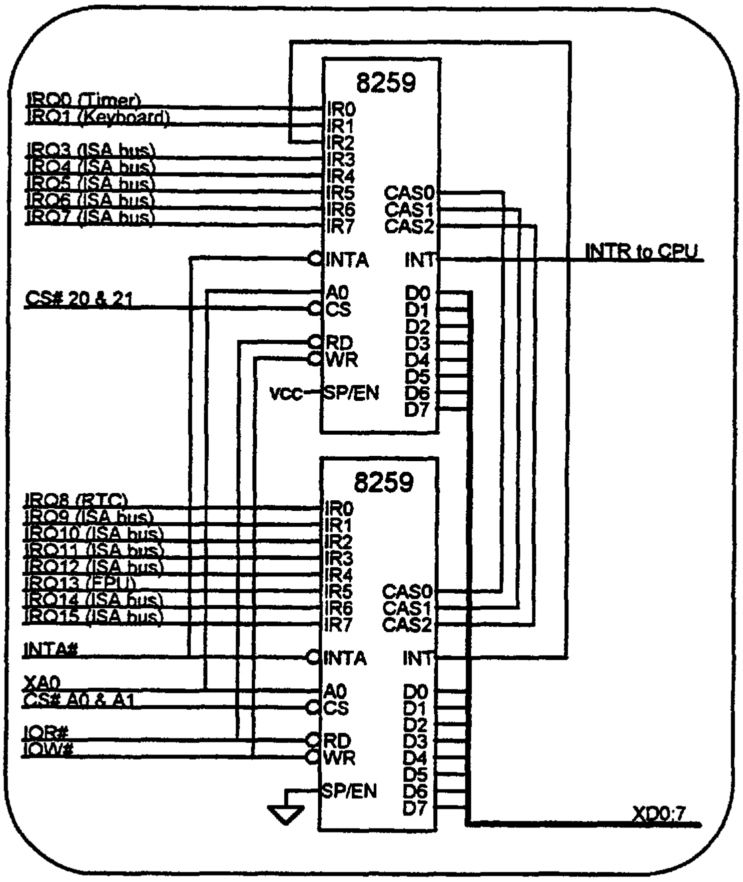


Figure 6.4 Interrupt Controller Internal Connections

In a PC/AT the interrupt controllers are programmed in edge triggered mode. This means an enabled IRQ flows through to INTR transparently until INTA occurs, but is latched by INTA. For proper recognition, an IRQ must remain asserted until after the corresponding INTA has occurred. The interrupt base for the master controller is 08 (yielding vectors 08-0F during INTA cycles) while the slave controller is set to 70h (yielding vectors 70h-77h). Windows reprograms them to 50h and 58h respectively. Other operating systems may put them at different places.

6.9.3. Timers

The IPC contains an 8254 compatible 3 channel timer. All three timer channels run off of a 1.19 MHz clock (14.31818 MHz divided by 12). The timers are used as follows:

Table 6.1 Timer usage and setup

Timer	Usage	Mode	Divider
0	Periodic Interrupt	Square Wave (mode 3)	0000 (65536)
1	Refresh	Pulse (mode 2)	0012h (18)
2	Speaker	Various (Generally Square Wave)	Various

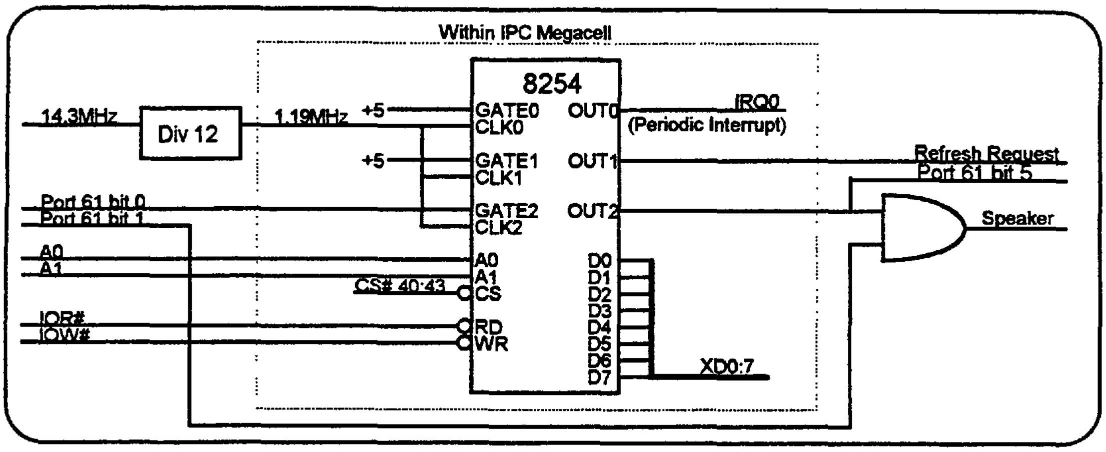


Figure 6.5 8254 Timer Internal Connections

The Timer 0 value results in IRQ1 occurring approximately 18 times per second, which is used by DOS for a time-of-day clock (separate from the 32.768 KHz based RTC, but initialized by DOS based on the hardware RTC).

