P/25 ENGINEERING MANUAL

PLEXUS COMPUTERS INC

2230 Martin Ave
Santa Clara, CA 95050
408/988-1755

Plexus Publication #5018-1A
June 18, 1982

REVISION RECORD

REVISION	DATE	DESCRIPTION
1A	7/1/82	Based on information from engineering during system development. Includes relevant information up to EN 156.

TABLE OF CONTENTS

PROCESSOR

I/O CONTROLLER

DISK AND TAPE CONTROLLER

PREFACE

AUDIENCE

This P/25 Hardware Engineering Manual is intended for use by engineers and technicians who need to understand the P/25 in order to modify it, repair it or interface other equipment. It describes the P/25 down to the component level.

This is not a user's guide or a programming manual. While users and programmers might find the information useful, for day-to-day instructions, they should see the Plexus P/25 User's Manual. For programming information, they should see the Plexus V7 UNIX Programmer's Manual--vol 1, 2A and 2B.

A working knowledge of digital theory and/or computer architecture is necessary to make full use of this manual.

EQUIPMENT LEVEL

This manual is current as of July, 1982. It describes systems having both 4 and 5 Mhz system clocks and memory boards with 64K RAM chips. Earlier systems may have memory boards with 16K RAM chips.

CONTENTS

Except for the system overview, each section in this manual describes a particular P/25 PCB. Each PCB is responsible for a distinct set of system functions, so each section describes both a PCB and a set of system functions.

The sections in this manual are as follows:

System Overview -- Describes the entire system at a basic level. Read the overview first to obtain a feeling for the entire system. This will enhance understanding of any sections read later.

Processor -- Describes the Processor board, which contains the Central Processing Unit (CPU) and the Memory Management Unit (MMU). This is the controller for the entire system; each P/25 contains one and only one processor board.

Input/Output Controller -- The input/output controller is really an intelligent communication processor (ICP) that controls interaction with up to eight serial I/O ports (RS232C standard) and one parallel port (Centronics standard). Each P/25 contains one or two ICPs.

Memory -- While there is memory on every PCB in the P/25, this section describes the system main memory (dynamic RAM) which connects to the rest of the P/25 via connector P2 from the processor board. Each P/25 contains one or two memory boards.

Backplane -- The backplane connects the components of the P/25 to each other. It consists of an industry standard Multibus (P796 Standard) and connector P2 which connects system memory to the processor. An IEEE Proposed Microcomputer System Bus Standard (P796 Bus) is included in this section.

Disk and Tape Controller -- This section describes the Intelligent Mass Storage Controller (IMSC) which controls interaction with 1 to 4 SMD type disk drives and a streaming cartridge tape drive.

The figures and schematics for each section are at the end of the section.

OVERVIEW

1. INTRODUCTION

The Plexus Computers P/25 is a 16-bit multiprocessing minicomputer designed specifically to run the UNIX* operating system. It supports 512K to 2M bytes of error-correcting memory, 16 terminals (serial ports), two parallel ports (printers), one streaming cartridge tape drive and one to four SMD-interface disk drives. The standard configuration includes a 22, 36 or 72M-byte (formatted) fixed Winchester disk drive and a 20M-byte four-track cartridge tape drive.

The P/25 uses an industry standard Multibus** backplane (IEEE P796 Standard), and a storage module disk (SMD) interface. This, combined with the UNIX operating system, increases the ability of designers to rapidly integrate software and hardware to meet specific application needs.

The P/25 has from 3 to 5 asynchronous processors. The central processing unit (CPU), which executes system processes and user programs, contains a Zilog Z8001A. It is helped by at least two other processors - a disk and tape controller, and an intelligent communication processor (ICP). These peripheral controllers handle all interaction with disks, tape drives, terminals and printers independently of the CPU, using DMA to access memory. This deloads the Z8001A, leaving it free to execute instructions without handling many of the routine interrupts.

The P/25 also supports one or two user-supplied Multibus-compatible PC boards which plug into the Multibus backplane.

The standard P/25 is composed of the following major components:

Central Processor Unit
Memory Management Unit
Memory Array
Intelligent Communication Processor (ICP)
Intelligent Mass Storage Controller (IMSC)
Power Supply
Multibus Backplane
Winchester Disk Drive
Streaming Cartridge Tape Drive
Enclosure

Each of the components is described in the following sections. A

*UNIX is a trademark of Bell Laboratories. Plexus Computers Inc is licensed to distribute UNIX under authority of AT&T.
** Multibus is a trademark of Intel Corporation.

block diagram (Figure 1) shows how they are interconnected.

2. CENTRAL PROCESSOR UNIT (CPU)

Each P/25 is equipped with one processor board, which contains the central processor unit and the memory management unit mounted on a double-height Multibus card.

2.1. Processor

The CPU uses a Zilog Z8001A microprocessor which runs at 5-Mhz. The board includes latches, buffers, control lines, local peripherals and local memory to support the Z8001A. It accesses all other system boards through the Multibus (connector P1) except system memory which it accesses through connector P2.

2.1.1. Local Peripherals

The system CPU controls the following local peripheral devices located on the processor board:

Counter/Timer Circuit -- The CPU has access to a 4-channel Zilog Z80B counter/timer circuit. Channel 0 is connected to channel 1 and programmed to divide the system clock down to 50 hz. The output signal of channel 1 is used to generate realtime interrupts to the Z8001A. Channels 2 and 3 are uncommitted and available for general software use.

Programmable Interrupt Controller -- Interrupts to the Z8001A are controlled by a programmable interrupt controller (PIC), which receives interrupt requests from various devices and passes interrupts to the Z8001A based on priority.

Clock/Calender -- A clock/calender circuit (realtime clock) gives the CPU access to time-of-day in hours, minutes and seconds, and days, months and years. A battery powers it for up to three months after system power is removed.

2.2. Software

The CPU uses four of the Z8001A address spaces. Standard I/O space communicates with ports on the Multibus. Special I/O space controls peripheral circuitry on the processor board. Memory instruction space and data space are used to access memory.

The CPU uses three of the Z8001A interrupt and trap capabilities.

Nonmaskable interrupts are used for power-fail detection. Vectored interrupts inform the Z8001A that a controller on the Multibus requires service, and a segment trap informs the Z8001A that an MMU error has occurred.

The Z8001A can be reset by asserting the Multibus INIT/ line, or it can reset itself by causing the INIT/ line to be asserted. It can also reset all Multibus controllers with the INIT/ line without resetting itself.

2.3. Diagnostic Port

The CPU has direct control of a diagnostic console port which attaches to connector P3.

3. MEMORY MANAGEMENT UNIT (MMU)

The memory management unit (MMU), located on the processor board, controls access to system memory. It receives logical memory addresses from the CPU or peripheral controllers and converts them into physical memory addresses for access to the memory array. It also handles memory protection, error detection and correction, and refresh.

3.1. Page Map RAM

The MMU contains a page map RAM which, under the control of the operating system, remembers the differences between physical memory addresses and logical memory addresses. This allows different programs and processes to use the same instruction set simultaneously. The physical memory consists of 32 map sets of 32 pages, each containing space for 2K bytes.

3.2. Memory Protection

Each page can be assigned a read-only and an invalid status. These protect critical data from illegal instructions. Memory accesses which do not match the page status are not allowed.

3.3. Error Detection and Correction

The MMU contains an AMD 2960 error detection and correction circuit which generates checkbits based on a modified hamming code. When data is stored in memory, a set of checkbits are stored with it. When the data is retrieved, a new set of checkbits are generated and compared with the old set. It they are not identical, it attempts a correction and sends the proper signal to the CPU.

3.4. Refresh

Refresh circuits for system memory are located on the processor board. It provides an 8-bit refresh address every 14.4 microseconds using a cycle-stealing technique.

4. MEMORY ARRAY

The P/25 memory array consists of three types of memory: dynamic RAM, EPROM, and static RAM. The dynamic RAM is located on separate printed circuit boards, while the EPROM and static ROM are located on the processor board.

4.1. Dynamic RAM Array

Dynamic RAM is the major component of system memory. It occupies physical addresses from 100,000 to 300,000H when fully equipped, and is the only system memory used during normal operation.

The dynamic RAM boards each contain 256K, 512K or 1024K bytes of memory arranged in 2-byte words. Each word consists of 16 bits of data and six error-correction bits. Each board also contains basic support functions for the RAM, including address decoders, buffers and RAS/CAS multiplexors. All control, address and data lines for the RAM array come from the MMU over a dedicated bus on connector P2. Although an array board occupies a slot in the Multibus backplane, it takes only power and ground from the Multibus connector.

4.2. EPROM

The EPROM contains the bootstrap instructions used to load the operating system. It also contains data used for factory testing and diagnostics. It occupies physical memory addresses from 0 to 3fff.

Four sockets for EPROM are located on the CPU board. The sockets accommodate either 2K x 8 (2716) or 4K x 8 (2732) MOS EPROMS. The minimum EPROM configuration is 2K words (two 2716 EPROMs) and the maximum is 8K words (four 2732 EPROMs). A jumper on the board is used to select the EPROM size. The access time is 450 nsec. A single wait-state is added when the CPU accesses the EPROM.

No error correction or detection is performed on accesses to EPROM.

4.3. Static RAM

The CPU has 1K words of static RAM which occupies addresses from 8000 to 87FE. It is implemented with four 1K x 4 static RAMS (2114). The RAM is byte-addressable and is used principally for diagnostic and initialization routines.

The static RAM has a 450-nsec access time, so a single wait-state is added when the CPU accesses static RAM. No error correction or detection is performed on accesses to static RAM.

5. INPUT/OUTPUT CONTROLLER

The input/output controllers are intelligent communication processors (ICPs) that handle data movement between terminals and printers and system memory. They use direct memory access (DMA) to bypass the CPU, freeing it from handling I/O interrupts. Each ICP controls eight serial ports (terminals) and one parallel port (printer). Each P/25 is equipped with one or two ICPs.

The ICP is a double-height Multibus card that features a Zilog Z8002 microprocessor.

5.1. Serial Ports

The eight RS232C serial ports are implemented with Z80A SIO/2 USARTS. They support baud rates from 50 to 38.4K hz by programming a Zilog CTC Counter/Timer. Each channel has its own DMA channel to move data from local memory to the transmit line. A DMA transfer is initiated by the SIO when its transfer buffer becomes empty. A rotating priority scheme ensures that the most recently serviced SIO has the lowest priority.

5.2. Parallel Port

The ICP has a single DMA-supported parallel port used to connect an industry-standard (Centronics-type) printer. Control logic associated with the port generates all the handshake signals necessary for the DMA to transfer data to the printer without intervention from the CPU.

5.3. Local Memory

Each ICP is equipped with 32K bytes of dynamic RAM, including parity bits and refresh. This is used to store operating instructions for the ICP and to buffer data between transfers.

5.4. Pseudo DMA

The ICP uses pseudo DMA to move data from its local RAM to system memory. The Z8002 initiates block moves between the local RAM and a 16K section of its memory space that is mapped into main memory. This can run in the background, without monopolizing either the Multibus or the local bus.

6. INTELLIGENT MASS STORAGE CONTROLLER (IMSC)

The IMSC is an intelligent disk and tape controller that contains an onboard Z8001 microprocessor. It receives commands from the CPU to move blocks of data between system memory (RAM) and the disk drives or tape drive. It has 128K bytes of local RAM which it uses to buffer entire disk tracks, thus decreasing the number of real disk accesses when two different processes require constant access to the disk. The buffers store the information from the disk and pass it to each process as if it were the only process using the disk.

The disk interface is an industry-standard SMD type. The tape drive is intelligent, performing many of the functions normally required of the tape controller. It communicates with the IMSC over eight data lines and eight control lines.

The IMSC has three distinct DMA circuits on board. A Z80 DMA chip controls the movement of data between the tape drive and the IMSC local RAM. The disk sequencer provides DMA between the disk drives and the IMSC local RAM. The IMSC processor (Z8001) uses its block move capability to move data from local RAM to or from the portion of its memory space which is mapped into the Multibus memory space. This is not a true DMA because the initiating device (the IMSC processor) stays involved.

7. MULTIBUS BACKPLANE AND INTERFACE

Most of the circuits in the P/25 are on PC cards that plug into a Multibus backplane. The backplane has 6 slots, and can contain one or two memory array boards, three bus masters and a processor board.

The only active circuits on the Multibus backplane are bus priority resolution circuits that arbitrate control of the Multibus, and a bus clock generator that provides the 10-Mhz signal used by bus masters to synchronize their bus arbiters.

Each standard-equipped P/25 contains one processor board, one or two intelligent communication processors (ICPs), one or two memory array boards (dynamic RAM) and a disk and tape controller board. This leaves room one or two user-selectable Multibus boards.

All PC cards except for the memory array are attached to connector P1 which is described in the _IEEE Proposed Microcomputer System Bus Standard_ (_P796 Standard_). The memory array boards attach to connector P2 and the diagnostic console attaches to connector P3.

In the P/25, the CPU acts like a bus master. The IMSC and the MMU (physically located on the processor board) act like bus masters or slaves. User-selectable boards may be either masters, slaves or both.

The P/25 Multibus uses a subset of the parallel arbitration technique described in the _P796 Standard_, section 2.4.2.2. Once a master is granted control of the Multibus, it maintains control by holding the BUSY/ line active. The amount of time a master is allowed to hold the Multibus has a serious effect on system performance and can be adjusted.

The bus priority out (BPRO/) and common bus request (CBRQ/) signals are not used by the P/25 standard units, but may be used by user-selectable I/O controllers.

8. POWER SUPPLY

The P/25 is equipped with a multiple-output, switching power supply which provides overvoltage and short-circuit protection. It supplies the following levels:

Volts		_Amps_
+5	at	25
+12	at	2.5
-12	at	1.7
+24	at	4
-5	at	1

9. RESET, POWER FAIL AND RESTART

The entire P/25 can be reset by asserting the INIT/ line on the Multibus. This line is controlled by three sources:

(1) front-panel reset button

(2) a control bit that can be set by the CPU

(3) the power supply's AC line monitoring circuitry.

The AC line monitor sends a power fail interrupt (PFIN/) to all devices on the Multibus when the line voltage drops below specification. The CPU responds to this interrupt and, after performing some housekeeping, resets the system by asserting the INIT/ line. Other processors in the system can use the PFIN/ internally.

When AC power is restored to the system, no effort is made to restart the system to its state prior to power failure because there are no provisions to save the contents of memory during the outage. Instead, when failure due to a power outage cannot be tolerated, an uninterruptible supply (UPS) for the AC line is available.

NOTE: The P/25 power supply is equipped with brownout protection, which provides uninterrupted power to the P/25 for 16 milliseconds in the event of a transient AC power failure.

When power is applied to the system, the INIT/ line is asserted for about 100 milliseconds. This assures that DC power is stable before the system is placed in operation.

10. STANDARD PERIPHERAL CONFIGURATION

The IMSC supports a number of industry-standard disk and tape drives, so it is possible to configure a wide variety of systems. However, the combination of a fixed Winchester disk and streaming cartridge tape drive has proven to be a very flexible and cost-effective combination, and is therefore offered as the standard.

10.1. Disk Drive

The P/25 comes equipped with one of three standard 8-inch fixed Winchester disk drives. These are compared in the following table:

Storage Capacity	Average Positioning Time (Head Positioning + Latency)
22 megabytes	38.55 milliseconds
36 megabytes	38.55 milliseconds
72 megabytes	28.3 milliseconds

Up to 3 additional disk drives may be added in a separate cabinet. The total disk capability is 336 megabytes.

(For further information on the drive, see the Plexus Winchester Disk Manual for the appropriate size disk drive.

10.2. Tape Drive

The P/25 comes equipped with a 20M-byte streaming cartridge tape drive. It has 8000 bpi storage density on 8 tracks, and can store 20M-bytes in 3.7 minutes.

(For further information on this drive, see the _Plexus Cartridge Tape Drive Manual_.

11. SPECIFICATIONS

The P/25 with its standard peripherals is packaged in a 19-inch wide, 24-inch deep, 10.5-inch tall tabletop cabinet which can fit in a standard 19-inch RETMA rack.

11.1. Processor Specifications

Electrical

95 - 130 volts AC @ 4a
190 - 250 volts AC @ 2a

Environmental

temperature:	5C to 38C (41F to 100F)
relative humidity:	20% to 80% noncondensing

12. BIBLIOGRAPHY

(1) Advanced Micro Devices AmZ8000 Family Data Book (AMD #AM-PUB098)

"Am9512 -- Floating Point Processor"

(2) Advanced Micro Devices Bipolar Microprocessor Logic and Interface Book (AMD #MLI-495)

"Am2960 -- EDC Cascadable 16-Bit Error Detection and Correction Unit"

(3) Intel Component Data Catalog (Intel #AFN-01300A-1)

"8237/8237-2 -- High Performance Programmable DMA Controller"

"8259A/8259A-2/8259A-8 -- Programmable Interrupt Controller"

(4) OKI Semiconductor MSM5832 -- Microprocessor Real-Time Clock/Calendar

(5) Signetics Z661 UART Product Specification

(6) Zilog Z8000 CPU Technical Manual (Zilog #00-2010-C)

(7) Zilog 1981 Data Book

"Z8420 -- Z80 PIO Parallel Input/Output Controller"

"Z8430 -- Z80 CTC Counter/Timer Circuit"

"Z8440 -- Z80 SIO Serial Input/Output Controller"

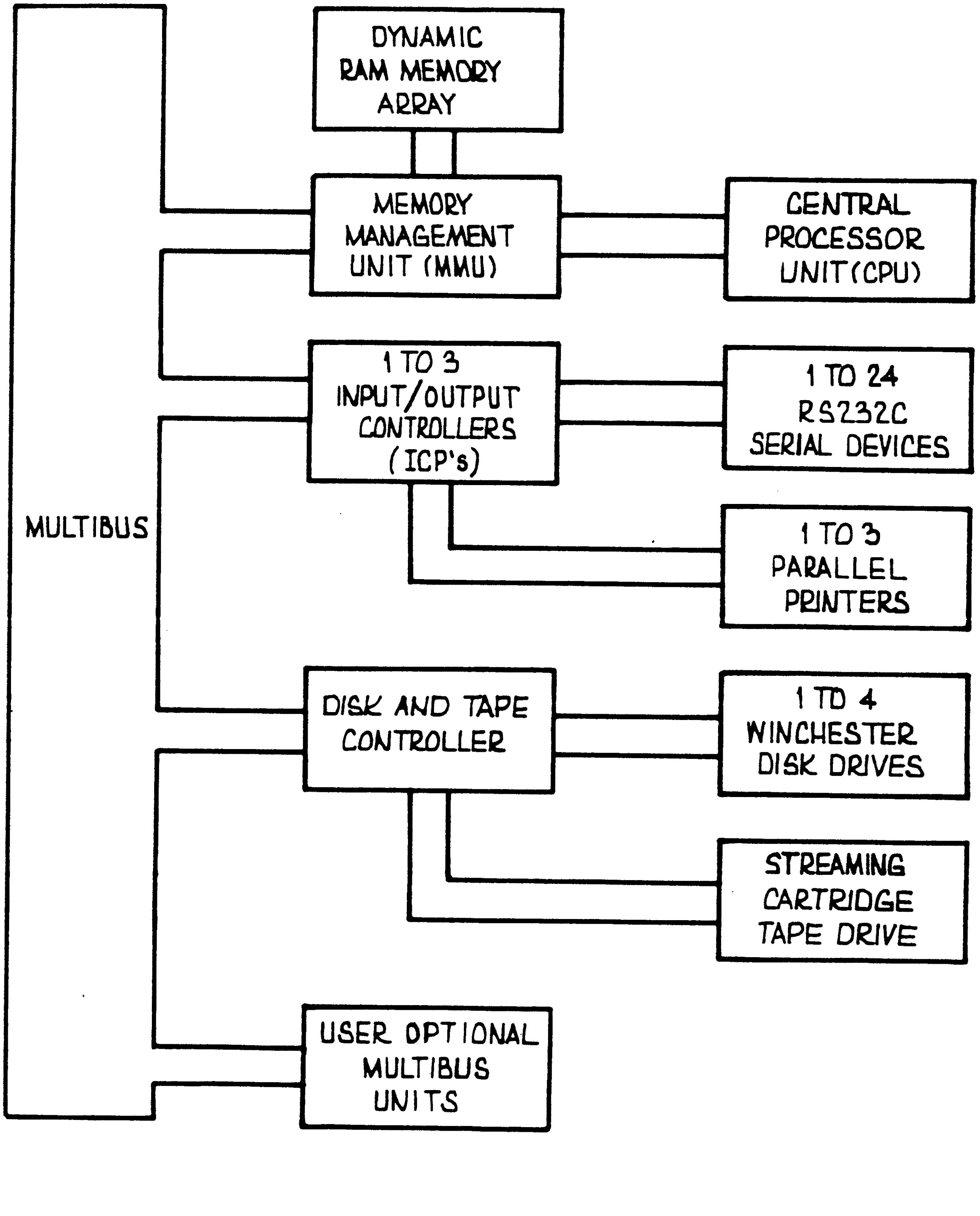

DYNAMIC RAM MEMORY ARRAY
MEMORY MANAGEMENT UNIT (MMU)
CENTRAL PROCESSOR UNIT(CPU)
1 TO 3 INPUT/OUTPUT CONTROLLERS (ICP's)
1 TO 24 RS232C SERIAL DEVICES
1 TO 3 PARALLEL PRINTERS
MULTIBUS
DISK AND TAPE CONTROLLER
1 TO 4 WINCHESTER DISK DRIVES
STREAMING CARTRIDGE TAPE DRIVE
USER OPTIONAL MULTIBUS UNITS

Figure 1. Block Diagram of the P/25

PROCESSOR

1. Z8001A MICROPROCESSOR

The P/25 CPU is built around the Zilog Z8001A 16-bit microprocessor. Nearly all features of the Z8001A are used, including separate instruction and data space, user/system modes, and nonvectored and vectored interrupts.

The Z8001A in the P/25 has 16 general-purpose registers and 110 distinct instruction types. For detailed descriptions of the registers and instructions, refer to the Zilog Z8000 CPU Technical Manual.

The functional blocks on the processor board are shown in Figure 1. All block references in this section are to Figure 1.

1.1. Programming Information

The following programming information is necessary to understand the functioning of the Z8001A CPU:

(a) Address Spaces -- All four address spaces provided by Z8001A are used in the P/25. The normal I/O space is used to communicate with I/O ports on the Multibus. The special I/O space is used to control the peripheral circuitry on the processor board. Memory instruction and data spaces are used to access the corresponding locations in main memory.

(b) User and System Process -- When a user process is running, the Z8001A is always operated in normal mode and unsegmented mode. But when a system process is running, the Z8001A operates in system mode and either unsegmented mode or segmented mode. Thus, user programs can only access one memory segment but system programs can access all memory segments. One segment occupies 64K bytes of address space in Z8001A.

(c) Data Types -- The Z8001A supports manipulation of eight data types. Each data type is supported by a number of instructions which operate upon it directly. These data types are:

Bit

Binary integer

Logic value (boolean)

Address

Decimal integer

String of bytes

String of words

Stack of words

(d) Byte and Word -- Both byte and word operations are provided, enabling the Z8001A to transfer data to an 8-bit master on the Multibus and enabling a Multibus master to access the 16-bit memory system.

(e) Segments -- There are 128 segments available in the Z8001A memory address space, providing a maximum address space of 8M bytes. In the P/25, the top segment (seg 127) is allocated to the memory mapper to reduce overhead when swapping processes. The lower segment (seg 0) is allocated to the local CPU memory (EPROM and static RAM).

1.2. Hardware Interface

The following Z8001A lines control peripheral hardware functions:

(a) Address/Data lines -- These 16 address/data lines are connected from the Z8001A to the local multiplexed address/data bus and through it to the other local buses.

(b) Segment lines(SN0 to SN6) -- In addition to their normal function, the least significant segment line (SN0) is ORed with ST2 to select either program or data address space.