The Speaker signal is inverted before being driven out on the SPKR pin (see Port 61h description).

6.9.4. RTC

The IPC contains the equivalent of a Motorola 146818 RTC/CMOS RAM chip. An extra 64 bytes of CMOS RAM have been added and are normally used to store chipset specific information. The RTC is accessed through an address register at port 70h, and a data port at address 71h. Bit 7 of port 70h is the NMI mask, and should be written to a 0, which enables the NMI.

The interrupt output of the RTC goes to IR0 of the slave controller, which is IRQ8. This may be programmed to provide a periodic interrupt, or an alarm interrupt. It is normally disabled.

The RTC runs off of a 32.768KHz crystal. It is powered by a small battery when the system power is off (the entire 4045 is powered up but is in very low ICC standby mode when PWRGOOD is low).

External RTC. The internal RTC can be disabled, if desired, to allow an external RTC to be used instead. Index 0Ah bit 7, when set to '1', disables the 4045 internal RTC. I/O address decoding for ports 70h/71h (except 70h bit 7) requires a few gates of external logic when the internal RTC is disabled. GPC or GPD from the 4041 may be useful as I/O decodes for an external RTC. Port 70h bit 7 (NMI disable) is implemented in the 4041 and remains under 4041 control as usual. The function of the 32KX1 pin changes to become the IRQ8# input for the external RTC interrupt. IRQ8# is active-low (unlike other IRQ inputs), allowing the INT# output of most typical external RTCs to connect directly to the 4045 IRQ8# pin without an external inverter. The 32 KHz oscillator for the external RTC becomes the responsibility of external logic. All CMOS RAM information needed by the system (normally 64 bytes minimum) also becomes the responsibility of external logic when the internal RTC is disabled.

Special care may be needed to maintain the lowest possible standby current in an external RTC when main system power is turned off. The most popular external RTC devices may have provision, such as an on-chip VCC detector, for achieving low standby power without requiring special support logic. When using an older device such as an MC146818, however, the CE# pin on the external RTC normally should be connected to PWRGOOD through an inverter (such as an MC14069) that is also connected to battery power. It is particularly important not to gate CE# with any devices or signals (such as IOR# or IOW# or any other 4041 signals) that do not remain powered when main system power is turned off. An external battery-backed inverter on PWRGOOD may be unnecessary if the external RTC has an on-chip VCC detector or other built-in provision for controlling standby mode automatically.

The PSRSTB pin of the 4045 connects to the internal PS signal of the RTC. This is used to detect loss of battery power. It is externally connected to an RC circuit. When SIPC power is lost, then restored, the RC circuit causes a low to high transition on PSRSTB after power is reapplied. This resets bit 7 of Status Register D (address offset 0D) in the RTC to a 0 (Valid RAM and Time, "VRT"), indicating that the date, time, and/or CMOS RAM contents may be invalid due to loss of both system and battery power.

Table 6.2 Real Time Clock/CMOS RAM addresses

Byte	Function
00	Seconds
01	Seconds alarm
02	Minutes
03	Minutes alarm
04	Hours
05	Hours alarm
06	Day of week
07	Date of month
08	Month
09	Year
0A	Status Register A
0B	Status Register B
0C	Status Register C
0D	Status Register D
0E-3F	Used by the standard AT BIOS
40-7F	Used by the chipset specific BIOS

The 4045 adds two features to the Real Time Clock:
Password Protection
Alarm Output during power down.

6.9.4.1. Password Protection

The RTC Password protect allows software, usually contained in the ROM BIOS (for maximum effectiveness) to lock out 8 bytes of the CMOS RAM, preventing reads or writes to it. This allows a password to be stored there, which the user must type to enable the computer. At power up, the password in the RTC can be read or written. After successful log-on by the user, the password is protected, preventing someone from reading and/or changing it from DEBUG (or another program). Before it is protected, the user, after successful log-on, has the option to change or disable the password.

When Port 92 bit 3 is written to a 1, a flip-flop is set in the 4045 which disables reads and writes to locations 38 through 3F of the CMOS. Port 92 bit 3 cannot be written back to a 0. This feature can be disabled through a config register, but once the actual protection has been done it cannot be disabled by the config bit (there are no security loopholes). Only a system reset will allow accesses again.

6.9.4.2. RTC IRQ output

When PWRGOOD is low, IGNNE# becomes the open collector RTC interrupt output. If this feature is used, a pull-up should be provided on the IGNNE# pin. The pin floats until an RTC interrupt occurs, at which time the SIPC drives the pin low. This feature is always enabled. It can be used by external circuitry, if desired, to power up the system as a result of an RTC alarm. It will function at the reduced RTC battery voltage.