(c) WAIT/ -- This line is pulled low, extending the transaction time, when any of the following occurs:

1) EPROM and static RAM read or write

2) Multibus I/O port read or write

3) Byte write to dynamic RAM.

(d) Segment Trap -- A segment trap is generated by the error handler when a memory mapping error or a EDC error output is detected.

(e) Nonmaskable Interrupt (NMI) -- This input is connected to POWER FAIL line on the Multibus.

(f) Nonvectored Interrupt (NVI) -- This input is reserved for diagnostics and is provided on connector P3.

(g) Vectored Interrupt (VI) -- Two Intel 8259A interrupt controllers (one master and one slave) provide 15 levels of interrupt.

(h) Multi-Micro (MO/) -- This Z8001A output pin controls the error detection and correction (EDC) single-bit interrupt circuits.

NOTE: The MO/ output is not used in earlier models of the processor unit.

(i) STOP/, BUSRQ/ -- Not used.

1.3. I/O Access

1.3.1. Standard I/O Space

When a standard I/O instruction is to be executed, the Z8001A requests control of the Multibus through the local bus arbiter. When control is granted, the CPU places the I/O port address on ADR0 to ADR15, and asserts the proper read or write control line (IORC/ or IOWC/). The addressed multibus port executes the command and returns an acknowledge (XACK/). The Z8001A allows the command execution to complete, then releases the Multibus. The proper number of wait-states are inserted automatically when standard I/O instructions are executed.

The CPU uses only word instructions when accessing Multibus ports (BHEN/ must be low). If the port supports only byte operations, the high-order byte is undefined when the port is read and ignored when the port is written.

The CPU can maintain control of the Multibus between I/O instructions by setting bit A1 of the control port PIO (7F88H) high. This causes the bus arbiter to hold the bus until the bit is reset. Until then, no other bus masters can use the bus. This is used primarily for diagnostic fault isolation and to test and set shared resource semaphores.

1.3.2. Watchdog Timer

A watchdog timer forces the completion of a CPU-initiated I/O cycle if the CPU does not receive XACK/ within 20 microseconds. This prevents the system from being disabled if the CPU executes a standard I/O instruction to a nonexistent port or if the device at the port fails to return XACK/. If the timeout occurs, a vectored interrupt is sent to the CPU, which must clear the interrupt by performing a special output (SOUT) to port 7FB3H.

1.3.3. Special I/O Space

The Z8001A special I/O space is used to control local (processor board) peripherals, and for diagnostic and housekeeping functions. As with the standard I/O space, only word instructions are allowed. For more information on the uses of special I/O space, see the Processor section 3, Local Peripherals.

1.4. Trace Mode Capability

To aid in software debugging, the CPU has trace mode capability, which generates a vectored interrupt to the Z8001A when the CPU fetches the first word of an instruction in normal mode. This is useful for single-stepping through programs. This interrupt must be reset by performing an SIO write to port 7FB2H. To shut off the trace mode, the CPU must mask off the trace mode interrupt in the interrupt controller.

1.5. Controls and Indicators

The processor board is equipped with four DIP switches. Three of the switches select the baud rate for the diagnostic console and the fourth selects automatic boot mode (the system re-initializes itself automatically after powerup or reset).

NOTE: The switch and LED functions are set by system software and are subject to change.

The processor board also has four LEDs (CR4 to CR7) which provide diagnostic information during powerup and system status information during normal operation. The LEDs indicate system status as follows:

CR 4 -- ON indicates a disk access

CR 5 -- ON indicates a tape access

CR 6 -- Not used

CR 7 -- ON indicates user access.

1.6. Circuit and Timing Analysis

A basic timing diagram of the Z8001A for both read and write operations is shown in Figure 2.

2. MEMORY MANAGEMENT UNIT (MMU)

The MMU is physically located on the processor board. Its basic functions are: a) to map logical address space into the physical address space and b) to check each memory reference for consistency with the attribute bits assigned to each page.

2.1. Memory Mapping

2.1.1. Address Mapping Scheme

Address mapping allows the P/25 to distinguish between logical and physical memory addresses. A physical address is an actual location in the memory array. A logical address is a location within a program whose physical address is relative to the physical address at which the program started. For example, logical address 132 in a program stored beginning at physical address 16000 is at physical address 16132.

The page mapper acts like a file clerk, placing logical files in physical locations and remembering where they are located. All programs and processes are mapped; the only direct access of physical memory occurs during powerup.

When the Z8001A is accessing memory, the lower five segment lines (SN0 to SN4) and the five highest offset address lines (A15 to A11) are input to the page map RAM, which produces the 12 high-order bits of the physical address. The lower five segment lines select one of 32 page maps, and the high five offset address lines select pages within the map. Each map contains from one to 32 pages, and each page contains 2K bytes of memory. The address within each page is selected by the lower 11 offset address lines (A0 to A10), which are unchanged by the page-mapping process.

The lowest segment line (SN0) is ORed with the Z8001A instruction/data line (ST2), so that while in instruction mode, only odd-numbered map sets can be accessed. This way, a program running in unsegmented mode (i.e. a system program) can access a user program but a user program cannot change a system program. This allows several user programs to use the same system program without being able to change it.

The Multibus accesses memory through the Multibus map RAM, which can select one of eight page maps. The output of the Multibus map RAM is used in place of the segment lines (and I/D bit) as input for the page map RAM.

Both arrangements are shown in Figure 3.

A typical process or program requires two map sets; one pointing to instructions and one pointing to data. The minimum memory for a single program is 4K bytes and the maximum is 128K bytes (2K to 64K for instructions and 2K to 64K for data). Up to 16 processes can be resident in memory simultaneously.

Several processes can execute the same program simultaneously. When this occurs, they share the same physical instruction space but each has its own data space and its own map set. Each map points to the same instructions in physical memory and to a separate data space in physical memory.

2.1.2. Powerup And Reset

When the P/25 is powered-on or reset, the MMU is disabled and physical memory addresses are generated by concatenating sixteen Z8001A address lines to the seven segment lines. This is required since the Z8001A starts loading the program counter and control word from address 2 in segment 0.

2.1.3. Programming Information

Before enabling the MMU, the Z8001A must load the page map RAM using either special I/O instruction or memory instructions. The page map appears as 1024 contiguous 16-bit ports which can be read or written using special I/O instruction. These ports are located at even addresses from 8000H to 87FEH. The same map location can also be addressed as 1024 contiguous word memory locations between 8000H and 87FEH in segment 127. Logically, these addresses can be thought of as 32 map sets each containing 32 pages, as shown in Figure 4.

Like the page map, the Multibus map must be loaded by the Z8001A before any controllers can access memory through the MMU. This map appears as eight contiguous 8-bit special I/O ports from FF88H to FF8FH. The format of the Multibus map is shown in Figure 5.

After the Multibus map has been loaded, the CPU enables the map by performing a special output to port FF90H.

The Z8001A can prevent Multibus controllers from accessing memory by writing to port FF98H. However, this is not necessary when the Z8001A is changing the page map or the Multibus map.

2.2. Memory Protection

The MMU checks each memory reference to ensure that it is consistent with the attribute bits assigned to the page. Each page can be assigned a read-only and an invalid status. If a page has its read-only bit set, the Z8001A or Multibus controllers cannot write to that page. The invalid bit indicates that the page is not resident in physical memory. In the event of an illegal memory reference, the action taken by the MMU depends on the current memory cycle and map attribute bits as follows:

Z8001A=0 Multibus=1	S=0 N=1	R=1 W=0	Read Only	Page Invalid	
0	0	0	0	1	Generate segment trap
0	0	0	1	1	Generate segment trap and inhibit write
0	0	1	-	1	Generate segment trap
0	1	0	1	0	Generate segment trap and inhibit write
0	1	0	0	1	Generate segment trap
0	1	0	1	1	Generate segment trap and inhibit write
0	1	1	-	1	Generate segment trap
1	1	0	1	0	Generate interrupt and inhibit write
1	1	0	0	1	Generate interrupt
1	1	0	1	1	Generate interrupt and inhibit write
1	1	1	-	1	Generate interrupt

If the illegal reference was generated by the Z8001A, the MMU asserts the Z8001A segment trap line. If the violation was caused by a Multibus controller, the MMU sends a vectored interrupt (MMUINT) to the Z8001A. Both are handled by the error handler (block 43).

When a memory violation occurs, the following information is stored in latches that can be read by the CPU:

(a) Address and map set numbers which caused the violation

(b) Status information

(c) The address of the first word of the last instruction fetched before the error occurred (Z8001A violations only).

Subsequent errors do not update these latches until the Z8001A resets the error condition by performing a write to port FFA0H. The format and address of this information is shown in Figure 11.

2.3. MMU Circuit Analysis

The MMU is shown in Figure 1 as blocks 3, 4, 8, 9, 13, 14, 15 and 41.

2.3.1. Normal Operation

The lower five segment lines and the highest five offset address lines are input to the page map RAM, where they select a page map entry. The page map RAM outputs a 12-bit physical memory address which passes through the buffer in block 15 to reside on connector P2. The offset address passes through block 14 to also reside on connector P2. During normal operation (in map mode), blocks 14 and 15 are enabled and block 13 is disabled. During reset (nonmap mode), block 13 is enabled and the physical address passes to connector P2 without changing.

NOTE: The 11-bit offset leaves block 4 with only 10 bits. The 11th (low-order) bit is always a zero, which places it on a word boundary.

2.3.2. Loading and Reading the Page Map RAM

The page mapper is a high-speed static RAM with 55-nsec access time.

When the Z8001A wants to load or read the page map RAM, it issues a memory reference to segment 127. The memory address decoder (block 39) generates a SEG127 signal (active high, U152 pin 10) which prevents the local bus arbiter from pulling MEMRQ/ low. (A memory cycle won't be started.) Instead, the peripheral interface controller (block 10) decodes the offset address and generates the PGMAPIOSEL/ signal (U166 pin 6). The static RAMs in the page mapper are selected and a read or a write operation is performed. The MMU controller is responsible for generating WE2148/ (sh 20) and CYCENDIO/ (sh 9). CYCENDIO/ signals the end of a cycle to the local bus arbiter.

When the page mapper is accessed as special I/O ports, the above sequence is the same except that the SEG127 signal stays inactive (LOW).

The Multibus mapper is a high-speed bipolar RAM inverting memory (74S189) which must be loaded with complemented (inverted) data. The Z8001A can only write to the Multibus map; the contents in the Multibus map are not readable. The MMU controller generates WE189/ (sh 20), CYCENDIO/ (sh 9) and the peripheral interface controller generates MBMAPCS/ (sh 9) to perform a Multibus map write operation.

2.4. Address Latches and Bidirectional Data Buffers

Four local buses are provided on the processor board. These are the:

(a) Multiplexed Address/Data Bus (MA/D) -- Passes data between the Z8001A and the other local buses through the buffers in blocks 3, 4, 5 and 7.

(b) Local Peripheral Bus -- Passes data between the peripheral devices on the processor board and the MA/D bus.

(c) Local Address Bus -- Passes data between the MMU, local memory and system memory. These leave the board in two groups; PADR 0 to PADR 9 and PADR A to PADR 15.

(d) Local Memory Data Bus -- Passes data between the MA/D bus (through buffer block 5), the page mapper (block 8) and the Multibus data lines (DAT 0 to DAT F) or the system memory data lines.

All address lines from the multiplexed address/data bus to the other local buses are latched at the trailing edge of AS/. Blocks 3 and 4 latch the logical segment lines and offset address lines from the Z8001A. Block 6 latches the I/O port address to select local peripherals. Block 5 contains the 16-bit bidirectional buffers between the multiplexed address/data bus and the local memory bus. These communicate between the Z8001A and system memory (or the Multibus). Block 7 contains the data buffers to communicate with peripherals on the processor board.

2.5. Wait-State Generator

Whenever a Z8001A read/write cycle cannot be completed within a period of three clock cycles, the wait-state generator (block 40) pulls WAIT/ line LOW and inserts a proper number of wait-states to finish the cycle. In the Z8001A, I/O transactions are at least four clock cycles long. The wait-state is not inserted when local peripherals are being accessed. The number of wait-states inserted for different types of Z8001A cycles are listed below:

EPROM and static RAM read or write	1
Byte write to dynamic RAM	1
Multibus I/O port read or write	20 usec max

2.6. Status Latches and Decoder

The read-write (RW/), byte-word (BW/) and system-normal (SN/) control lines are latched by the status latch decoder (block 2). It also decodes the status lines (ST0 to ST3) to generate the following signals:

DATAMEM	-- Data memory
STDIO	-- Standard I/O
SPECIO	-- Special I/O
VIACK	-- Vectored interrupt acknowledge
SEGTRAPACK	-- Segment trap acknowledge

2.7. Memory Access Arbitration

The local bus arbiter receives requests for main memory from the CPU through the CPUREQ line (see the processor schematics, page 17, U55, pin 6), from Multibus controllers through the MBREQ line (see the schematics, page 12, U105, pin 5) and for refresh from the REFREQ line (see the schematics, page 20, U101, pin 9). On a first-come-first-served basis, the arbiter asserts the grant line (ZGRANT, MBGRANT or REFGRANT) to the winning device. When requests from two different devices are received, the arbiter interleaves the GRANT outputs in the following order: Z8001A, refresh and Multibus.

3. LOCAL PERIPHERALS

3.1. Introduction

The peripherals on the processor board include a parallel input/output controller (Z80-PIO), a counter/timer circuit (Z80-CTC) a realtime clock, a master and slave programmable interrupt controller (PIC) and a diagnostic port (USART). The board also includes four LEDs and a selector switch (4 STSP DIP).

NOTE: Older processor boards (4-Mhz) are equipped with a floating-point processor. This has been replaced with software on the newer (5-Mhz) boards.

All local peripherals communicate with the Z8001A via the local peripheral bus and are addressed as ports in the Z8001A special I/O address space. The port addresses (in hexadecimal) are as follows:

PIO	-- 7F88 to 7F8F
CTC	-- 7F80 to 7F87
Realtime clock	-- Controlled by PIO
Master PIC	-- 7F90 to 7F97
Slave PIC	-- 7F98 to 7F9F
USART	-- 7FA0 to 7FA7
Status Switches	-- 7FA8 to 7FAF

3.2. Peripheral Interface Controller

The peripheral interface controller generates chip-select signals and control logic for various peripheral devices on the processor board. It is actually a collection of chips, most of which are shown on page 9 of the processor shcematics. It appears in Figure 1 as block 10.

The peripheral interface controller reads the address/data lines (AD0 to ADF) and the AS/ and DS/ signals from the Z8001A. When it sees the address of a chip under its control, it generates the proper chip-select signal (SEL), using the AS/ and DS/ to ensure proper timing.

It uses the R/W signal from the Z8001A to generate RDZ80/ which causes the CTC, PIO and USART to read when it goes low. It also generates IORQZ80/ which the CTC and PIO use for their IORQ/ inputs.

The following table shows the chip-enable signals and the devices they control:

Device	Signal
CTC	CTCSEL
PIO	PIOSEL
Master PIC	8259MASTSEL
Slave PIC	8259SLVSEL
USART	UARTSEL
Status Latches	STATUS0SEL to STATUS3SEL and STATWSEL
Multibus Map	MBMAPSEL
Page Map	PGMAPIOSEL
Error Detector (EDC)	2960SEL

3.3. Parallel Input/Output (PIO) Controller

The Z8001A controls various local functions through the PIO chip. The PIO has three 8-bit ports. One port communicates with the Z8001A through the local peripheral bus. The other two ports (A and B) connect to the devices controlled by the PIO.

Port A is set for output only; the PIO receives instructions from the Z8001A and outputs the data it receives on Port A. Port B is programmed for input and output; the PIO receives instructions from the Z8001A and outputs them on port B, but port B also receives data and outputs it to the Z8001A. The port assignments are shown in the following table:

NOTE: Port A and B pin assignments are shown on page 6 of the processor schematics.

PIO Port A bit	I/O	Function	Active Level
0	O	Software Reset	Low
1	O	Multibus Lock	High
2	O	Enable Memory Map	Low
3	O	Enable Error Handler Trap	Low
4	O	MSM5832 Hold	High
5	O	MSM5832 Head	High
6	O	MSM5832 Write	High
7	O	Z8001A Reset Protect	Low

NOTE: The MSM5832 is the CMOS calender clock chip described in part 3.4. of this section.

PIO Port B bit	I/O	Function	Active Level
0	I/O	MSM5832 Data 0 and LED 0	Low
1	I/O	MSM5832 Data 1 and LED 1	Low
2	I/O	MSM5832 Data 2 and LED 2	Low
3	I/O	MSM5832 Data 3 and LED 3	Low
4	O	MSM5832 Address 0	N/A
5	O	MSM5832 Address 1	N/A
6	O	MSM5832 Address 2	N/A
7	O	MSM5832 Address 3	N/A

Port B, bits 0 through 3 serve a double purpose: they take data from the realtime clock and output it to the Z8001A, and they take data from the Z8001A and output it to the status LEDs. When they are outputting data to the LEDs, the MSM5832 HOLD line is activated. For more on the status LEDs, see Processor section 1.

3.3.1. Programming Information

To control each PIO output port, the Z8001A must write two instructions. The command instruction determines the mode in which the port operates, and the data instruction supplies the output data. Port A is set to operate in mode 0 (output mode), and port B is set to operate in mode 3 (control mode), because it transfers data in both directions.

Each instruction goes to a unique port address within the PIO address space as follows:

Port A	Command	7F8A
Port A	Data	7F88
Port B	Command	7F8B
Port B	Data	7F89

See the _Zilog 1981 Data Book_, "Z80-PIO Product Specification" for additional information.

3.3.2. I/O Bit Definition

The following PIO lines require special attention:

Port A, Bit 0 (Software Reset) -- The CPU can reset the whole system by pulling this bit low.

Port A, Bit 7 (Z8001A Reset Protect) -- This bit works with the above bit. If this bit is set low, a software reset clears all I/O boards on the Multibus, but not the processor board.

Port A, Bit 2 (Enable Memory Map) -- When this bit is low, the Z8001A runs in mapped mode.

Port A, bit 3 (Enable Error Handler Trap) -- When this bit is inactive (high), the circuit which generates the segment trap for memory mapping errors or EDC errors is disabled.

Port A, Bit 1 (Multibus Lock) -- After powerup or reset, Multibus controllers are not allowed to access the memory. The Multibus access is permitted only after this bit is set (active LOW).

3.3.3. Counter/Timer Circuit

The CPU uses a four-channel counter-timer (Zilog Z80B CTC) to provide realtime interrupts to the Z8001A. The ZC/TO0 output (channel 0) is connected to the CLK/TRG input of channel 1. The ZC/TO1 output (channel 1) divides the system clock down to 50 hz. It is used to trigger a flip-flop (U123) which asserts the CTCINTR (active HIGH) to cause the interrupt.

The Z8001A resets the interrupts by performing a special I/O write to port 7FB1.

Each Z80-CTC channel must be programmed by the Z8001A prior to operation. It writes two 8-bit words to the desired channel over the local peripheral bus. The first word is the channel control word which sets the operating mode for the channel. The second word is the time constant (see the Zilog 1981 Data Book, "Z8430 CTC Product Specification"). The channel control word for channel 0 consists of the following:

D0	1	Identifies the control word
D1	1	Software reset enabled
D2	1	Time constant follows
D3	0	Timer is triggered automatically when time constant is loaded
D4	0	Falling edge triggered
D5	0	Prescaler value = 16
D6	0	Timer mode
D7	0	Interrupt disabled

Channel 1 is programmed the same as channel 0 except that it operates in counter mode (D6 is set to 1).

3.4. Realtime Clock

A clock calendar gives the CPU access to the time of day in hours, minutes and seconds, and also to the day, month and year.

This battery-operated circuit has its own crystal oscillator and remains running even when AC power is removed. The battery is automatically charged whenever the system is operating. It powers the clock for three months between charging.

The clock calendar function is provided by a MSM5832 CMOS clock circuit. This circuit inputs its data to the PIO, which passes it to the Z8001A over the local peripheral bus. Its address, data and control lines are listed in the Processor section, part 3.2.

Figures 6 and 7 show the PIO/realtime clock circuit read/write timing requirements (see _OKI Semiconductor MSM5832 Microprocessor Realtime Clock/Calender Product Specification_).

3.5. Programmable Interrupt Controller (PIC)

The vectored interrupt structure in the P/25 is handled by two Intel 8259A PICs. One functions as the master PIC, the other the slave PIC. The master PIC (page 14, U106) accepts interrupt requests from onboard peripherals. The slave PIC (page 14, U107) accepts interrupt requests from Multibus I/O controllers.

The basic functions of the PICs are to: a) accept interrupt requests, b) determine the priority of interrupt requests, c) assert a vectored interrupt to the Z8001A based on priority and d) provide the interrupt vector to the interrupt service routine.

In the P/25, there are 14 levels of interrupt priority, assigned as follows:

MASTER PIC

	Priority	Function
High	0	Unused
	1	Multibus watchdog timer
	2	CTC clock
	3	Multibus MMU error
	4	Unused*
	5	Trace, normal address space
	6	USART interrupt
Low	7	Cascade from slave PIC

* In older (4-Mhz) systems, priority 4 is assigned to the floating point processor.

SLAVE PIC

	Priority	Function
High	0	Multibus interrupt (INT0/)
	1	" (INT1/)
	2	" (INT2/)
	3	" (INT3/)
	4	" (INT4/)
	5	" (INT5/)
	6	" (INT6/)
Low	7	Multibus interrupt (INT7/)

*S3 "Programming Information" The Intel 8259A PIC requires initialization programming before it can process interrupt requests. The initialization sequence involves two to four byte-writes to each PIC. In the P/25, the initialization command words (ICWs) are issued as shown in Figure 8 (see the Intel 8259A Programmable Interrupt Controller Product Specification).

Since the PIC is not programmed in the automatic-end-of-interrupt (AEOI) mode, each individual interrupt service routine is required to issue a nonspecific end-of-interrupt command to clear the in-service bit in the corresponding PIC.

3.5.1. Circuit Analysis

The vectored interrupt cycle proceeds as follows:

A local peripheral device or a Multibus master requests an interrupt by asserting an interrupt line on one of the PICs.

The PIC determines the priority of the interrupt (see interrupt priority table above).

The PIC asserts the Z8001A VI/ line.

The Z8001A waits until it reaches an interruptable point, then starts an interrupt cycle by sending the proper status to the status latch decoder (block 2), which sends VIACK/ to the interrupt acknowledge handler (block 42).

The interrupt acknowledge handler outputs the first INTA/ pulse, which lasts for one clock cycle. This pulse causes the PIC to freeze the state of priority resolutions and causes the master PIC to issue an interrupt code to the slave.

The second INTA/ pulse. which is actually a copy of the DS/ signal from the Z8001A, causes the PIC to output the interrupt vector on the local peripheral bus. This is the end of the PIC's involvement.

The Z8001A reads the vector on AD0 to AD8. It goes to the vector indicated and executes the instructions there.

The high five bits of the 8-bit vector are common to all eight interrupt request lines on a given PIC. They define a memory location where an interrupt instruction table is located. The lower three bits carry the code (0 to 7) for the interrupt. These bits define the table entry specific to each interrupt.

Since the Z8001A requires all vectors to be on even boundaries (bit 0 must be zero), the PIC lines are shifted one bit, so that data line 0 from the PIC connects to data line 1 of the local peripheral bus; PIC line 7 connects to peripheral bus line 8. Data line 0 is only pulled LOW during an interrupt acknowledge cycle. This has the effect of placing the interrupt vector produced by the PIC on an even boundary. It also implies that when the PIC is read or written, the data must be shifted one bit to the left.