6.10. Address Buffers

The SIPC drives all of its address lines for DMA cycles and some of its address lines for refresh cycles. It also drives SA17:19 for all except master and master refresh cycles.

CPU or Local Master cycles. SA0:7 and A8:9 are address inputs for IO cycles. SA17:19 are driven from A17:19. All other SIPC address lines are floated and not used.

DMA cycles. SIPC address utilization during DMA is described in the DMA Controllers section.

ISA Master cycles. All address lines are floated. A0:9 are inputs, and are used for IO decode.

Hidden Refresh. SA0:7 are driven with the refresh address. This address is incremented following each refresh cycle, including master refreshes. SA17:19 are driven from the DMA Page Register (bits 1:3), which will normally be low (they are programmable). All other address lines are floated.

Master Refresh. As far as SPIC address lines are concerned, this is the same as Hidden Refresh except for SA17:19, which are floated.

CPU Snooped Write Back. When WBACK# goes low, A8:9 and A17:23 are floated if they are being driven. WBACK# may go low when any device owns the bus, but the SIPC is only affected during DMA cycles, since that is when A8:9 and A17:23 are driven.

6.11. Internal IO Decode and configuration register access.

6.11.1. Configurations Registers.

The SIPC is allocated index registers 01, and 08 to 0Fh. 08, 09, and 0Ah are implemented in the 4035. 0Bh and 0Ch are added for the 4045 for the superset functions. 01 is inside the IPC core.

The index registers are accessed through ports 22 and 23. 22 is a write only address register, and 23 is a read/write data port. The address of the register is first written to 22, then the data is read from or written to port 23. Port 22 must be written before each access to 23.

The 4045 adds the ability for SMM code to access the configuration registers through ports 26 and 27. This is required to prevent the SMM code from splitting a port 22/23 access by user code. The index value is stored separately for ports 22 and 26, and there is a separate "accessed" flag for 26/27.

6.11.2. IO decodes.

IO decoding is done from SA0:7 and A8:9. The 4045 decodes the following IO:

IPC ranges
Port 22/23 and 26/27 for configuration registers
Port 60h for resetting the mouse interrupt
Port 61h (also known as "Port B")
Port 92h
Ports F0h & F1h for the numeric coprocessor error reset

A full list of ports and their functions is given in Section 3.

Optionally, in the 4045, the SA19 pin may be converted to a chip enable input which may be used to decode upper address bits, allowing a 16 bit IO decode. The chip select function is controllable from an index register. The 4041 may optionally provide a chip select signal for the SIPC to use, which is low when A10:15 are all low.

6.12. 486 Floating point logic

The SIPC has two modes for floating point error handling. It may either do the error handling support internally or externally.

Internal mode is used for the 486, or other CPU which contains the coprocessor internally and has the FERR# and IGNNE# signals. In this mode the SIPC receives the FERR# signal and generates IGNNE#. Internally, it generates IRQ13 to the IPC megacell. There is no IRQ13 input signal in this mode. The internal IRQ13 and externally driven IGNNE# signals operate as follows:

> The falling edge of FERR# causes internal IRQ13 to become asserted, interrupting the CPU. Meanwhile, the coprocessor freezes on the error-causing coprocessor instruction.
>
> When the IRQ13 interrupt handler issues an output to F0h, an internal INTCLR# is generated. This clears IRQ13 and causes IGNNE# to become driven low, unfreezing the coprocessor so that the interrupt handler can execute further coprocessor instructions as needed for error recovery. FERR# is still low at this point.
>
> IGNNE# remains driven low until the IRQ13 interrupt handler issues a coprocessor command that results in FERR# returning to the high state.

The coprocessor normally will freeze upon attempting execution of any non-control instruction that causes an error (FERR#). Asserting IGNNE# to the coprocessor allows the error handler to execute further non-control instructions as needed for PC/AT compatible recovery from the error condition. (Control instructions generally can be executed anytime, with or without IGNNE#.)

Output to F1h is treated the same as output to F0h. INTCLR# is generated in response to either port being written to. In 286/287-based systems, output to F1h caused a coprocessor reset. This is not required in 386 and 486-class systems, but some software may still expect an output to F1h to clear a coprocessor interrupt..

External mode is used with an external coprocessor such as a Weitek coprocessor. External mode is required for the Weitek because it generates IRQ13 directly. The FERR# pin becomes the IRQ13 input pin in this mode, and the IGNNE# pin becomes INTCLR#, which is a decode of IO ports F0h and F1h. An external PAL uses these signals, as well as the coprocessor error signals, to handle the AT compatible coprocessor function.

6.13. Keyboard and Mouse Interrupts.

The 4045 and 4035 both have the ability to receive IRQ1 and the mouse interrupt (usually IRQ12) directly from an external keyboard/mouse controller. In addition, the 4045 also has the ability to receive these interrupts via the control link from the internal keyboard/mouse controller in the 4041. When the internal controller is used, the 4041 and 4045 work together to latch IRQ1 and IRQ12 until an I/O read from port 60h occurs. The I/O read from port 60h clears the interrupt. This is a feature of PS/2 keyboard/mouse compatibility. To clear IRQ1, port 60h is a 16-bit decode in the 4041. To clear IRQ12, port 60h is a 10-bit decode in the 4045 (optionally a 16-bit decode if the IOCS# feature is implemented).