3.6. Diagnostic Port

The diagnostic console is supported by an industry-standard RS232C port (USART) attached to connector P3. The following table shows the EIA RS232C pins supported when a terminal is connected to P3. (If a modem is attached, the pins are identical to the ICP SIO pins -- see the Input/Output Controller section in this manual):

RS232C Pin	I/O	Name	2661 Pin	Function
2	I	TxD	RxD	Serial data from terminal
3	O	RxD	TxD	Transmitted data from diagnostic port
20*	I	DTR	DSR	General purpose input
6	O	DSR	DTR	General purpose output
8	I	DCD	DCD	General purpose input
5	O	CTS	RTS	General purpose output
4*	I	RTS	CTS	General purpose input
7	-	Ground	-	Signal ground
1	-	Ground	-	Protective ground

*These pins must be enabled (held high) for the UART to operate.

This connector is set up to connect directly to a terminal; data modems require a null modem cable.

The diagnostic (console) port consists of a Signetics 2661 enhanced programmable communication interface (EPCI) and RS232C transceivers. The EPCI is a universal synchronous/asynchronous data communication controller chip which can support various communication protocols. It contains a baud rate generator which can generate 16 different baud rates between 50 and 19,200 baud. The 2661 EPCI's TxRDY (Transmitter Ready), RxRDY (Receiver Ready) and TxEMP (Transmitter Empty) interrupt lines are all ORed together to create a single vectored interrupt to the Z8001A.

The diagnostic terminal baud rate is selected by three of the switches described in the Processor section, part 1.5.

3.6.1. Programming Information

The port addresses assigned to the registers in the 2661 EPCI are as follows:

Port Address	Function
7FA0H	Data holding register
7FA1H	Status register SYN1/SYN2/IDLE registers
7FA2H	Mode register 1
7FA2H	Mode register 2
7FA3H	Command register

4. MULTIBUS INTERFACE

4.1. Introduction

Most P/25 system components communicate with each other over the Multibus backplane, which occupies connector P1. The exceptions are the memory boards (dynamic RAM), which occupy connector P2. Refer to the IEEE Proposed Microcomputer System Bus Standard (P796 Bus) for a more complete description of the Multibus.

The processor board can act as a Multibus master or slave. As a master it can access up to 64K 16-bit ports in the standard I/O address space. As a slave, it occupies the upper half of the Multibus logical address space.

The Multibus interface can handle both byte and word transfers on the Multibus.

4.2. Multibus Memory Access

The timing diagram for the Multibus memory access is shown in Figure 9.

A Multibus memory request is recognized when both the most significant address line (ADR 13/) and the control line (MRDC/ or MWTC/) are activated by a Multibus master. Block 33 contains the logic to generate the MBREQ signal (see the processor schematics, page 12) asserts a request for local bus on the processor board. The local bus arbiter grants the request on a first-come-first-served basis by asserting MBGRANT/, which enables blocks 30 and 31. A memory cycle is started as discussed in this manual in the Memory section.

In a Multibus read cycle, data read from the memory is latched into U31 and U52 when the CYCEND/ is asserted. The processor board can start the next memory cycle or onboard I/O cycle without waiting for completion of the Multibus cycle. The XACK/ is asserted on the Multibus (sh 12, U136 pin 5). The Multibus master removes the MRDC/ or MWTC/ command. This ends a complete Multibus cycle.

4.3. Processor Board Multibus Access

The timing for the processor board Multibus access is shown in Figure 10.

The processor board accesses the Multibus controllers as I/O ports in the Z8001A standard I/O address space. The access starts when the IORQMB signal (page 9) goes active (HIGH). The IORQMB signal is generated at the rising edge of AS/ when a standard I/O instruction is executed. The synchronization of IORQMB and BREQ/ (Multibus signal) is achieved by flip-flop U173 (sh 10).

The Multibus exchange priority logic is located on the backplane. Priority is resolved in a parallel fashion whereby each slot on the backplane has a fixed Multibus access priority. This results in a location-dependent priority scheme.

If the Multibus is free (BUSY/ = high), a Multibus I/O cycle can be initiated immediately. Otherwise, the processor board waits until the current Multibus master relinquishes control of the bus (BUSY/ goes high). When this happens, J-K flip-flop U171 (sh 10) pulls the BUSY/ line low, locking out other Multibus masters. The MBIOENAB signal notifies the CPU that the Multibus has been granted. It also enables the data and control buffers (blocks 34 and 36) connected to the Multibus. A Multibus timer starts and the buffers output the address of the Multibus slave to be accessed.

When the slave responds by pulling the XACK/ line low, it sets a flip-flop (U176 -- processor schematics, page 10) which pulls the WAIT/ line high. The Z8001A leaves the wait-state and finishes the cycle.

If the slave fails to respond within 20 microseconds, the timer causes a Multibus timeout interrupt to be generated. The Z8001A leaves the wait-state and resets the interrupt by performing a write to I/O port 7FB3H. This ensures that the P/25 cannot be locked due to a failed Multibus card.

The trailing edge of DS/ from the Z8001A triggers the end of a Multibus I/O cycle. IORQMB, BREQ/, MBIOENAB/, IORD or IOWR/, and address or data go inactive sequentially. The BUSY/ line goes high, freeing the Multibus.

5. LOCAL BUS ARBITER AND TIMING SEQUENCER

This section describes the processor board timing logic for the local bus arbiter and the timing sequencer.

In the P/25 processor board, the local address bus and the local memory/data bus are shared by the Z8001A, Multibus controllers and the refresh circuits. The local bus arbiter is designed to handle the sharing of the buses. It arbitrates the buses on a first-come-first-served basis for every memory access. If there is contention, the priority is assigned in the order: Z8001A, refresh and Multibus. The GRANT lines (ZGRANT/, REFGRANT/ and MBGRANT/) signal the winning device.

The timing sequencer is designed to generate the proper timing for the different types of memory cycles. For example, the PMREQ starts a memory cycle when a mapped physical address is valid and the LEIN controls the latching of input buffers in the EDC chip.

5.1. Local Bus Arbiter

As shown in the block diagram (Figure 1), the local bus arbiter receives: a) the MBREQ signal from the Multibus select logic (block 33), b) the REFRQ signal from the refresh circuits (block 26) and c) the AS/ signal from the Z8001A. The one which is granted memory access has its corresponding GRANT/ line asserted (MBGRANT/, ZGRANT/ and REFGRANT/). These control lines direct the flow of data in the processor board.

The MEMRQ/ and CYCEND/ signals are used to handle handshaking between the local bus arbiter and timing sequencer. The MEMRQ/ starts a memory cycle and CYCEND/ ends a memory cycle.

The local bus arbiter controls six different transitions from one type of memory access to another. These are:

Z8001A to Multibus

Z8001A to refresh

Multibus to Z8001A

Multibus to refresh

Refresh to Z8001A

Refresh to Multibus

The memory transition is synchronized by running the local bus arbiter with a 20-Mhz clock.

5.2. Timing Sequencer

The timing sequencer (block 22) is a bipolar-PROM-based state machine. Nine different types of timing programs are stored in the PROM. Only one program is executed for each type of memory access. Outputs from the timing sequencer control the timings required in a read/write cycle.

The following signals are output from the timing sequencer:

PMREQ	-- Asserted when a valid mapped physical address is available.
REFSH	-- Generated in a refresh cycle. Functionally equivalent to PMREQ.
PREAD	-- Controls the direction of the data buffers on connector P2 for read operations.
PWRITE	-- Controls the direction of data buffers on P2 for write operations.
LEIN	-- Latches data into the EDC buffers.
LEOUT	-- Latches data into the EDC output buffers and samples the error/ and multi-error/ lines.
WE/	-- WRITE ENABLE strobe generated in a write cycle.
OE0,1/	-- Enables output latches in the EDC chip.
OESC	-- Enables output buffers of EDC chip.
GEN/	-- Active when CPU is in a memory write cycle.
DZEN	-- Controls the Z8001A data buffers.
CYCEND/	-- Signals the end of a memory cycle.

6. ERROR HANDLER

The error handler (block 43) has two functions. It generates a segment trap or a vectored interrupt when errors occur and it latches relevant information about the error in the error status registers.

The P/25 detects two kinds of errors on the processor board. The first kind, a memory mapping error, occurs when a memory access violates the protection bits assigned to a page. The second kind, an EDC error, is generated by the EDC chip when the data checkbits stored in memory do not match the new checkbits generated for comparison during a read operation.

The operating environment in which a memory mapping error is generated and the action associated with the error are discussed in Processor section 2 and in Memory section 2.

6.1. Circuit Analysis

6.1.1. Memory Mapping Errors

When a memory access violates the attribute bits assigned to a page, the ERROR line is pulled low, setting flip-flop U156. If the memory access was initiated by a Multibus controller, U170 is set and MMUINT is sent to the PIC which generates a vectored interrupt. If the access was generated by the CPU, a segment trap is generated. The CPU can mask the trap by setting bit 7 of PIO port A.

If the error is caused by a write to a read-only protected page, then the write operation is suspended by asserting SUPPRESS WRITE (see the processor schematics, page 15).

6.1.2. EDC Errors

When the EDC detects a single-bit error, it corrects it and activates the ERROR line to the error handler. When it detects a multiple-bit error, it activates the ERROR line and the MUL-ERROR line.

The error handler uses these inputs to generate the proper action: a) if the access was initiated by a Multibus master (other than the CPU) it sends a MMUINT to the PIC, which generates a vectored interrupt, and b) if the access was initiated by the CPU, the error handler sends SEGTRAP to the Z8001A, which generates a segment trap.

Because the EDC corrects single-bit errors, the interrupt (or trap) serves to log the error. For multiple-bit errors, the interrupt (or trap) prevents incorrect data from being written to the requesting device.

Both error lines to the error handler are sampled at the falling edge of LEOUT signal.

When the P/25 runs in the unmapped mode, the read-only bit and page-invalid bit in the page mapper must be cleared to allow the error handler to sample the EDC ERROR/ and MULTIERROR/ outputs.

6.1.3. Error Recovery

Before the error handler can detect the next error, the flip-flops (U156 and U170) must be cleared with a write to special port FFA0H. This is also required to enable the error handler after powerup or reset.

6.2. Error Status Registers

Error status registers located at port addresses FF90H, FF98H, FFA0H and FFA8H latch relevant status information when an error occurs. The registers latch the address of the memory location that caused the error, several status bits that describe the access, and a 6-bit syndrome code if the error was generated by the EDC. The format and data are shown in Figure 11.

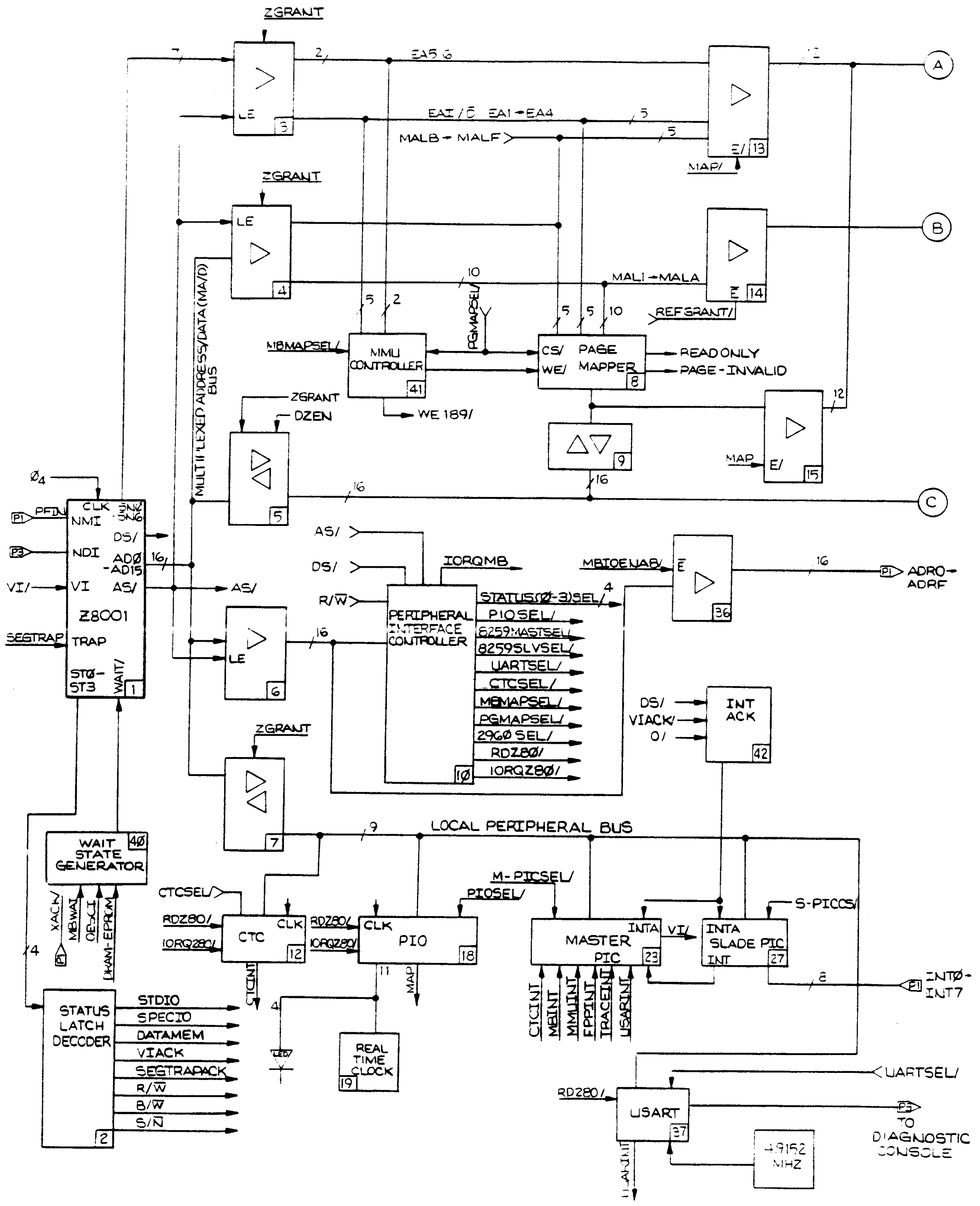


Figure 1. P/25 Processor Board Block Diagram -- Part A

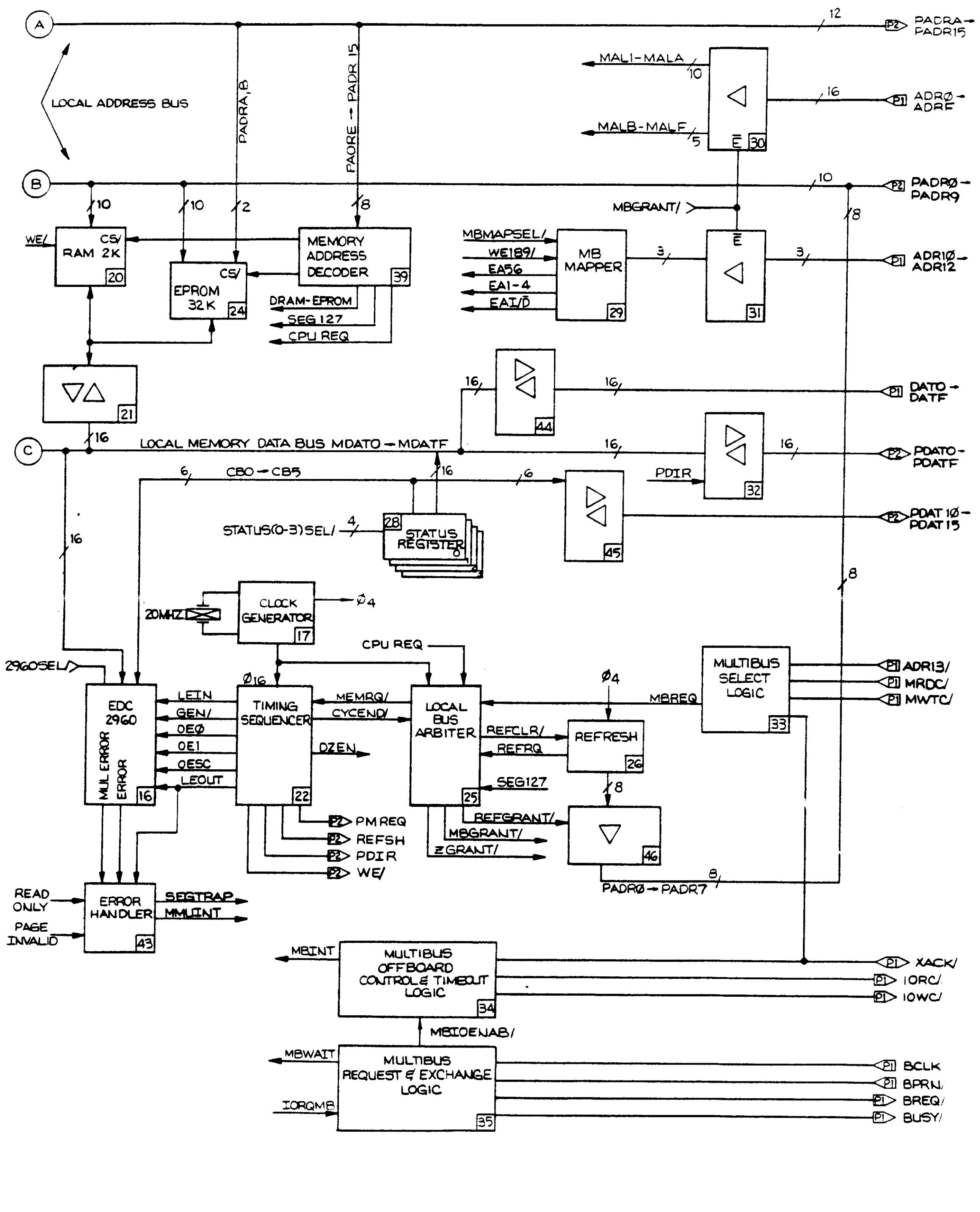


Figure 1. P/25 Processor Board Block Diagram -- Part B

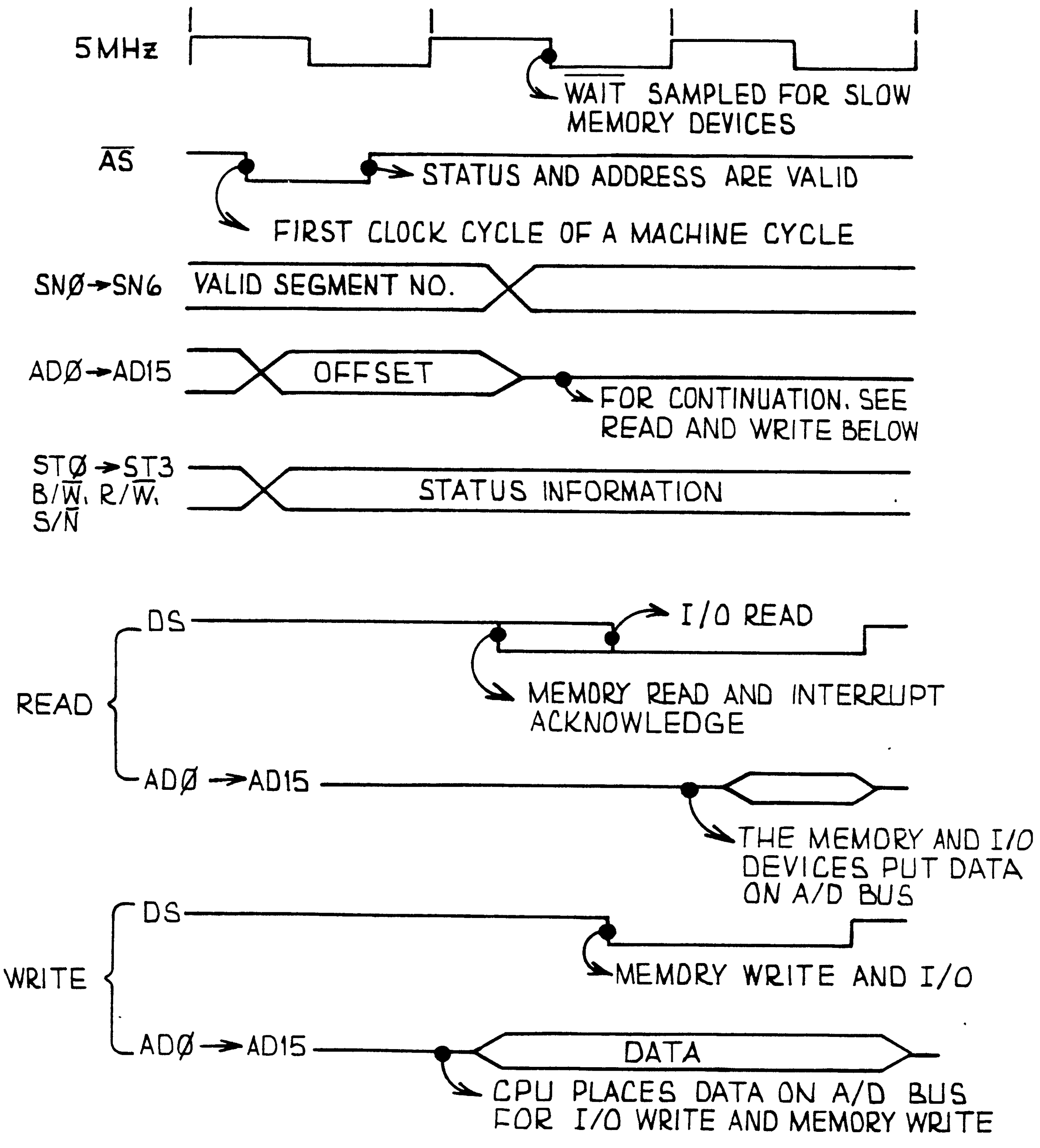

5MHz
WAIT SAMPLED FOR SLOW MEMORY DEVICES
AS
STATUS AND ADDRESS ARE VALID
FIRST CLOCK CYCLE OF A MACHINE CYCLE
SNØ→SN6
VALID SEGMENT NO.
ADØ→AD15
OFFSET
FOR CONTINUATION, SEE READ AND WRITE BELOW
STØ→ST3 B/W, R/W, S/N
STATUS INFORMATION
READ
DS
I/O READ
MEMORY READ AND INTERRUPT ACKNOWLEDGE
ADØ→AD15
THE MEMORY AND I/O DEVICES PUT DATA ON A/D BUS
WRITE
DS
MEMORY WRITE AND I/O
ADØ→AD15
DATA
CPU PLACES DATA ON A/D BUS FOR I/O WRITE AND MEMORY WRITE