In systems that use an external mouse controller, IRQ12 appears on the ISA bus and potentially can be used or monitored by other system resources. External mouse controllers sometimes drive IRQ12 with an open-collector buffer. The 4045 implements a similar capability. When the internal mouse controller is being used, the IRQ12 pin is driven by the 4045 as an output, as well as feeding back into the interrupt controller logic in the 4045. The pin is driven high when the mouse interrupt is asserted, and driven low whenever the mouse interrupt is de-asserted. The 4045 floats IRQ12 when the internal mouse option is disabled. IRQ1, unlike IRQ12, is not driven out when the internal keyboard controller is used.

The IRQ1 pin is always an input on the 4045, and the pin is ignored (bypassed via the Control Link) when using the internal keyboard controller in the 4041.

6.14. Port B and Speaker logic

The SIPC contains the following bits of Port B (IO port 61)

Table 6.3 SIPC Port B bits

Bit	IO Write	IO Read	Usage
0	Tmr Gate 2	Tmr Gate 2	Speaker circuit
1	Spkr Data	Spkr Data	Speaker circuit
2	ENB RAM PCK	ENB RAM PCK	(In the 4041/4031)
3	EN IO CK	EN IO CK	(In the 4041/4031)
4	-	REF Det	from ref det circuit
5	-	OUT2	from tmr 2 output
6	-	-	(In the 4041/4031)
7	-	-	(In the 4041/4031)

When reading port 61h, the 4045/4035 and 4041/4031 respond as follows:

> SIPC drives **bits 0-5** onto the XD bus, and the 4041/4031 passes them through transparently to the CPU local data bus. Bits 2 and 3 have no function in the 4045 other than readback. Bits 4 and 5 are read-only.
>
> SIPC drives **bits 6-7** to zero on the XD bus, but the 4041/4031 drives the proper values onto the CPU local data bus. Bits 6-7 originate in the 4041/4031 and are not implemented in the 4045/4035. Bits 6 and 7 are read-only.

In the 4045, bit 4 toggles on each Refresh Request (Timer 0 output pulse). The bit continues to toggle even when ISA refresh is disabled.

Refer to the Port 61h register description in Section 3 for further functional description of Port 61h.

6.15. Manufacturing Test Modes

The 4045 and 4035 have two test modes to support board-level manufacturing test:
Putting all outputs in a high-impedance state (Hi-Z)
Checking each pin's soldering connection (AND tree)

The Hi-Z mode is entered simply by driving PWRGOOD low. (PSRSTB should be pulsed low while PWRGOOD is low to reset the IPC functions, but this probably won't affect pin tri-stating.) While PWRGOOD is low, all outputs are placed in high-impedance state except the 14MHz output (which is driven low) and the 32 KHz output (which remains responsive to the 32 KHz input).

The connectivity test mode is activated as follows:
Drive PWRGOOD low, then pulse TEST# low momentarily (A20M# pin).
While TEST# is low, drive hex code FEh onto XD7:0. This code is latched on the rising edge of TEST#.

Timings aren't critical. Setup and hold times of 50 ns minimum should be adequate.

In the connectivity test mode, all pins (with a few exceptions) become inputs to an AND-gate tree. The final output from the AND tree is the SPKR pin. When all inputs are high, SPKR is high. When any pin is driven low, SPKR goes low. The only pins that do not become AND inputs are:

SPKR, which is the AND output.
TEST# and PWRGOOD, which are used for entering and leaving test mode.
14MX1, 14MX2, 32KX1, and 32KX2, which are specialized oscillator pins.

To terminate the connectivity test mode, pulse PWRGOOD low momentarily (or follow the activation procedure with code 00h instead of FEh).

7. Electrical Specifications

The DC electrical specifications listed below apply to both the 84041 and 84045 individually unless otherwise noted.

84041 / 84045 ABSOLUTE MAXIMUM CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply voltage	—	7	V
V_I	Input voltage	-0.5	5.50	V
V_O	Output voltage	-0.5	5.50	V
T_{OP}	Operating temperature	-25	85	C
T_{STG}	Storage temperature	-40	125	C

Note: Permanent device damage may occur if the absolute maximum conditions are exceeded. The functional operation should be restricted to the recommended operating conditions. See next table.

84041 / 84045 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CC}	Supply voltage	4041 Operating	4.75	5.25	V
V_{CC}	Supply voltage	4045 Operating	4.50	5.25	V
V_{CCSB}	Supply voltage	4045 Standby	2.2 *	5.25	V
T_A	Ambient temperature	—	0	70	C

* V_{CCSB} minimum is needed for 32.768 KHz oscillator operation and RTC/CMOS RAM data retention. V_{CCSB} applies when PWRGOOD is low.

84041 / 84045 DC CHARACTERISTICS

Symbol	Parameter	Min	Max	Unit
C_{IN}	Input capacitance	—	10	pF
$C_{I/O}$	I/O capacitance*	—	20	pF
C_{OUT}	Output capacitance*	—	20	pF
V_{IL}	Input low voltage	-0.5	0.8	V
V_{IH}	Input high voltage	2.0	V_{CC} + 0.5	—
V_{OL}	Output low voltage at max I_{OL} *	—	0.4	V
V_{OH}	Output high voltage at max I_{OH} *	2.4	—	V
V_{CL}	Clock output low at max I_{OL} *	—	0.4	V
V_{CH}	Clock output high at max I_{OH} *	V_{CC} - 0.5	—	V
I_{IL}	Input leakage current	-10	20	µA
I_{OLZ}	Output leakage current (high impedance state)	-10	20	µA
I_{CC}	V_{CC} supply current, 4041 operating	—	200	mA
I_{CC}	V_{CC} supply current, 4045 operating	—	50	mA
I_{CCSB}	V_{CC} supply current, 4045 standby mode, 3.0V	—	10	µA

* Refer to the Pin Overview tables in Sections 2.2 and 2.3 for I_{OL}, and I_{OH} specifications for each pin. C_{OUT} and $C_{I/O}$ in the above table specify the load imposed on an external signal driver when the pin is in the high-impedance (non-driving) state.

The AC Characteristics for the 84041 and 84045 adhere to the following conventions unless otherwise noted:

- Each AC timing parameter definition gives the name of the signal being specified, the name of the signal being used as a reference for the parameter, the type of relationship (setup, hold, delay, etc.), the applicable timing diagram, and the Min/Max timing limits for the parameter. ("Formula Specifications" are explained in a separate section.)
- Each timing parameter applies to all timing cycles and cases in which the signals follow the stated relationship. In any case where the cycle type significantly affects the Min/Max limits of the parameter, separate parameters are listed along with an indication of the cycle types to which they apply.
- Refer to the various functional timing diagrams throughout this document for further description of signal timing relationships and cycle types.
- Output specifications apply at the specified maximum pin load listed.
- Signals that are not listed should not be critical in a normal system implementation and are not directly specified.
- All times are in ns.

84041 AC CHARACTERISTICS - Input Requirements

Parameter	Description	Fig.	Min	Max
f_{MAX}	SCLK, CWS# frequency	—	0	33.3MHz
t1	SCLK, CWS# high time	7.5	12	—
t2	SCLK, CWS# low time	7.5	12	—
t3	CLK2 skew from SCLK	7.3	—	—
t6	SUSPA# setup before SCLK	7.3	10	—
t7	SUSPA# hold after SCLK	7.3	3	—
t8	HLDA setup before SCLK	7.3	14	—
t9	HLDA hold after SCLK	7.3	3	—
t12	SMIACT#/SMIADS# setup before SCLK	7.3	14	—
t13	SMIACT#/SMIADS# hold after SCLK	7.3	3	—
t14	ADS#, W/R#, D/C#, M/IO#, RDY#, BRDY#, BLAST#, HITM# setup before SCLK	7.3	10	—
t15	ADS#, W/R#, D/C#, M/IO#, RDY#, BRDY#, BLAST#, EADS#, HITM#, hold after SCLK	7.3	3	—
t18	D0-31, DP0-3 setup before SCLK	7.3	5	—
t19	D0-31, DP0-3 hold after SCLK	7.3	3	—