Figure 2. Z8001A Read/Write Timing

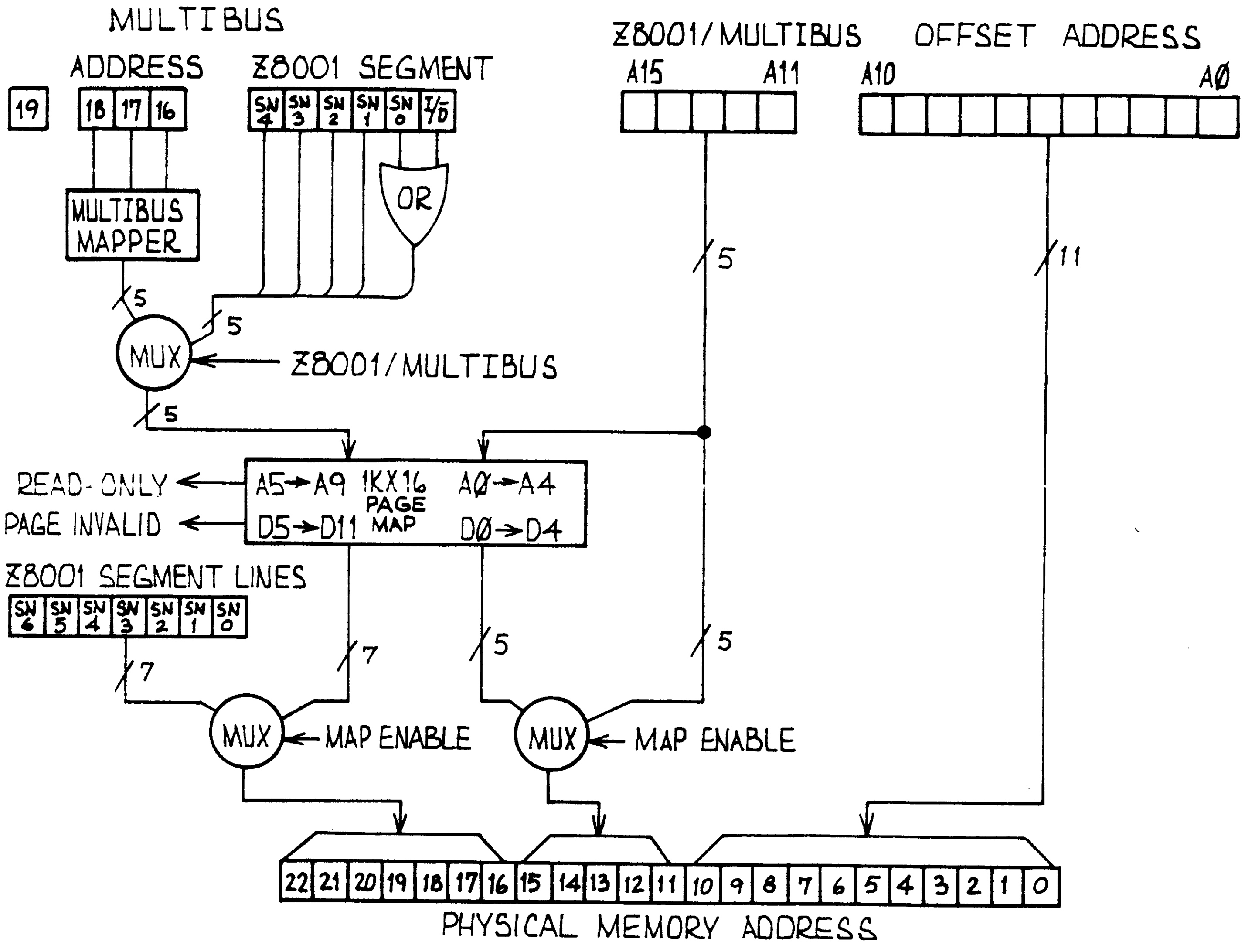


Figure 3. P/25 Address Mapping Scheme

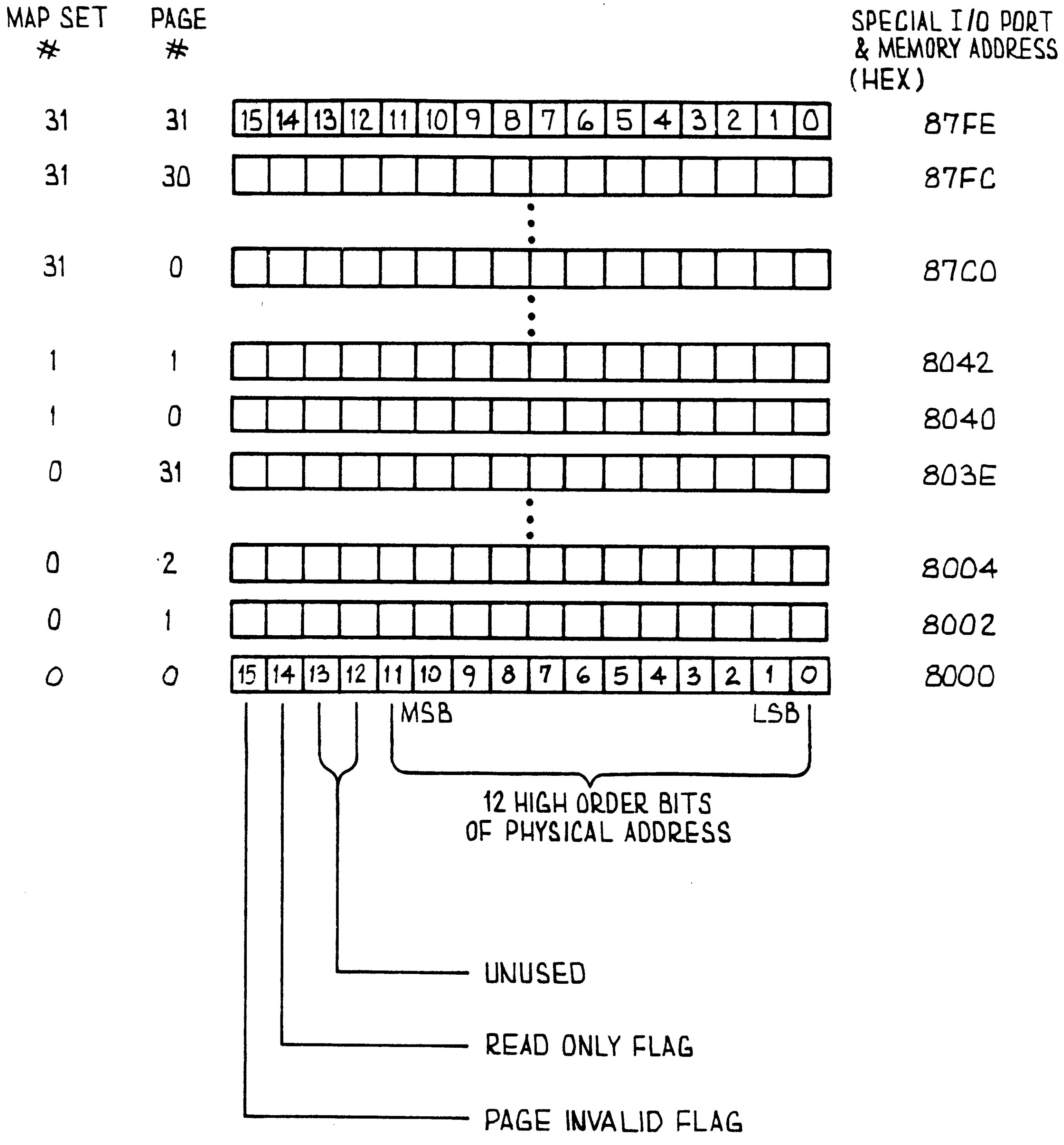


Figure 4. Page Mapping Layout

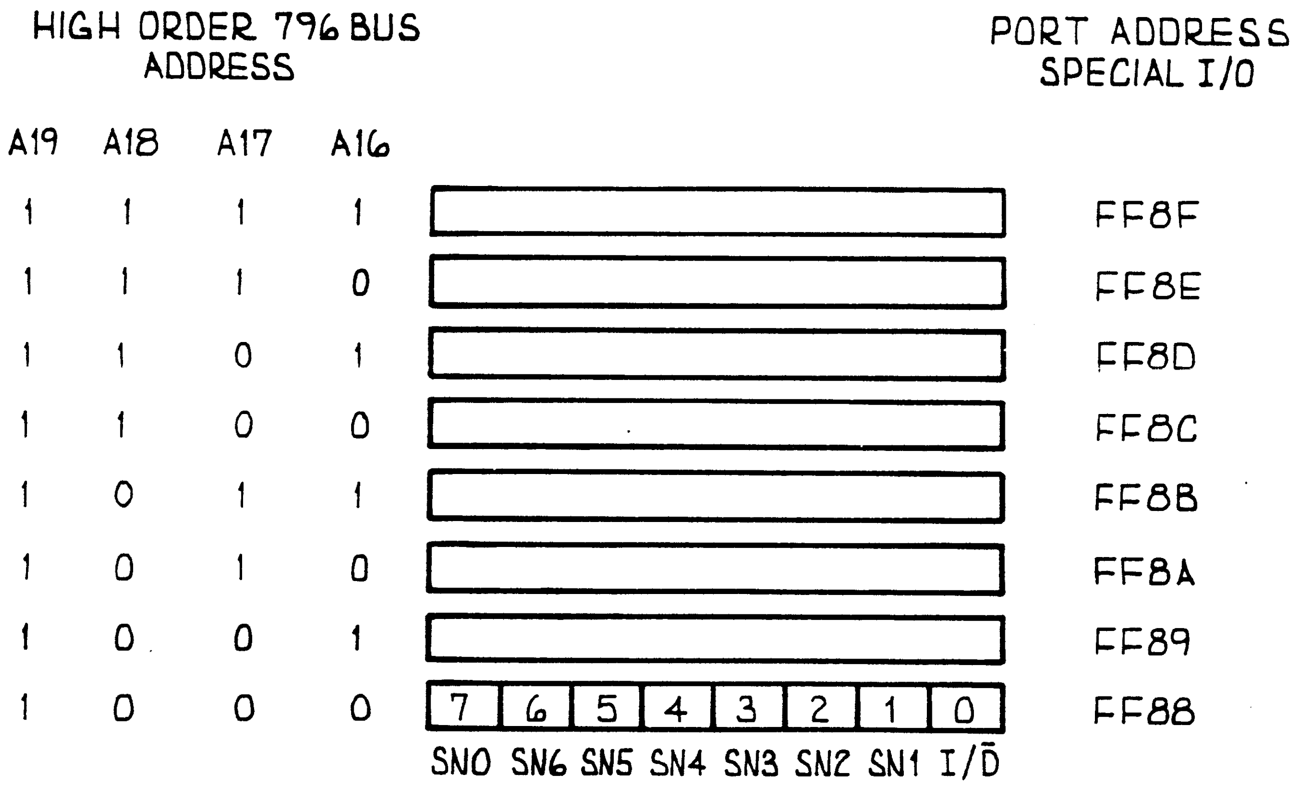


Figure 5. Multibus Map Layout

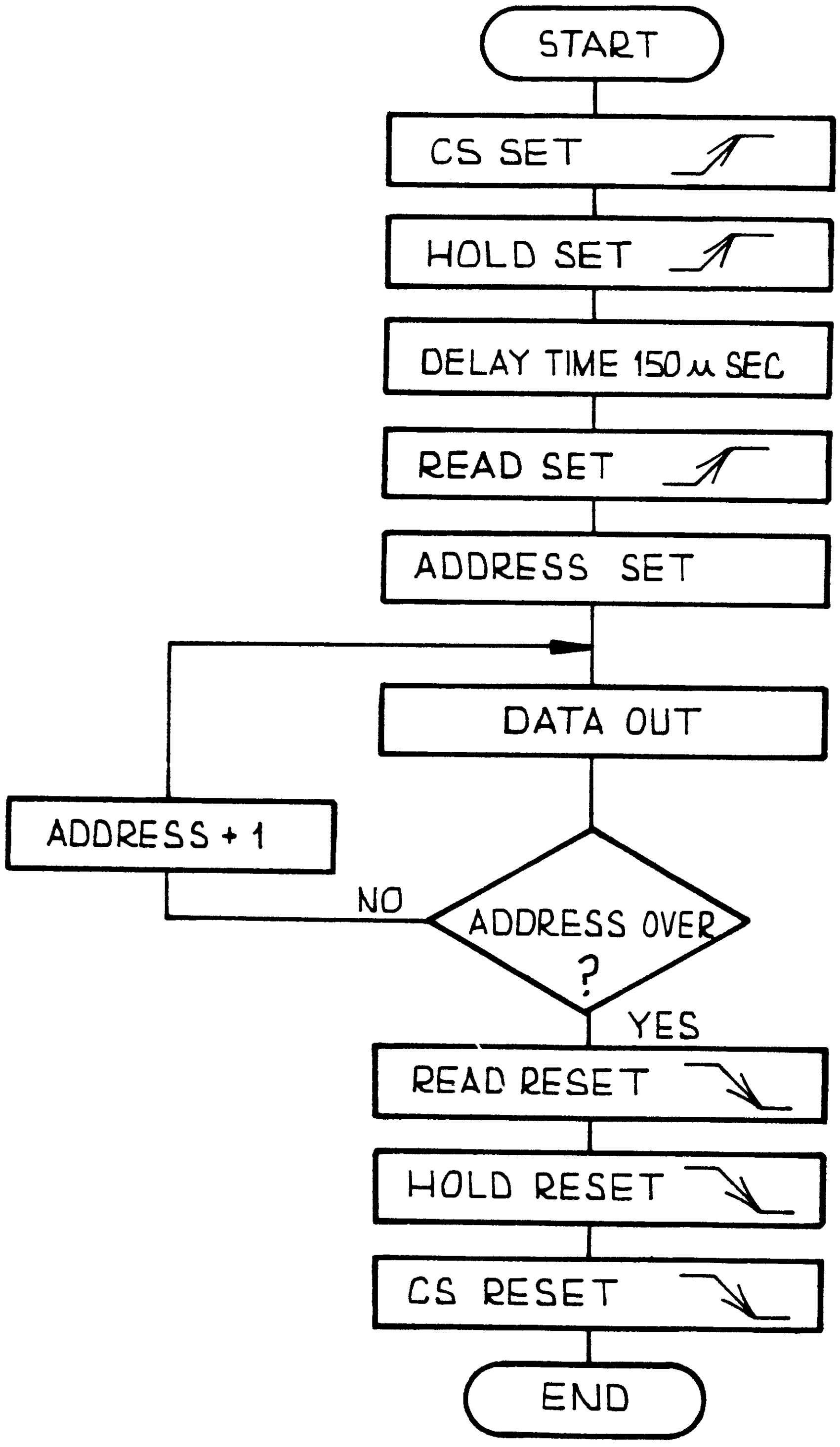


Figure 6. Read Cycle For Realtime Clock

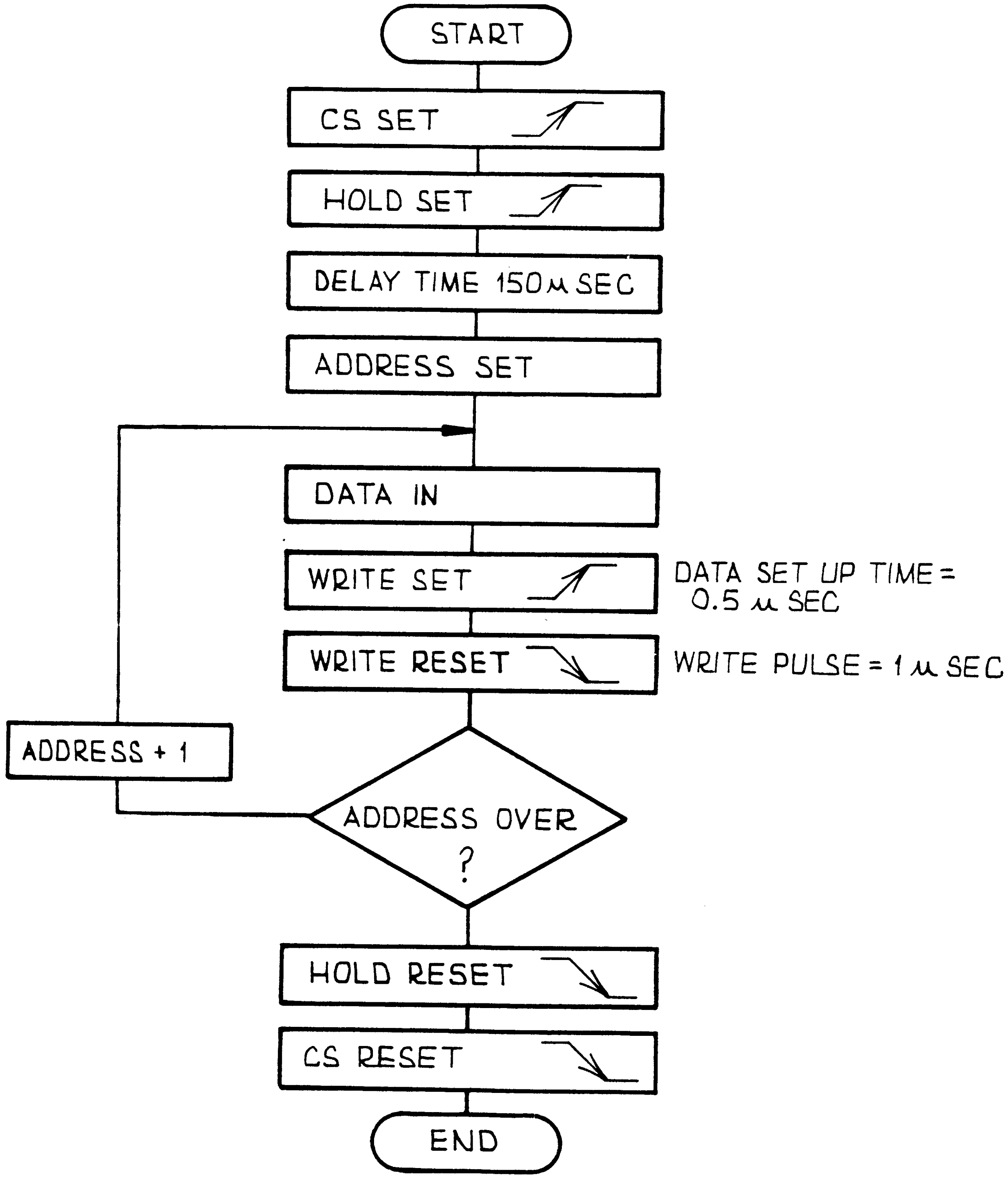


Figure 7. Write Cycle For Realtime Clock

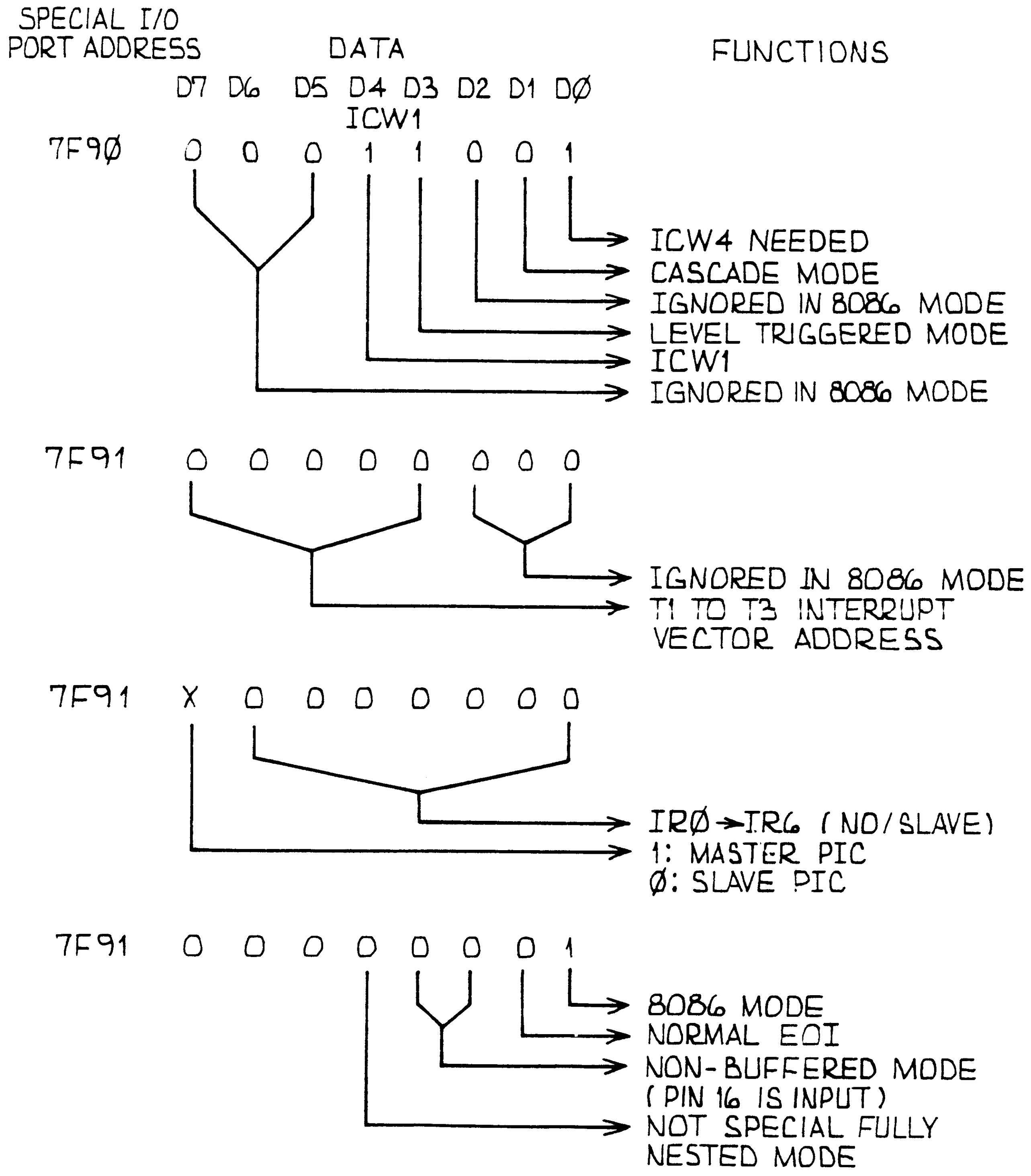


Figure 8. ICWs For 8259A (PIC)

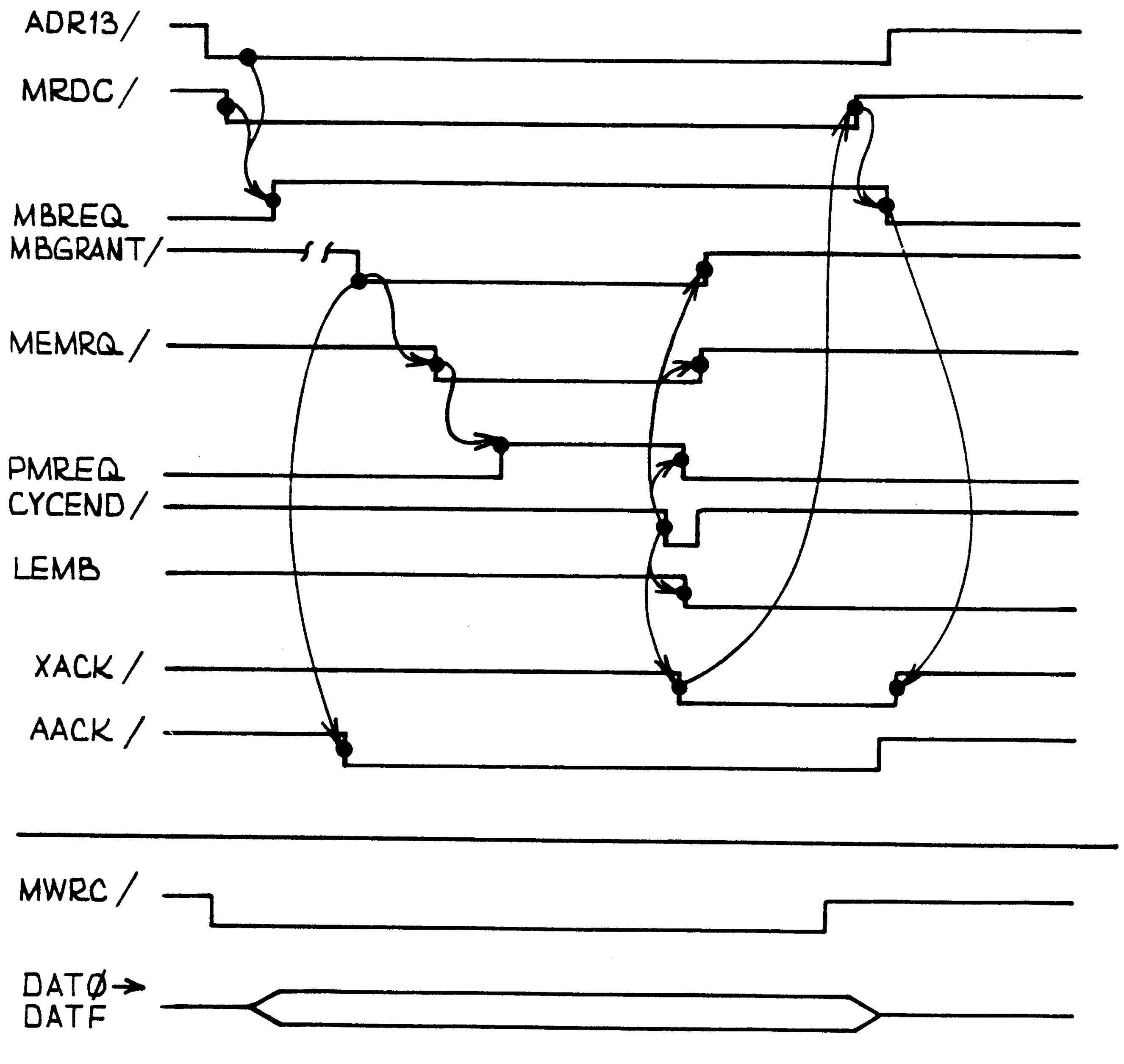


Figure 9. Multibus Memory Access Timing (Read/Write)