84041 AC CHARACTERISTICS - Output Responses

Parameter	Description	Fig.	Load (C_L)pF	Min	Max
t30	CLK2OUT, CPUCLK advance before SCLKOUT	7.1	30	0	2
t31	STPCLK# delay from SCLK	7.1	30	—	13
t32	ADS#, W/R#, D/C#, M/IO#, RDY#, BRDY#, FLUSH#, EADS#, SMI# delay from SCLK	7.1	75	—	20
t33	BRDY# delay from CPU address	7.1	75	—	40
t34	BRDY# delay from TAG0-10	7.1	75	—	38
t35	DP0-3 delay from D0-31	7.1	65	—	16
t36	KEN#, WBACK# delay from SCLK	7.1	30	—	12
t40	CA2 delay from CPU address	7.1	75	3	15
t41	CA2 delay from SCLK	7.1	75	3	16
t42	CA3A, CA3B delay from CPU address	7.1	50	3	13
t43	CA3A, CA3B delay from SCLK	7.1	50	3	12
t44	CWEA#, CWEB# delay from CWS# (0WS write hit)	7.1	50	3	10
t45	CRDA#, CRDB# delay from SCLK	7.1	50	3	13
t46	CWEA#, CWEB# delay from SCLK (1WS write hit)	7.1	50	3	16
t47	TAGWE# delay from SCLK	7.1	50	—	16
t48	TAG0-10 delay from SCLK	7.1	30	—	17
t60	RAS0#-7 delay from SCLK	7.1	120	—	16
t61	CAS0#-3 delay from CLK2	7.1	65	3	20
t62	DWE# delay from W/R#	7.1	240	—	20
t62a	DWE# delay from W/R#	7.1	50	—	15
t63	MA0-11 delay from CPU address (DRAM access)	7.1	200	—	40
t63a	MA0-11 delay from CPU address (DRAM access)	7.1	50	—	25
t64	MA0-11 delay from SCLK (DRAM access)	7.1	200	—	25
t64a	MA0-11 delay from SCLK (DRAM access)	7.1	50	—	15
t70	XD0-7 delay from SCLK	7.1	65	—	20
t71	XD8-15 (MA2-9) delay from SCLK	7.1	65	—	22
t72	D0-31 delay from XD0-15	7.1	75	—	20
t73	SDIR0-1 delay from SCLK	7.1	40	—	27
t74	BALE, MEMR#, MEMW#, IOR#, IOW# delay from BUSCLK	7.1	240	—	9
t75	IOCHRDY delay from ISA command during DMA or ISA Master cycles	7.1	240	—	25
t76	MEMCS16# delay from CPU address	7.1	240	—	—
t77	SBHE# delay from BUSCLK	7.1	240	—	10
t78	XA0-1 delay from BUSCLK	7.1	50	—	8
t80	IDEIOR#, IDEIOW# delay from SCLK	7.1	40	—	17
t81	IDECS0-1# delay from CPU address	7.1	40	—	23
t82	IDEEN# delay from SCLK	7.1	40	—	18

FORMULA SPECIFICATIONS

Formula specifications are provided to aid in performing worst-case timing margin calculations for actual system designs. Formula specifications refer to the minimum or maximum result of a specified calculation involving other chip parameters. For any particular chip, formula specifications usually express a "tracking" relationship over temperature and voltage for the parameters involved in the formula.

For any given test chip, formula specifications may be verified by the following procedure:

1. At selected combinations of temperature and voltage, the actual values of the parameters involved in the formula are measured.
2. The resulting formula value is then computed at each temperature and voltage measurement point. The specified formula limit applies to any 84041 chip at any combination of temperature and voltage within the rated operating range.

84041 AC CHARACTERISTICS - DRAM and Cache Specifications

Parameter	Description	Formula	Min	Max
te41	Tracking, CA2 vs. CWE# rise	t41-t46	0	—
te43	Tracking, CA3 vs. CWE# rise	t43-t46	5	—
te60	Tracking, RAS vs. CAS	t60-t61	—	10
te62	Tracking, DWE vs. RAS	t62-t60	—	5
te62a	Tracking, DWE vs. CAS	t62-t61	—	5
te62b	Tracking, DWE vs. CAS	t61-t62	—	2
te63	Tracking, MA vs. RAS	t63-t60	—	25
te63a	Tracking, MA vs. RAS	t63a-t60	—	15
te63b	Tracking, MA vs. RAS	t60-t63	—	4
te63c	Tracking, MA vs. RAS	t60-t63a	—	6
te63d	Tracking, MA vs. CAS	t63-t61	—	32
te63e	Tracking, MA vs. CAS	t63a-t61	—	15
te63f	Tracking, MA vs. CAS	t61-t63	—	4
te63g	Tracking, MA vs. CAS	t61-t63a	—	6
te64	Tracking, MA vs. RAS	t64-t60	—	10
te64a	Tracking, MA vs. RAS	t64a-t60	—	5
te64b	Tracking, MA vs. RAS	t60-t64	—	4
te64c	Tracking, MA vs. RAS	t60-t64a	—	6
te64d	Tracking, MA vs. CAS	t64-t61	—	13
te64e	Tracking, MA vs. CAS	t64a-t61	—	5
te64f	Tracking, MA vs. CAS	t61-t64	—	4
te64g	Tracking, MA vs. CAS	t61-t64a	—	6

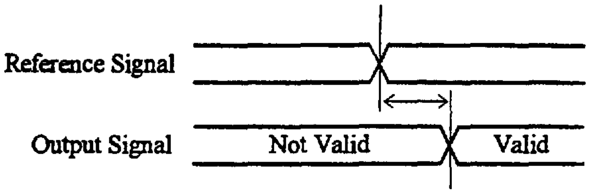
84045 AC CHARACTERISTICS - Input Requirements

Parameter	Description	Fig.	Min	Max
f_{MAX}	SCLK frequency	—	—	33.3 MHz
t100	SCLK high time	7.5	12	—
t101	SCLK low time	7.5	12	—
t104	HLDA setup before SCLK	7.3	14	—
t105	HLDA hold after SCLK	7.3	3	—
t106	WBACK# setup before SCLK	7.3	18	—
t107	WBACK# hold after SCLK	7.3	3	—
t110	LREQ#0-2 setup before SCLK	7.3	10	—
t111	LREQ#0-2 hold after SCLK	7.3	3	—
t112	IOCHRDY setup before SCLK or 14.3 MHz input	7.3	10	—
t113	IOCHRDY hold after SCLK or 14.3 MHz input	7.3	3	—

84045 AC CHARACTERISTICS - Output Responses

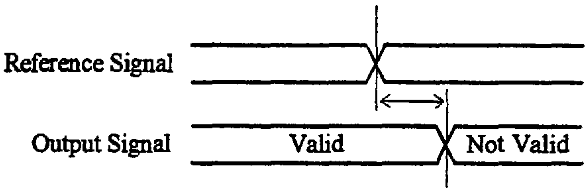
Parameter	Description	Fig.	Min	Max
t120	CPURESET, SYSRESET fall from SCLK	7.1	—	22
t121	HOLD delay from SCLK	7.1	—	19
t125	LGNT0-2# delay from SCLK	7.1	—	—
t126	REFRESH# delay from SCLK	7.1	—	30
t127	TC delay from SCLK or 14.3 MHz input	7.1	—	50
t128	DACK# delay from SCLK or 14.3 MHz input	7.1	—	50
t129	FLUSH# delay from SCLK or 14.3 MHz input	7.1	—	22
t140	MEMR#, MEMW# delay from SCLK or 14.3 MHz input	7.1	—	15
t141	SMEMR#, SMEMW# delay from SCLK or 14.3 MHz	7.1	—	15
t142	IOR#, IOW# delay from SCLK or 14.3 MHz input	7.1	—	15
t143	SBHE# delay from SCLK or 14.3 MHz input	7.1	—	—
t144	SA0-7, A8-9, SA17-19, A17-19, A20-23 delay from SCLK or 14.3 MHz input	7.1	—	70

Figure 7.1



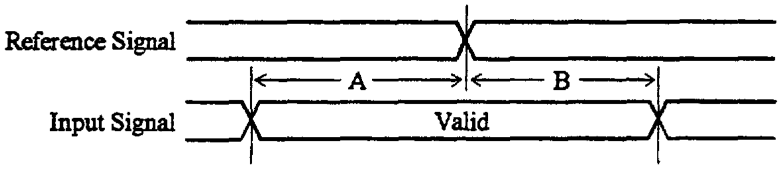
Output valid from reference.
Output delay from reference.
Output rise from reference.
Output fall from reference.
Output active from reference.
Output inactive from reference.

Figure 7.2



Output hold from reference

Figure 7.3



A = Input setup before reference
B = Input hold after reference

Figure 7.4

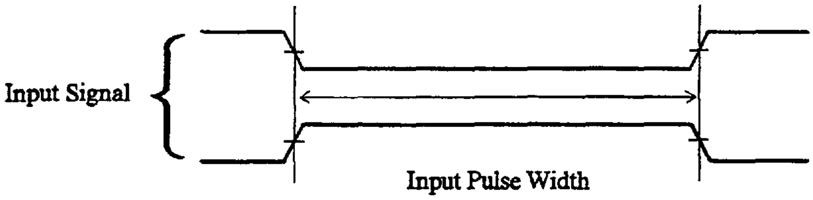
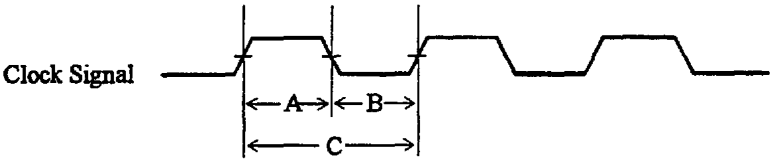