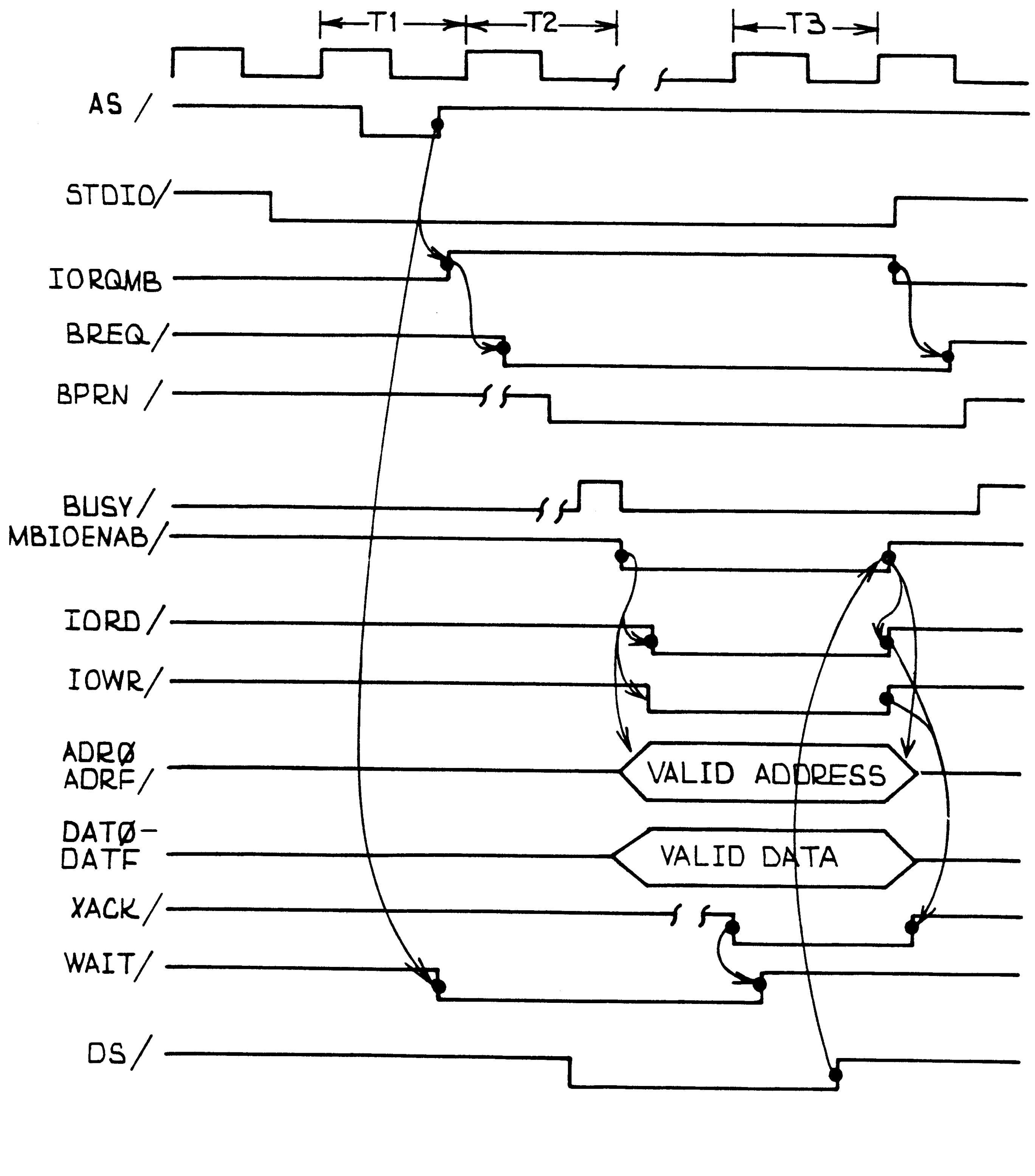


Figure 10. Processor Board Multibus Access Timing

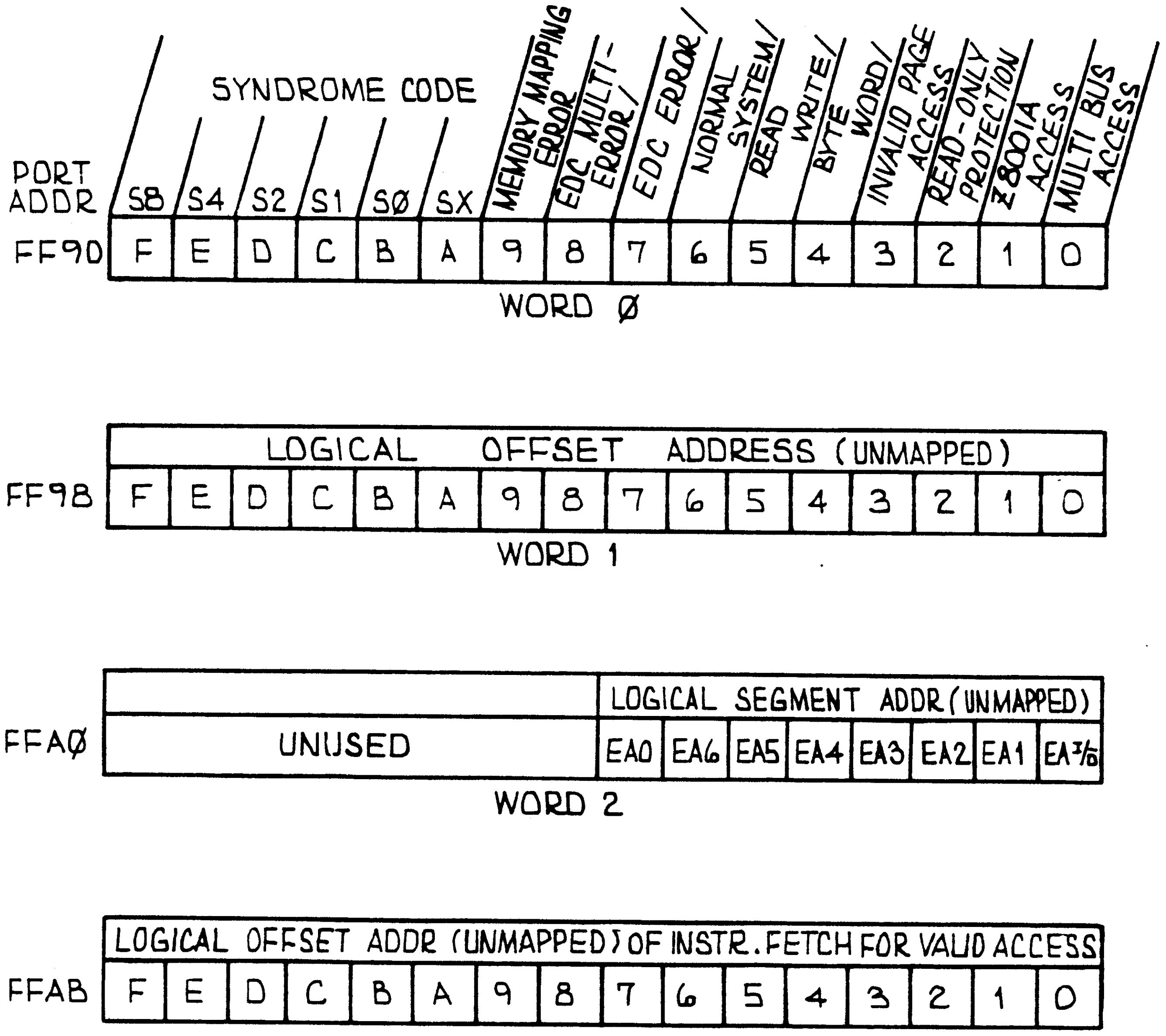
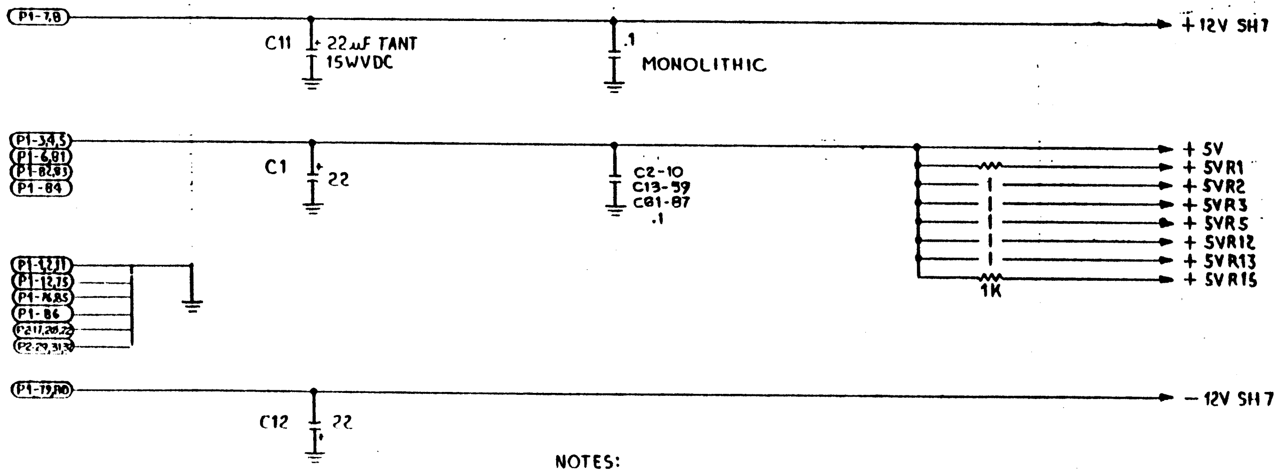

PORT ADDR
SYNDROME CODE
S8
S4
S2
S1
SØ
SX
MEMORY MAPPING ERROR
EDC MULTI-ERROR/
EDC ERROR/
NORMAL
SYSTEM/
READ
WRITE/
BYTE
WORD/
INVALID PAGE ACCESS
READ-ONLY PROTECTION
Z8001A ACCESS
MULTI BUS ACCESS
FF90
F E D C B A 9 8 7 6 5 4 3 2 1 0
WORD Ø
LOGICAL OFFSET ADDRESS (UNMAPPED)
FF98
F E D C B A 9 8 7 6 5 4 3 2 1 0
WORD 1
LOGICAL SEGMENT ADDR (UNMAPPED)
FFAØ
UNUSED
EA0 EA6 EA5 EA4 EA3 EA2 EA1 EA⁷/₀
WORD 2
LOGICAL OFFSET ADDR (UNMAPPED) OF INSTR. FETCH FOR VALID ACCESS
FFAB
F E D C B A 9 8 7 6 5 4 3 2 1 0

Figure 11. Error Status Registers

REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED
U1	S24Ø	3		U29	2148H-3	5		U57	LS245	3		U85	LS173	16	
U2	S241	,18,19	3	U3Ø	LS241	5		U58	LS245	3		U86	LS32	9,16	
U3	LS3Ø	4		U31	LS373	12		U59	LS374	16		U87	LS245	8	
U4	LSØ4	4		U32	LS241	12		U6Ø	LS374	16		U88	LS245	8	
U5	LSØ4	4		U33	LS374	16		U61	LS138	8		U89	LSØ8	3,8,9	1
U6	LS24Ø	5	2	U34	LS374	16		U62	LS138	8		U9Ø	LS32	8,14,17	
U7	S24Ø	5		U35	LS24Ø	8		U63	2732	4		U91	LSØ2	9,14	
U8	LS24Ø	5		U36	LS373	8		U64	2732	4		U92	LS74	14	
U9	S24Ø	5,1Ø	1	U37	LS373	8		U65	LS93	2Ø		U93	LS245	4	
U1Ø	LS24Ø	6		U38	LS24Ø	8		*U66	LS28Ø	13		U94	LS245	4	
U11	8287	12		U39	LS3Ø	9		*U67	LS28Ø	13		U95	AM2960PC	3	
U12	8287	12		U4Ø	LS26Ø	9		U68	LS374	16		U96	LS9Ø	2Ø	
U13	LSØ4	11	2	U41	SØ4	1Ø	3	U69	LSØØ	3	3	U97	LS92	2Ø	
U14	LS24Ø	2		U42	S374	16		U7Ø	SØ2	11,12		U98	LS86	12,13	1
U15	S373	2		U43	2114	4		U71	S189	11		U99	.S74	19	1
U16	S373	2		U44	2114	4		U72	S189	11		U1ØØ	LS27	12	
U17	S373	2		U45	2114	4		U73	S373	2		U1Ø1	S112	12,2Ø	
U18	LS24Ø	11		U46	2114	4		U74	Z8ØØ1-A	2		U1Ø2	SØ5	11,12	2
U19	LSØ4	2,8,14	1	U47	LS24Ø	2Ø		U75	S241	2,4,17	2	U1Ø3	S287	15	
U2Ø	LSØ4	14		U48	LS245	5		U76	2732	4		U1Ø4	LSØ4	9,16	
U21	S24Ø	3	4	U49	LS245	5		U77	2732	4		U1Ø5	LS138	9	
U22	S24Ø	3		U5Ø	LS241	2,5	3	U78	LS93	2Ø		U1Ø6	8259A	14	
U23	S24Ø	3		U51	LS241	12		U79	LS241	4,13,14,17	3	U1Ø7	8259A	14	
U24	S24Ø	3		U52	LS373	12		U8Ø	LSØ4	5,12,16	1	U1Ø8	SØØ	19	
U25	S24Ø	3		U53	8287	2		U81	LS32	11,15		U1Ø9	LSØ4	4	2
U26	2148H-3	5		U54	LS24Ø	11	2	U82	SØ8	12	2	U11Ø	LS1Ø	4	
U27	2148H-3	5		U55	LSØ4	2,11,14		U83	LSØ2	1Ø,11,15		U111	LSØ2	4,15,17	
U28	2148H-3	5		U56	LS3Ø	17		U84	LS173	16		U112	LS1Ø	4,12	

REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. ID	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO	GATES NOT USED
U113	LSØ4	4,9,19		U141	SØ2	3		U169	LSØ8	15,17	1	Y1	OSC4.9M	7	
U114	LS32	15,2Ø	1	U142	2661-1N	7		U17Ø	LS74	15		Y2	OSC32M	18	
U115	LSØ2	12,13,15		U143	9512-DC	7		U171	S1Ø9	1Ø	1	DL1	DDU-4	17	
U116	S112	15		U144	S114,	18	1	U172	SØØ	1Ø	1	DL2	DDU-4	19	
U117	S74	12		U145	LS21	18	1	U173	S74	1Ø					
U118	S112	12		U146	SØ4	3,19	1	U174	LS74	9					
U119	SØ4	1Ø,11,12	2	U147	S287	19		U175	LS74	9					
U12Ø	LSØ2	15		U148	S287	19		U176	LS74	1Ø					
U121	S32	2,12	1	U149	SØ4.	19,2Ø		U177	LSØ4	6	3				
U122	LS138	9		U15Ø	S112	2Ø		U178	MSN58	6					
U123	LS74	6,1Ø		U151	SØ8	19	3		32RS						
U124	Ø6	4,6,7,1Ø		U152	LSØ4	2,15,17		U179	LS241	6,7					
U125	LS1Ø	6,7,1Ø		U153	S26Ø	16,17		U18Ø	S241	17,18,19					
U126	LSØ4	6,7		U154	S74	17		U181	S112	18					
U127	LS32	6,7,17		U155	S32	15,17	1	U182	S112	18	1				
U128	LS74	7		U156	LS279	15,17	1	U183	S26Ø	17,18					
U129	LS1Ø	19	1	U157	S112	9,17		U184	SØ5	17,2Ø	3				
U13Ø	SØØ	19		U158	Z8ØACTC	6		U185	LSØ4	17,2Ø					
U131	S114	19		U159	Z8ØAPIO	6		U186	LS74	17					
U132	S114	19		U16Ø	S114	18		U187	S2Ø	17					
U133	S114	19		U161	LSØØ	18,2Ø	1	U188	S1Ø	17					
U134	S114	19		U162	LS112	2Ø		U189	S74	1Ø					
U135	S112	19	1	U163	S112	18		U19Ø	LS27	1Ø	1				
U136	S24Ø	12,17	4	U164	S112	18		U191	S74	9	1				
U137	LS74	15,17		U165	S11	1Ø,18		U192	1488	7	1				
U138	S32	12,17,1Ø		U166	S1Ø	9,1Ø		U193	1489	7					
U139	SØ8	12,16,17		U167	SØ3	17		U194	LSØØ	9,15	1				
U14Ø	SØ8	3,11,15		U168	SØØ	17	1	U195	LS123	6,1Ø					

* U66,U67 NOT REQ. FOR ECC OPTION.



[A] THE MEMORY PARITY CIRCUIT STILL EXISTS ON THE P.C. BOARD 60-00080 BUT IS NOT SHOWN IN THE SCHEMATIC, 60-00090.

NOTES:
1. ALL CAPACITORS ARE IN MICROFARADS ±20% UNLESS SPECIFIED OTHERWISE.
2. RESISTORS ARE IN OHMS ±5% 1/4 WATT.

60-00080

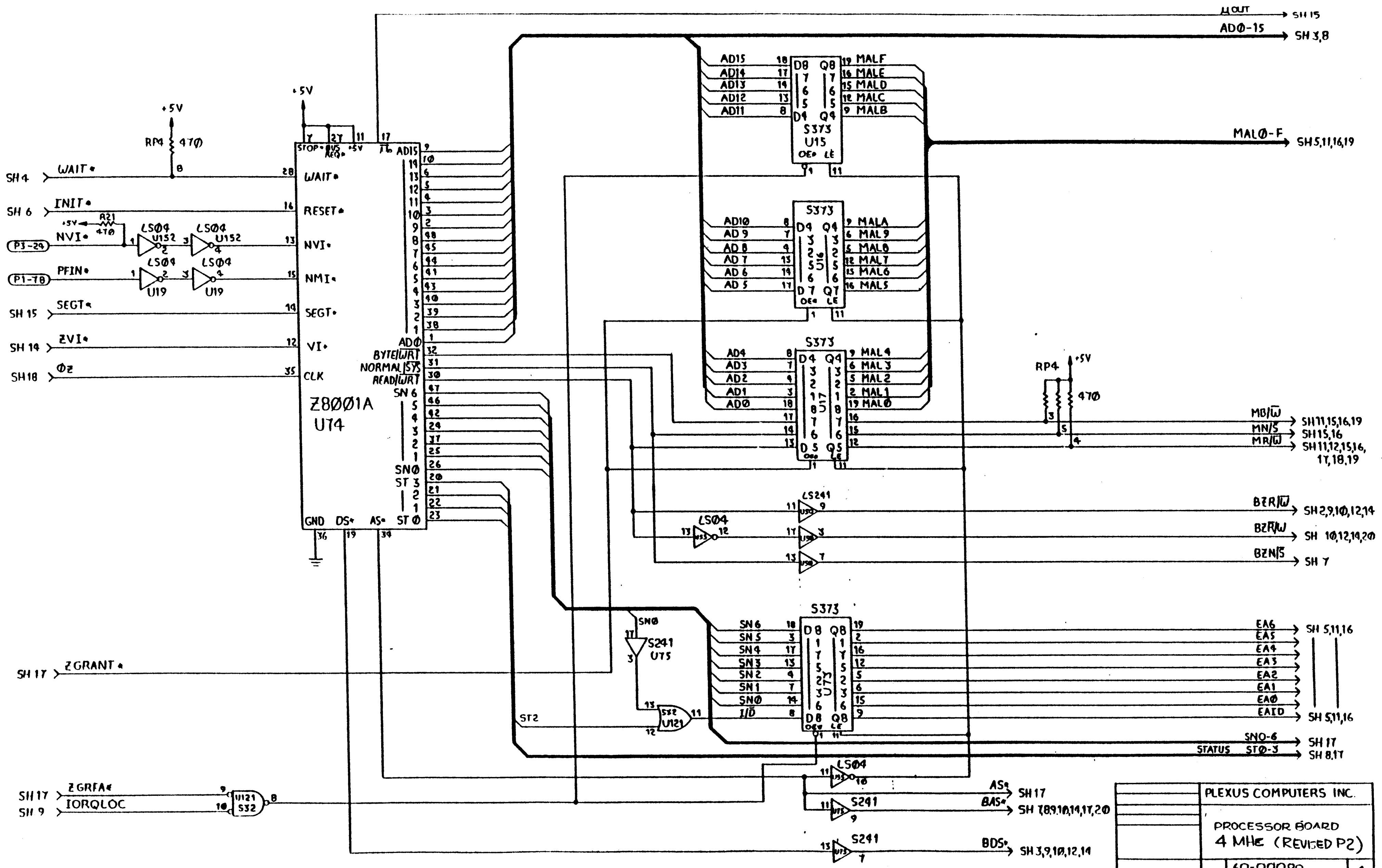
REV	DESCRIPTION	DATE	APPV'D
A	EN123 WAS: 60-00080 IS: 60-00090		
G	EN112 ADD R22, 23	4-2-82	WAM
F	EN106	4-2-82	WAM
F	EN103 CHG AT U 134, 150, 168	4-2-82	WAM
E	C102 REPLACED WAS: 20PF, IS 7PF EN 86	1-14-82	WAM
D	U185 REPLACED BY 74LS04, U77 BY 74S74	1-14-82	WAM
C	SWITCHED TO 32K EPROMS ENGR'G PROD. RELEASE EN 65	10-28-81	WAM

[illegible] 7-81

PLEXUS COMPUTERS INC

PROCESSOR BOARD
4 MHz (REVISED P2)