Figure 7.5



A = Clock high time
B = Clock low time
C = Clock cycle time

Timing Waveforms

8. Mechanical Specifications

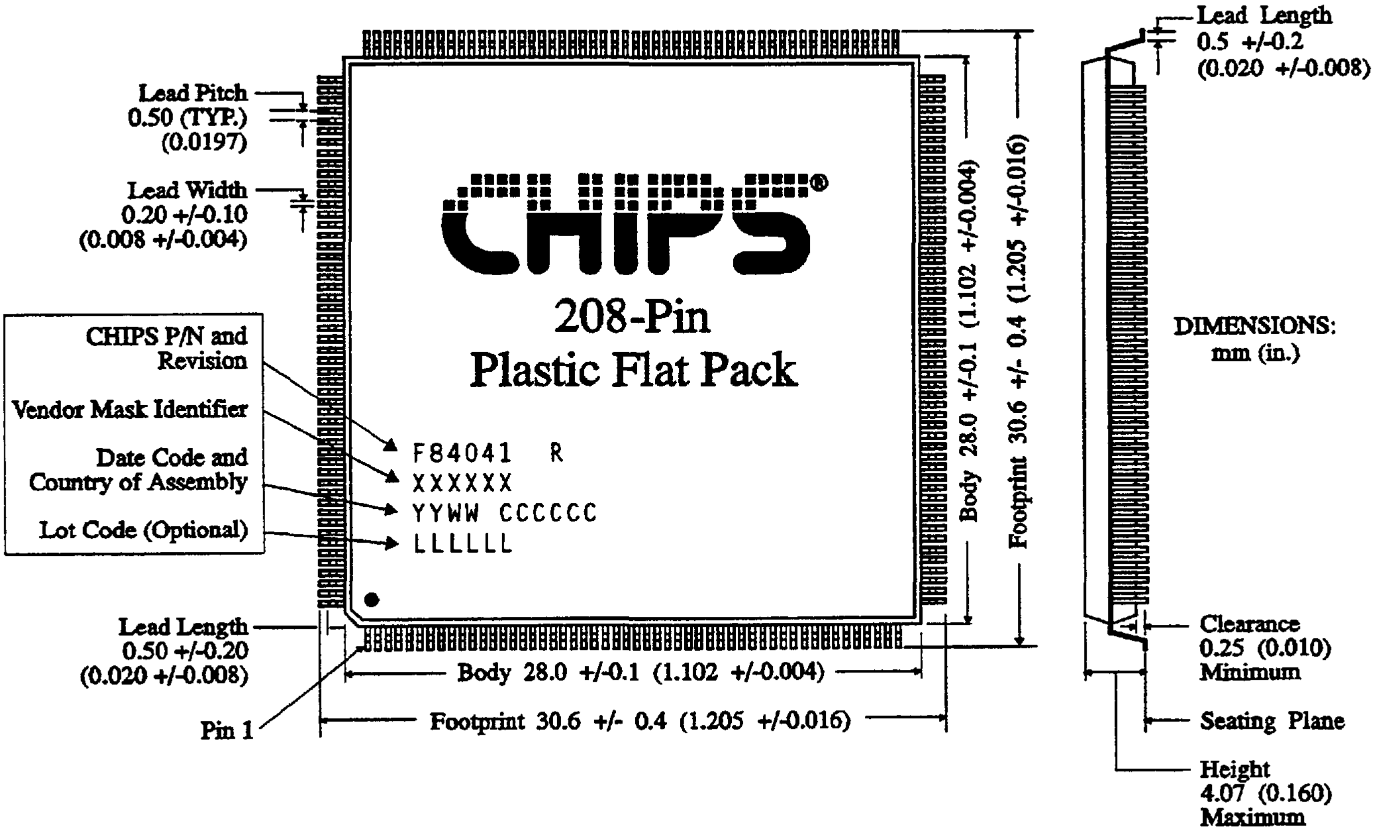


Figure 8.1: 84041 Packaging Dimensions

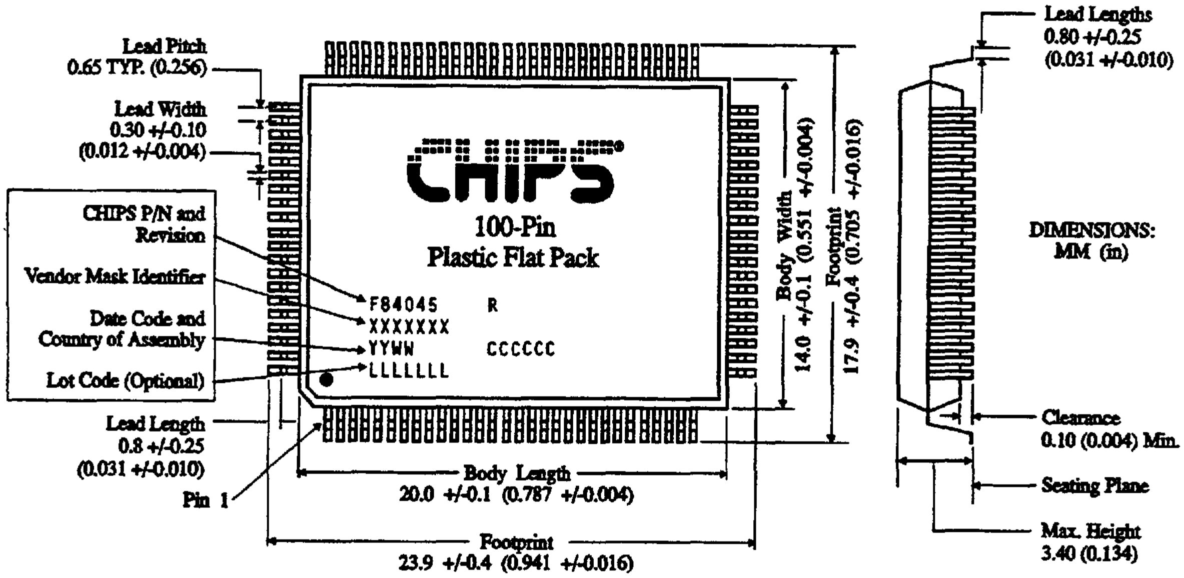


Figure 8.2: 84045 Packaging Dimensions

DBC7

012CC7-CCC Rev-11

2/11/95