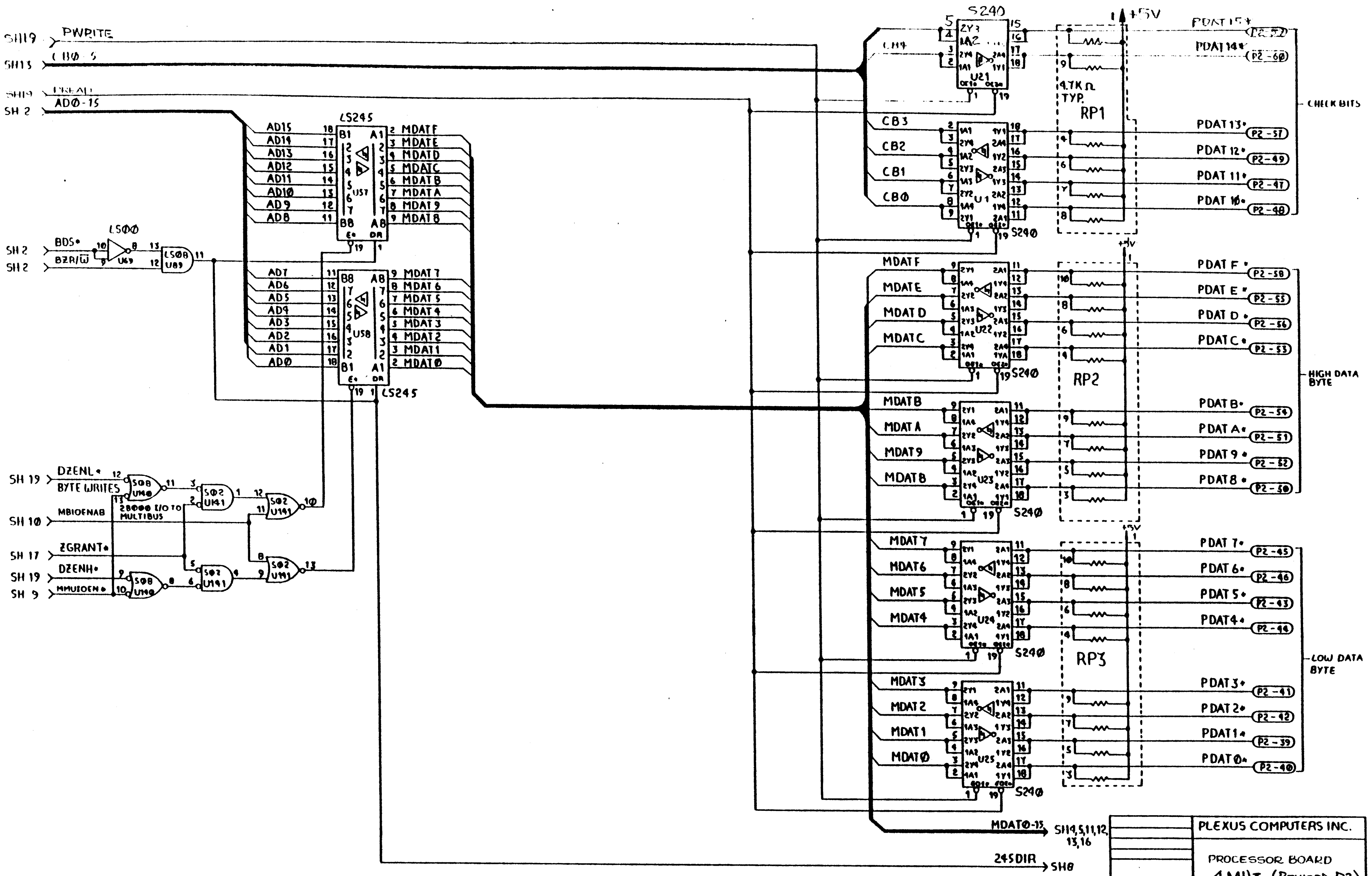
SIZE	SCALE -	SH 1 OF 20	REV
	60-00090		A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

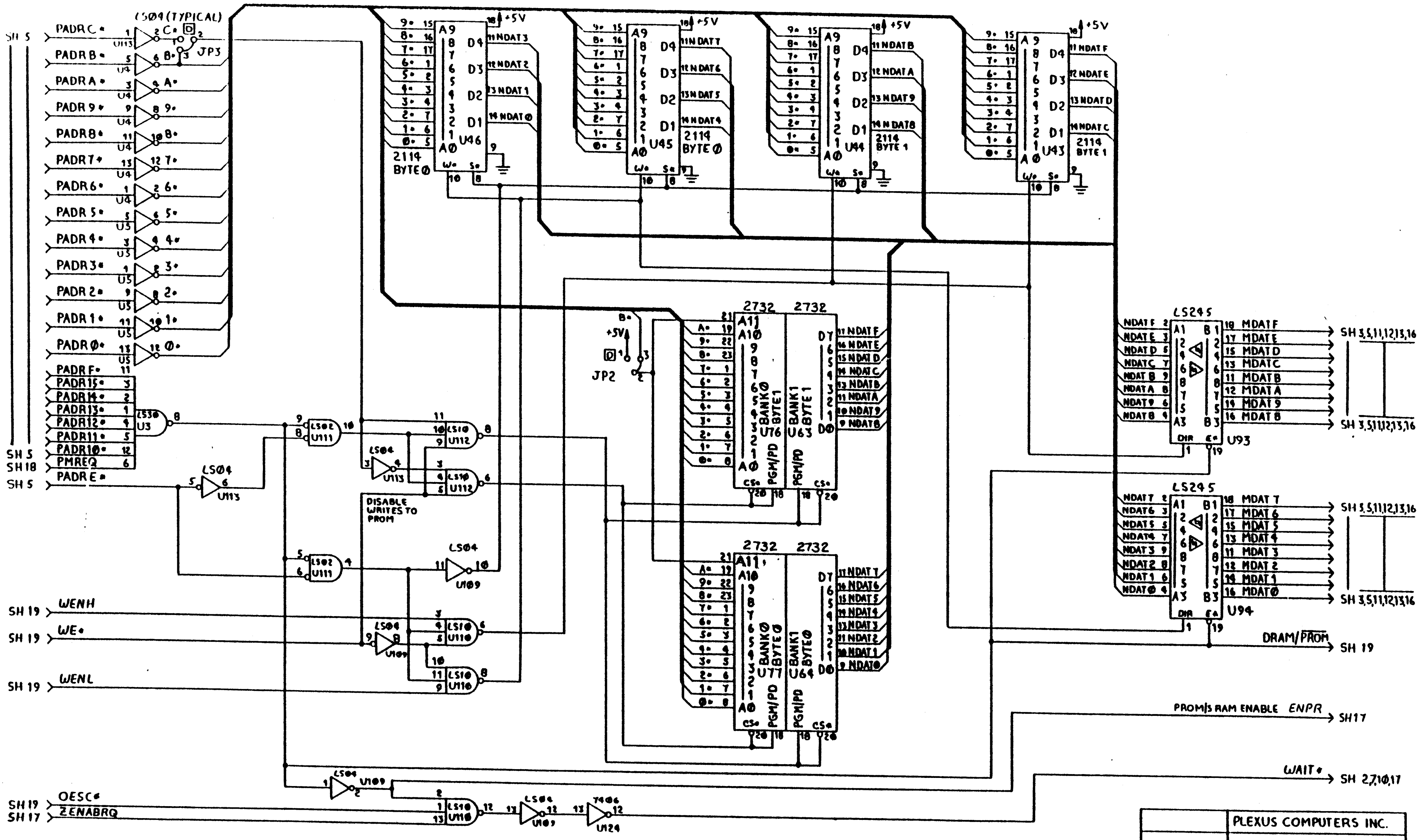
SIZE	SCALE —	SH 2 OF 20	REF
	60-00090		A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

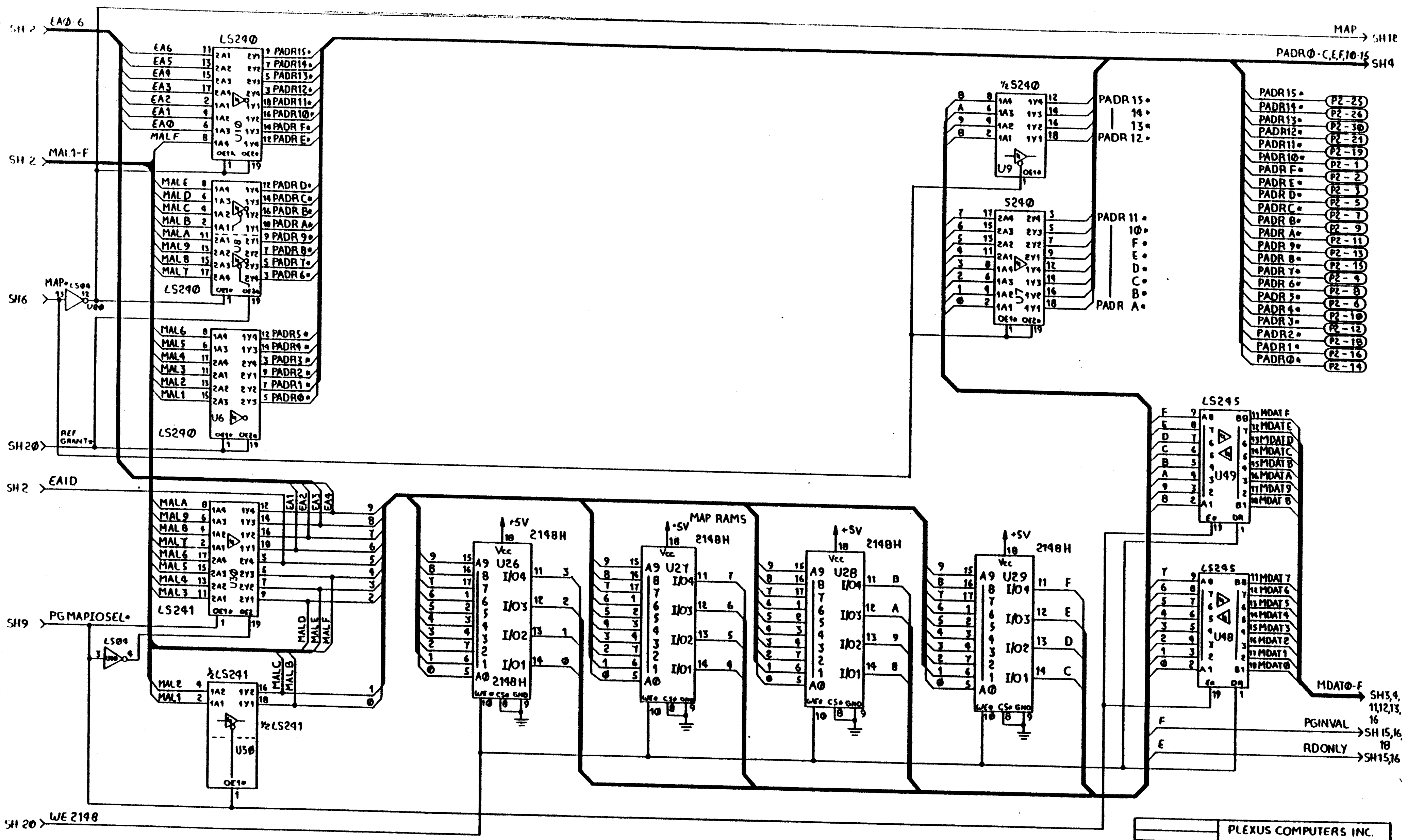
	60-00090		A
SIZE	SCALE —	SH 3 OF 28	REV



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHZ (REVISED P2)

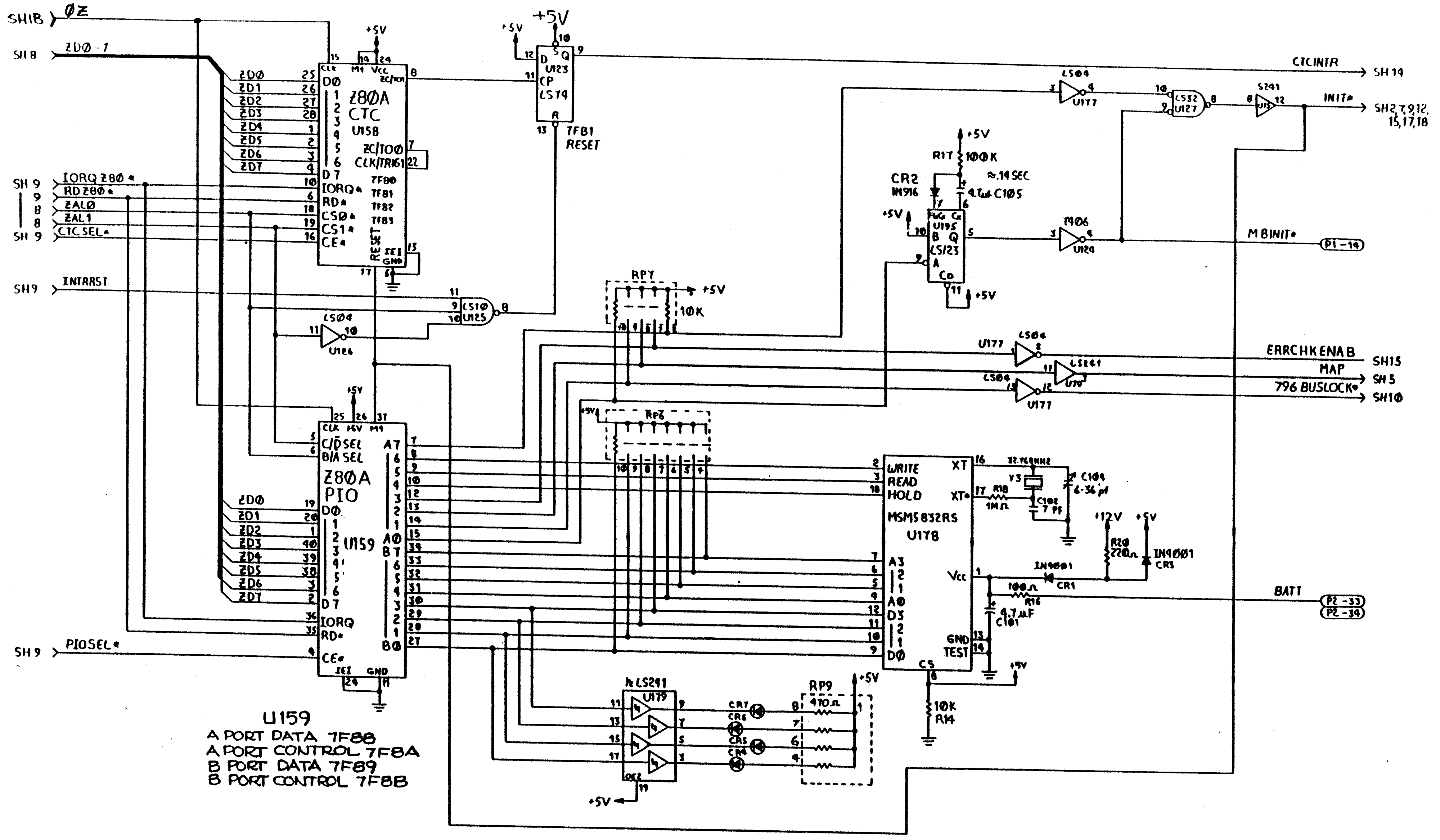
SIZE	SCALE —	60-00090 SH 4 of 26	REV A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz

SH 5 OF 20 | 60-00090 | A

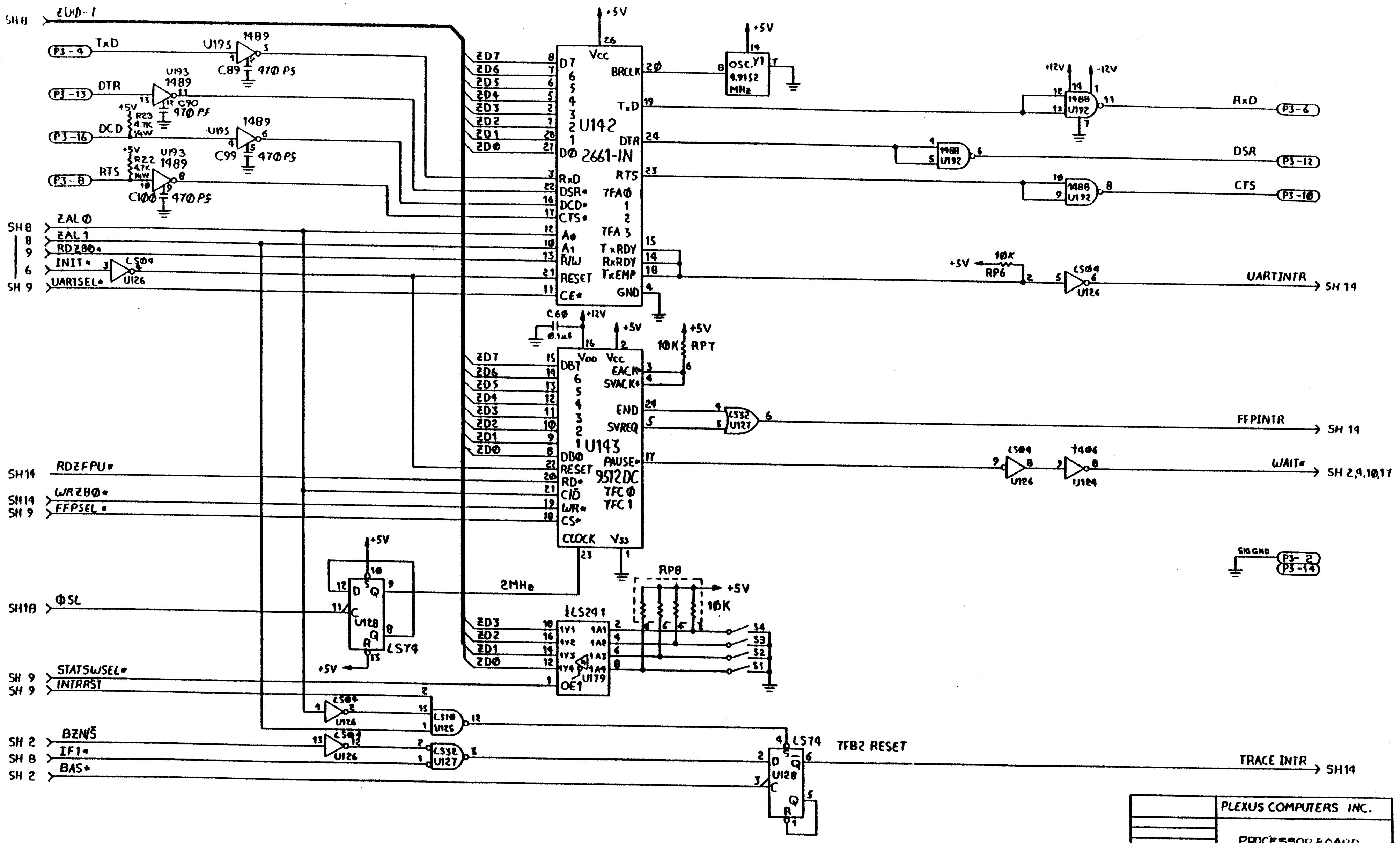


U159
A PORT DATA 7F88
A PORT CONTROL 7F8A
B PORT DATA 7F89
B PORT CONTROL 7F8B

PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

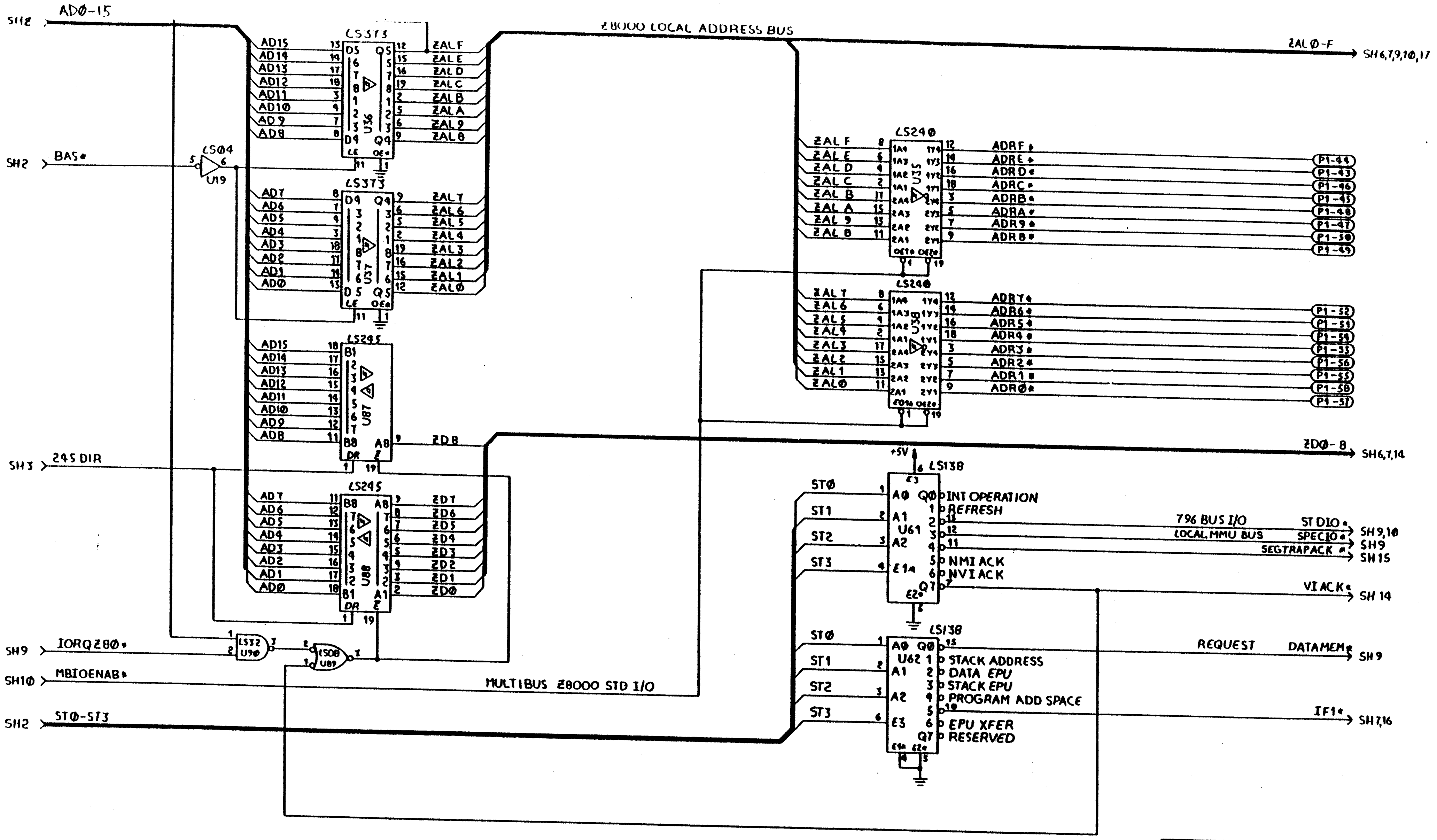
SIZE	SCALE		REV
	60-00090 —	SH 6 OF 20	A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

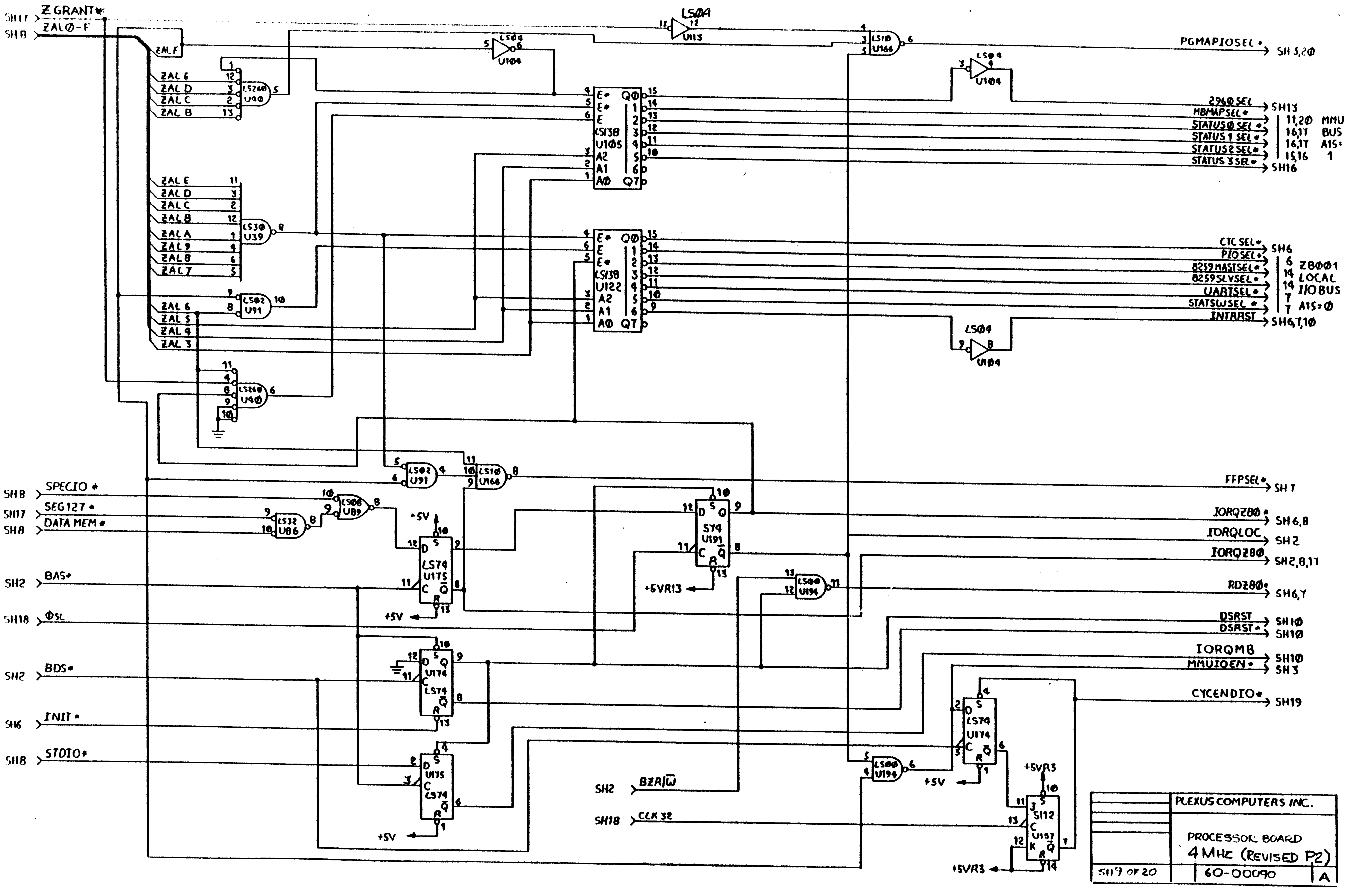
SIZE	SCALE —	SH 7 OF 20	REV
	60-00090		A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

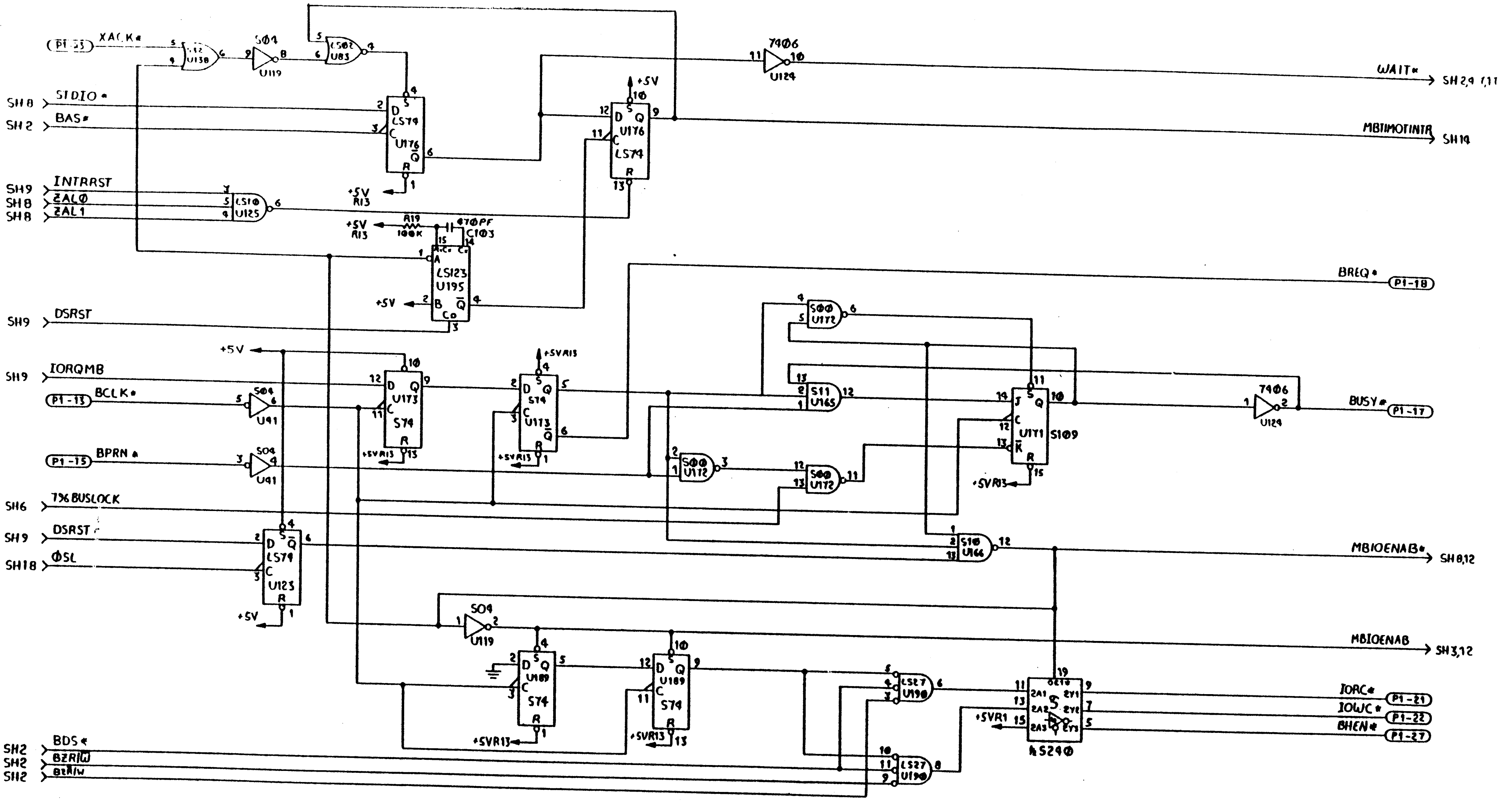
SH 8 OF 20 | 60-00090 | A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

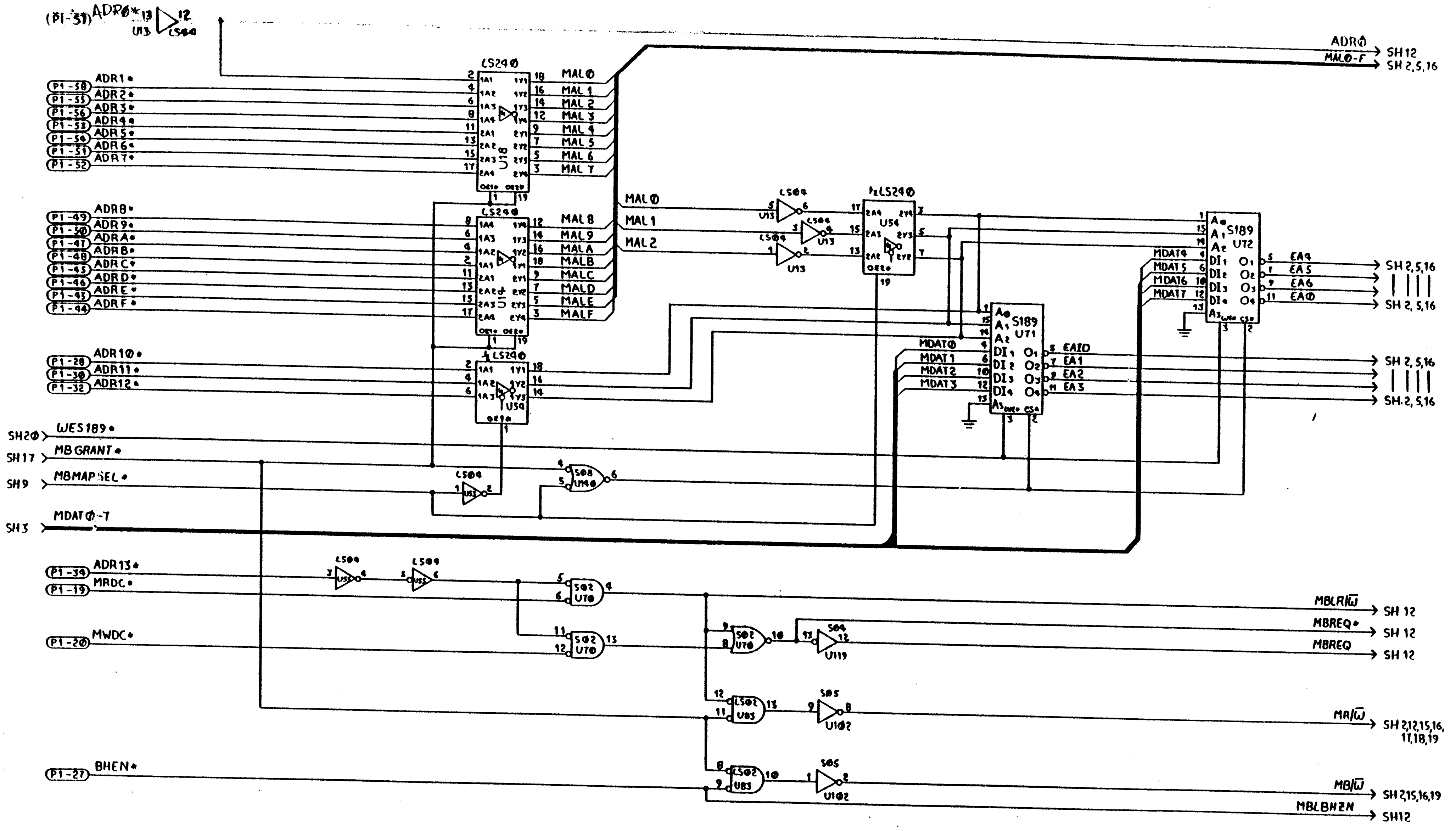
SH19 OF 20 | 60-00090 | A



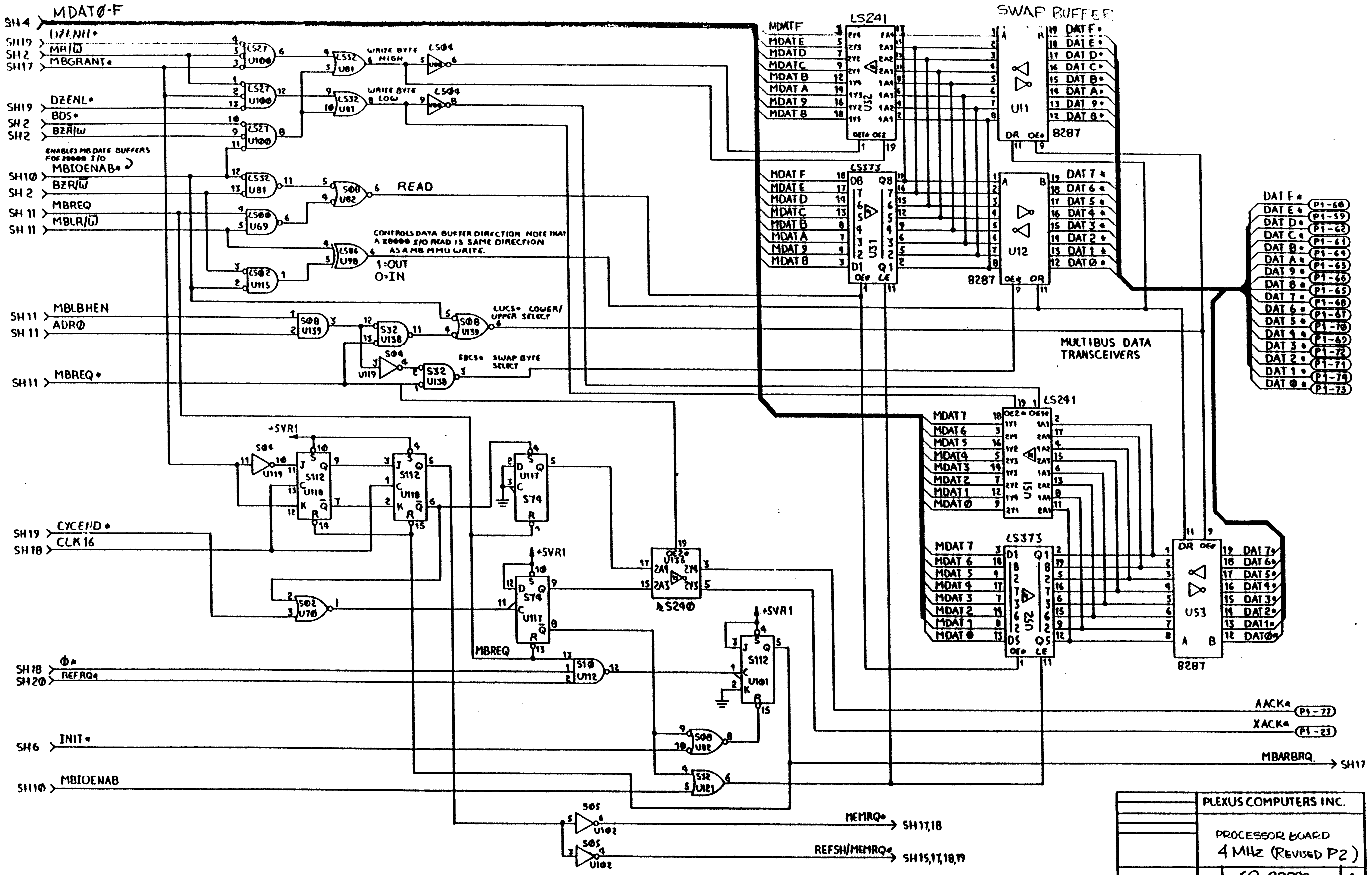
PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

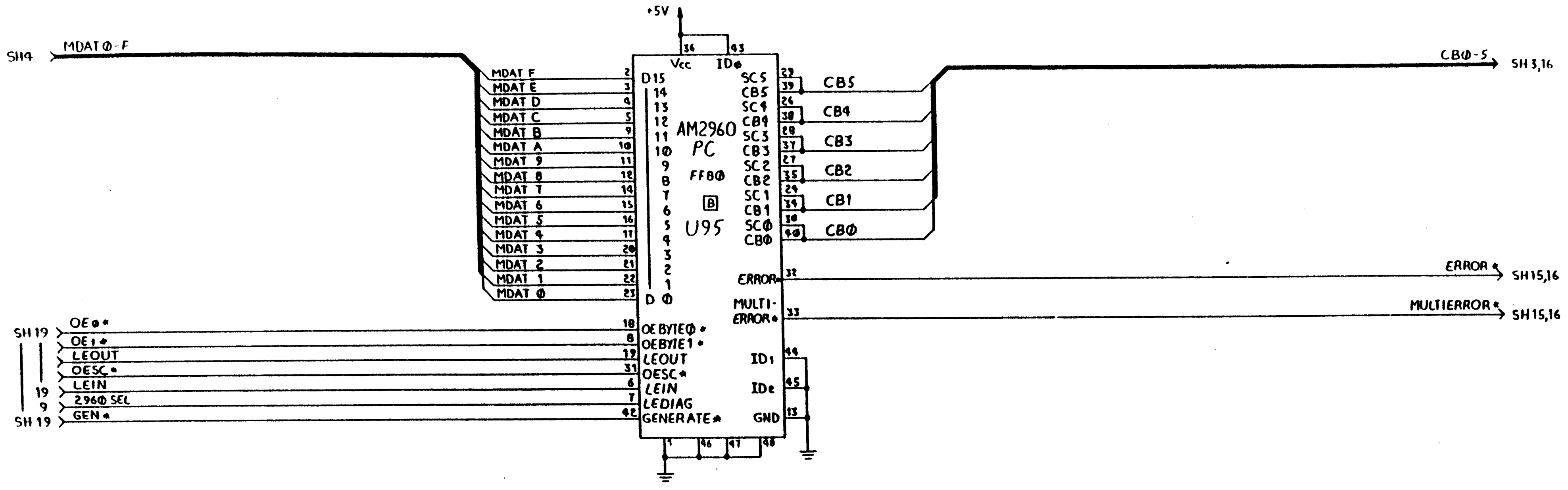
SIZE	SCALE —	60-00090	REV A
		SH 10 OF 20	



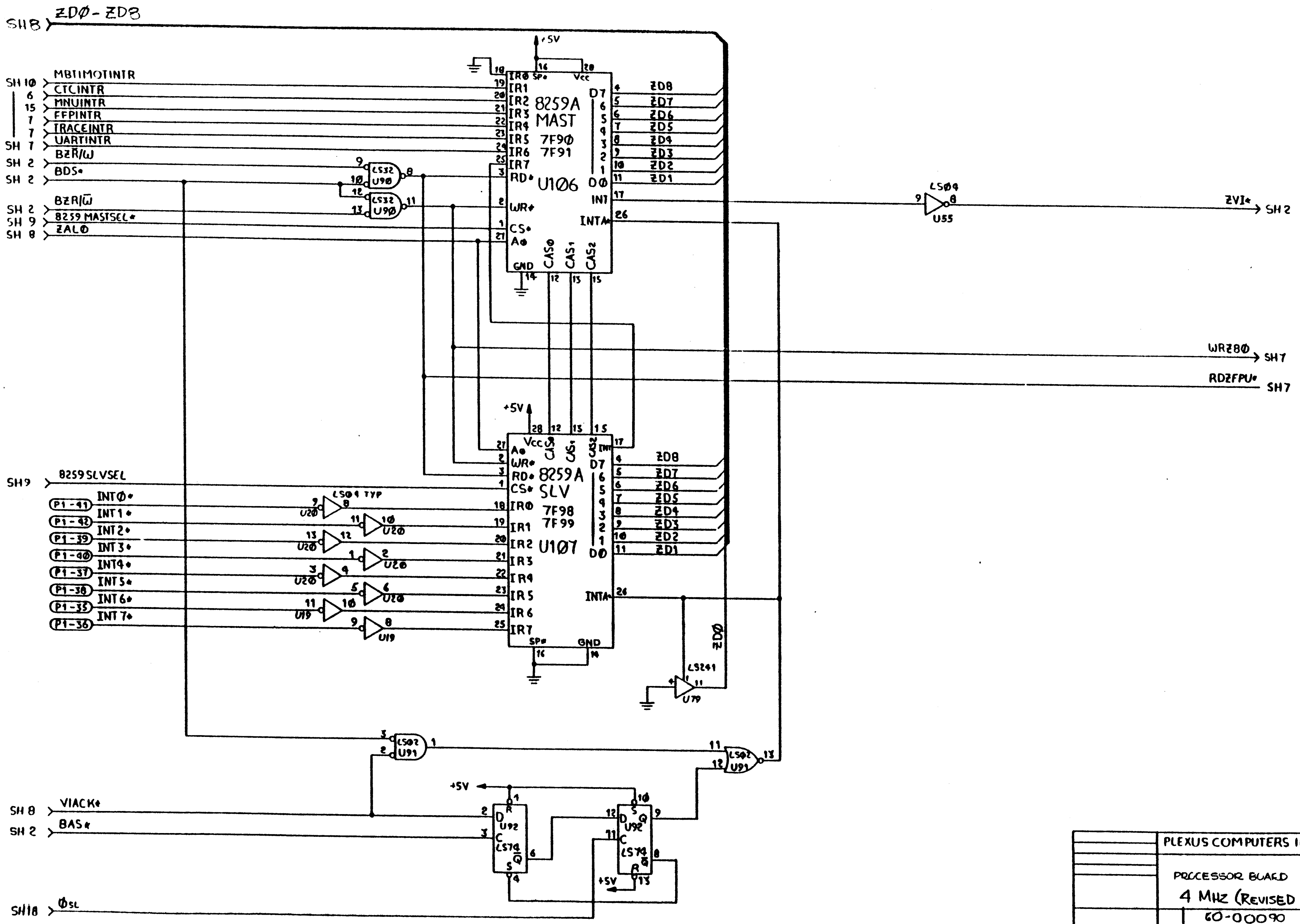
PLEXUS COMPUTERS INC
PROCESSOR BOARD 4MHz (REVISED P2)
60-00090 — A
SIZE SCALE — SH 11 OF 20 REV



PLEXUS COMPUTERS INC.			
PROCESSOR BOARD 4 MHz (REVISED P2)			
	60-00090		A
SIZE	SCALE —	SH12 OF 20	REV



	PLEXUS COMPUTERS INC		
	PROCESSOR BOARD 4 MHz (REVISED P2)		
SIZE	60-00090		A
	SCALE —	SH 15 OF 20	REV

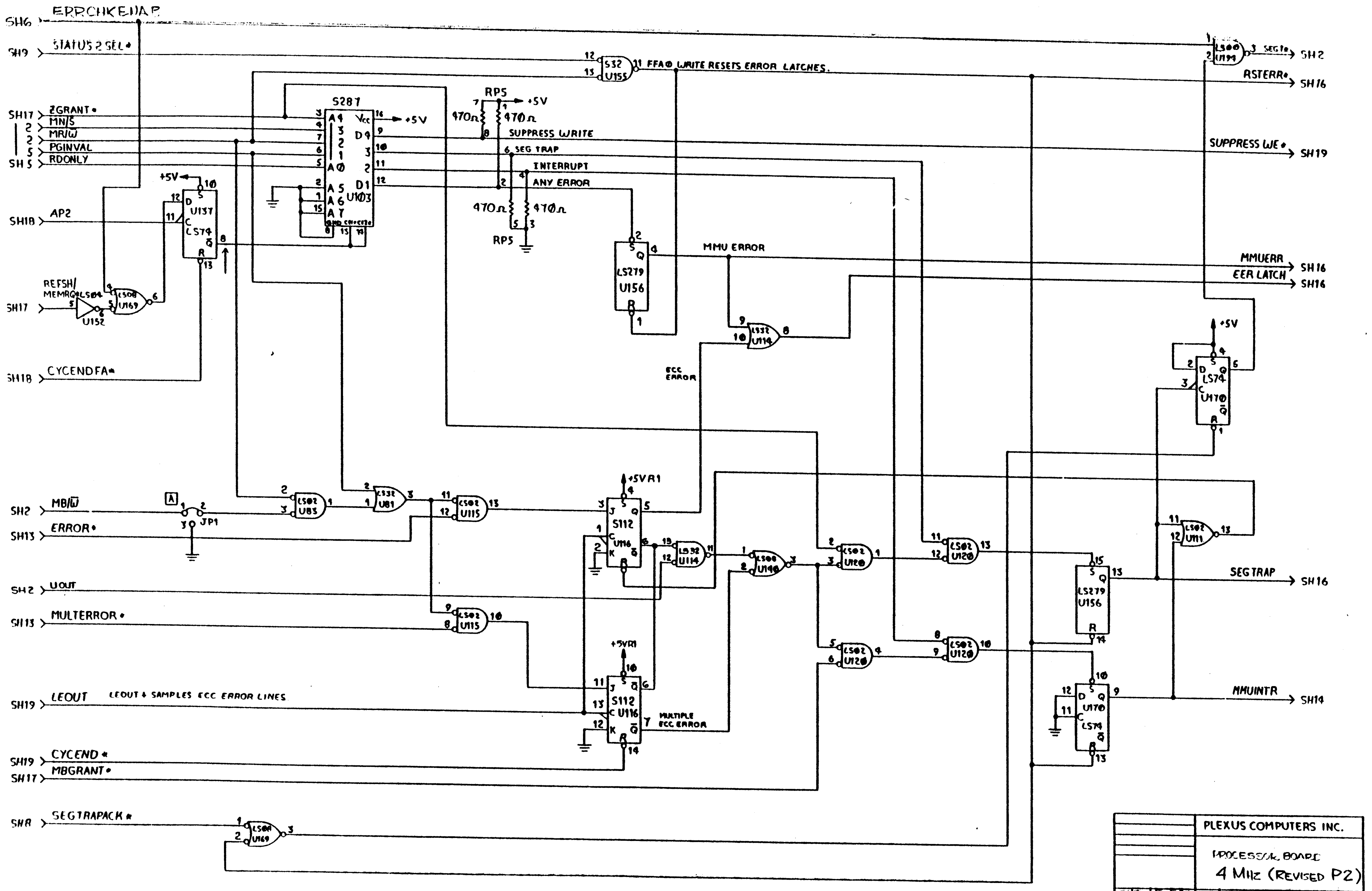


PLEXUS COMPUTERS INC.

PROCESSOR BOARD

4 MHz (REVISED P2)

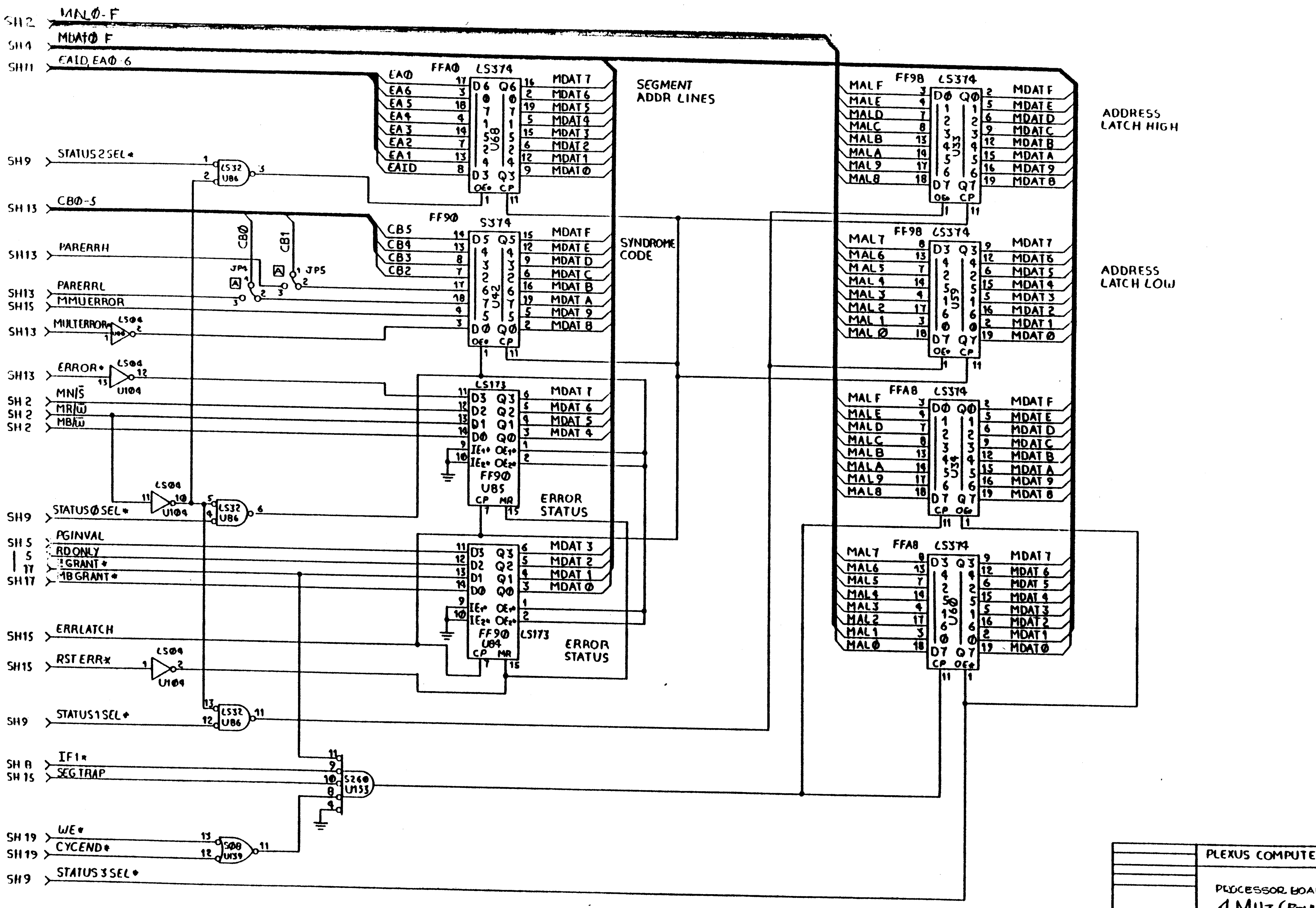
SIZE	SCALE —	60-00090 SH18 OF 20	A REV



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

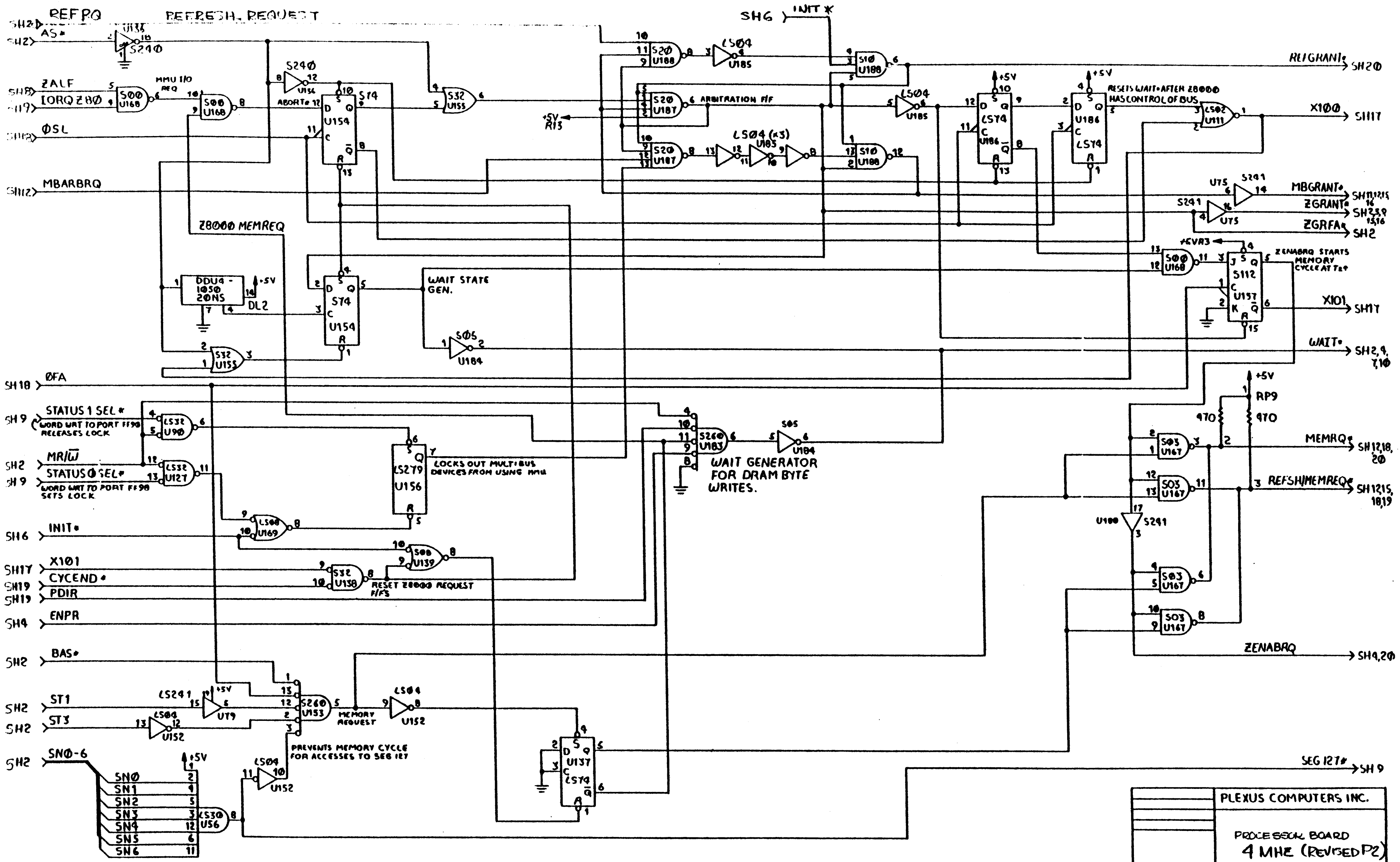
SHT 15 of 20 | 60-00090 | A



PLEXUS COMPUTERS INC

PROCESSOR BOARD
4 MHz (REVISED P2)

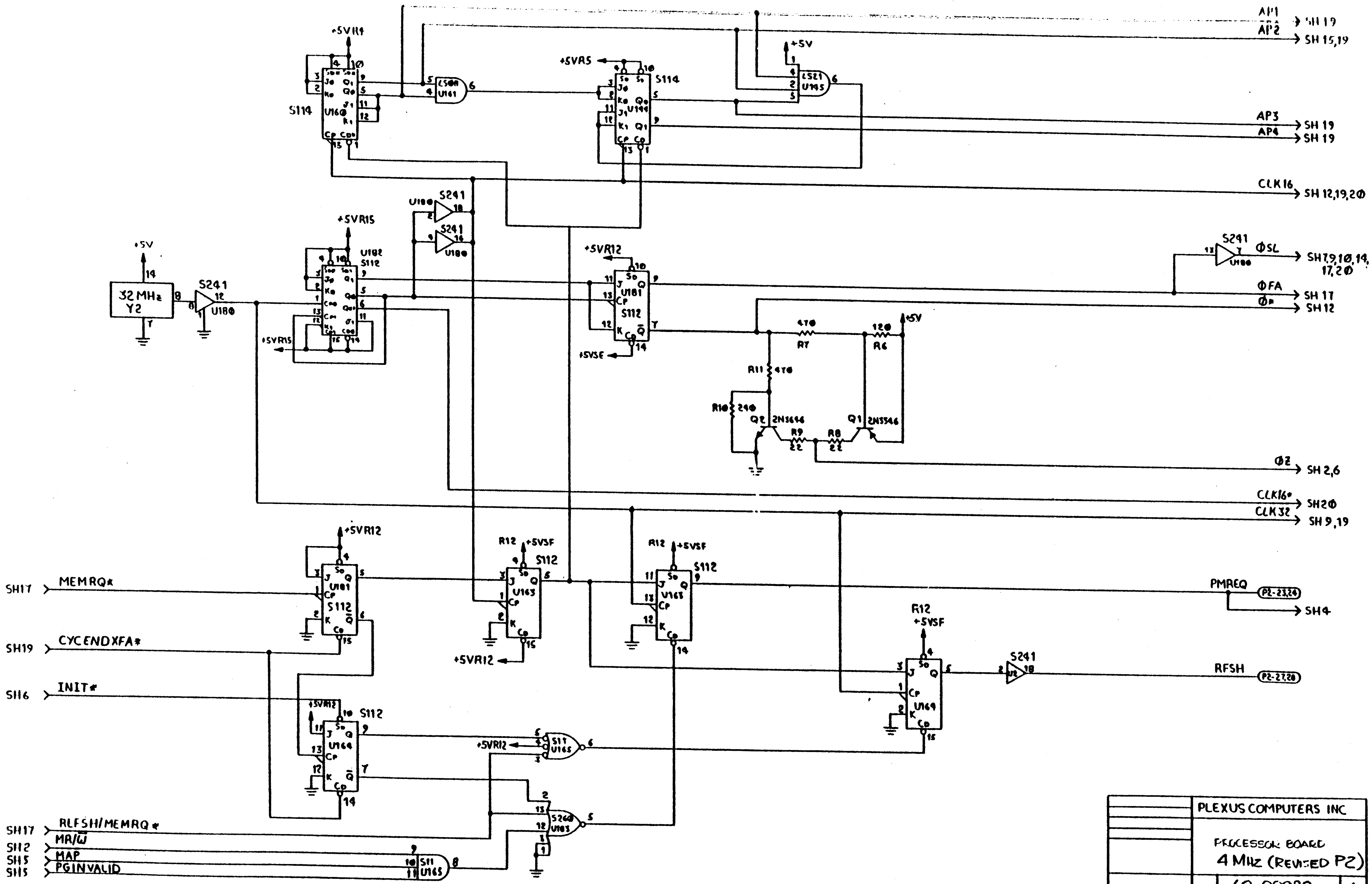
60-00090 | A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHZ (REVISED P2)

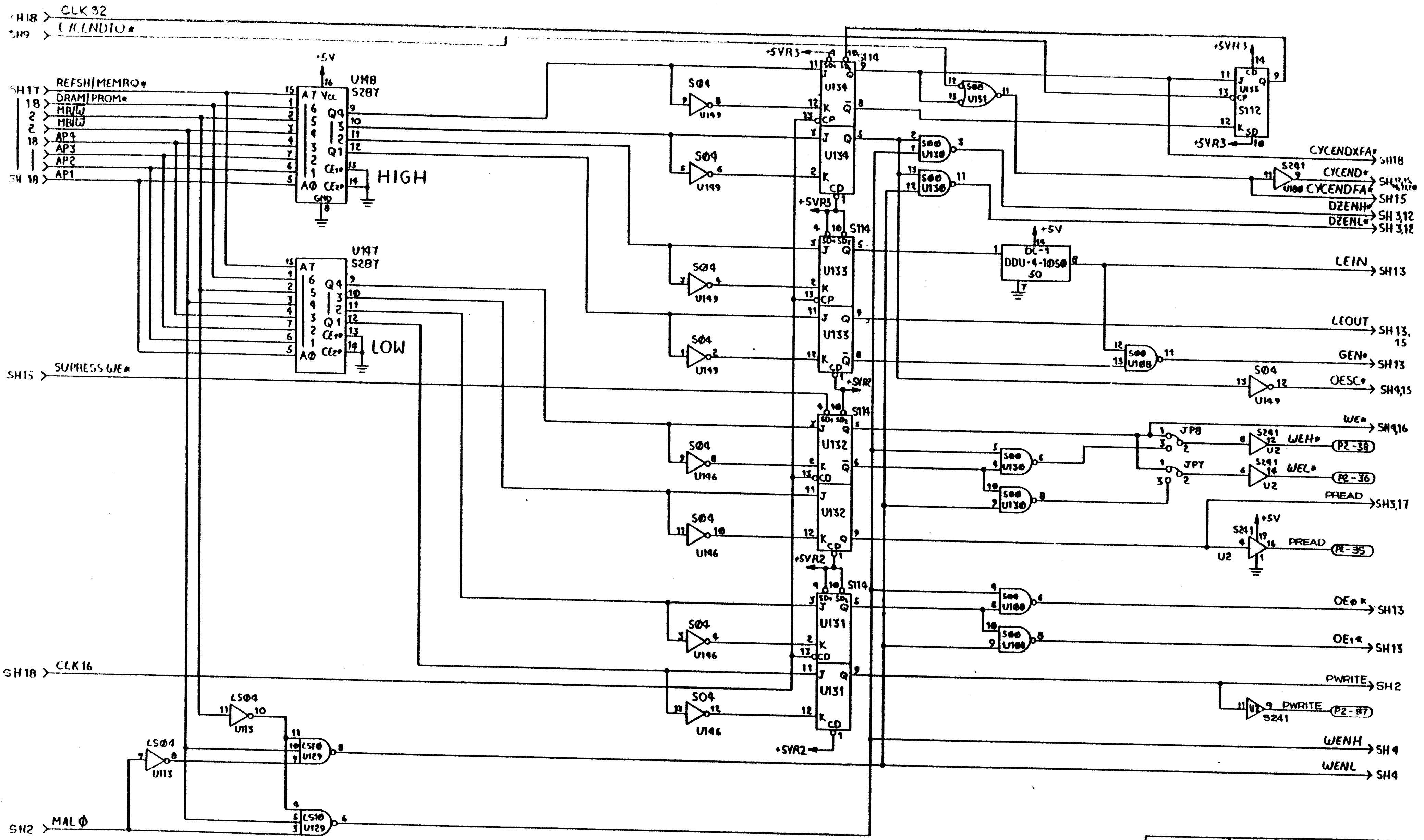
SIZE	SCALE —	60-00090 SH 17 OF 20	REV A



PLEXUS COMPUTERS INC

PROCESSOR BOARD
4 MHz (REVISED P2)

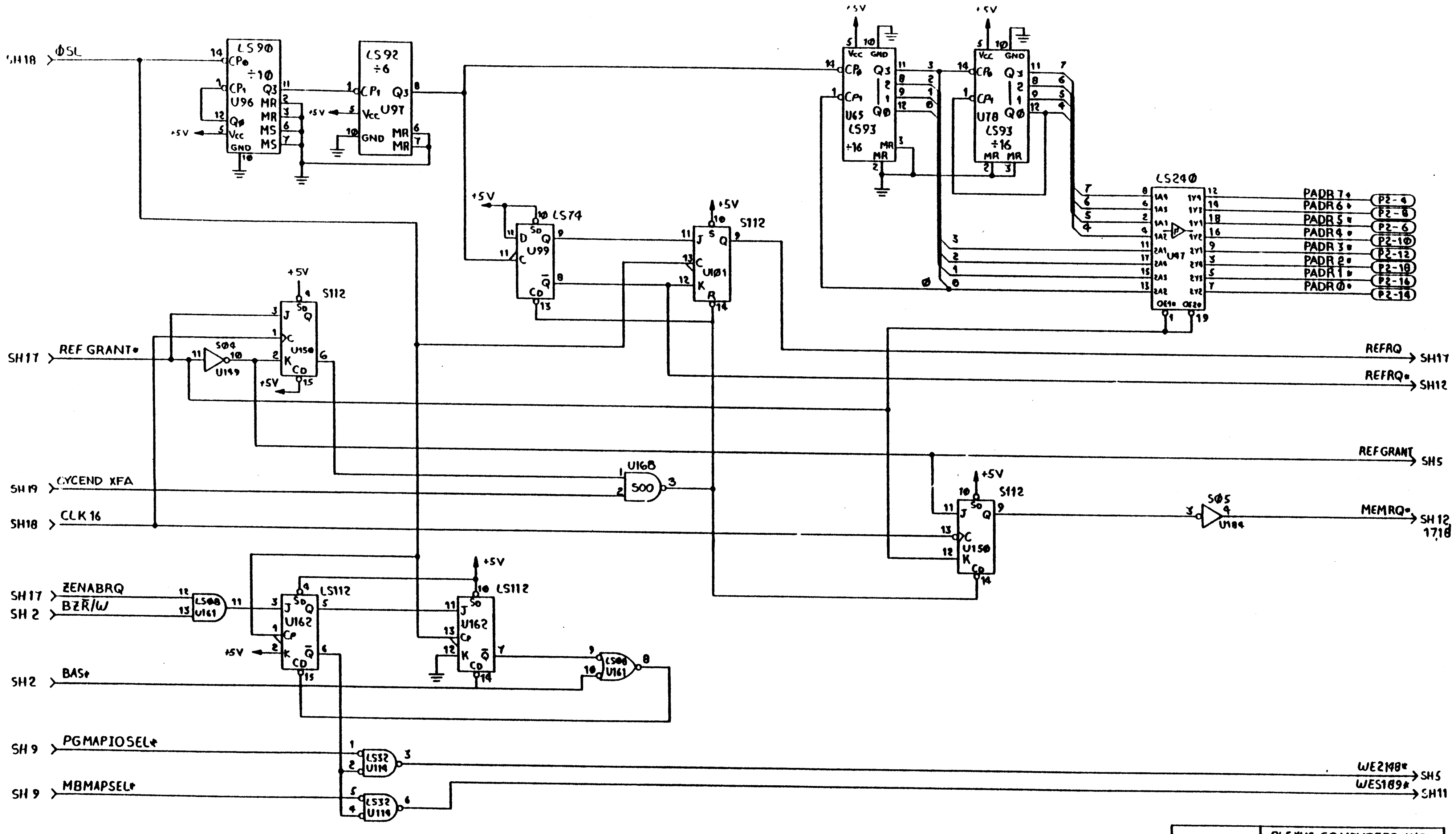
SIZE	SCALE —	SH 18 of 20	REV.
	60-00090		A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4MHz (REVISED P2)

SHT 19 OF 20 | 60-00090 | A



PLEXUS COMPUTERS INC.

PROCESSOR BOARD
4 MHz (REVISED P2)

SIZE	SCALE	60-00090	REV
	—	SH 30 OF 36	A

INPUT/OUTPUT CONTROLLER
(INTELLIGENT COMMUNICATION PROCESSOR)

1. ICP OVERVIEW

The I/O controller is an intelligent communication processor (ICP) which handles the interface between I/O devices and the CPU, leaving the CPU processor free to execute instructions. Each Plexus P/25 can be equipped with one or two ICPs; each ICP has connections for 8 serial ports (SIO) and 1 parallel port (PIO). The SIO ports are standard RS232C interfaces and the PIO port interfaces a Centronics (type) parallel printer.

The ICP is a double-height Multibus module that contains a 16-bit Z8002 microprocessor. In its memory space it has 32K bytes of RAM with parity, 16K bytes of PROM and 16K bytes mapped into system main memory. It communicates with the I/O devices through nine DMA ports and with the CPU processor through the Multibus.

The purpose of the ICP is to pass data between system memory (RAM) and I/O devices (terminals, printers, disk and tape drives), using DMA to bypass the CPU. The CPU provides instructions in the form of pointers to command and status blocks located in main memory. The ICP processor fetches the blocks and executes the instructions. This arrangement frees the CPU from handling typically slow I/O operations.

The ICP includes buffers to store the data between destinations and control logic to access the Multibus, over which all data is passed.

A block diagram (Figure 1) shows the basic organization of the ICP.

2. PROCESSOR

The ICP uses a Z8002 (unsegmented) 16-bit microprocessor operating at 4.0 Mhz. It is always operated in system mode and there is no separation of instruction and data nor of special and standard I/O spaces (special I/O space is not used). The ICP processor can directly address 64K bytes of memory. The lower 48K is resident on the ICP and the upper 16K is mapped into main memory via the Multibus and system MMU.

All three forms of interrupt -- nonmaskable, vectored and nonvectored -- are used by the ICP processor to keep informed of the ICP activities. Nonmaskable interrupts are only used to inform the ICP processor that a memory parity error or power failure has occurred. Vectored interrupts are used for a variety of purposes. Nonvectored interrupts are used to inform the Z8002 that it has control of the Multibus.

The processor and associated logic to interface the rest of the board is shown on page 2 of the ICP schematics. Paks U41 and U43 are bidirectional bus drivers used to drive/receive data into and out of the processor. They are active as long as the processor has the bus (BUSACK/ nonactive) with direction control coming from TO'CPU, a derivative of the processor's R/W and DS/ signals.

Paks U57 and U58 are latches used to hold the I/O or memory address while data transfer is taking place. These are unidirectional latches (out from the processor) which latch the address output from the processor (for the SIO, PIO, DMA, PROM etc). Paks U70 and U71 are control bus drivers which provide inverted and noninverted drive capability for the processor control bus used all over the ICP. Pak U128 generates the 4-Mhz clock required by the Z8002, the SIO, PIO, DMA and miscellaneous logic functions. This clock is MOS-compatible, as required by the Z8002.

Pak U92 decodes the four status lines coming out of the processor and pak U93 provides the ability to tristate IORQ/ and VIACK/ so that the DMA can take over the control bus when it is active. Note that both U92 and U93 are active only when I'BUSACK is not active (DMA does not have the bus). See the Zilog Z8000 CPU Technical Manual for complete details of the processor operation and timing.

The ICP has two separate buses going to and from the Z8002. The data goes through U41 and U43, and addresses go through U57 and U58. These are shown on page 2 of the ICP schematics, and in Figure 1.

3. LOCAL MEMORY

The ICP contains 32K bytes of dynamic RAM composed of two 16K-by-9 banks (including parity), from location 4000H to BFFFH. The memory is organized as shown on page 4 of the ICP schematics. The RAM chips are U5 to U13 and U22 to U29.

Refresh is controlled by the ICP processor. Parity generation and checking can be enabled or disabled by the processor. A parity error sets a latch (U86) and generates a nonmaskable interrupt to the processor. The parity latch can be read by the processor using I/O instructions and can be reset by writing to the parity error reset port (U110).

Paks U40 and U44 multiplex the 16-bit address coming out of the processor onto a 7-bit bus which goes directly to the dynamic RAM chips. Pak U76 generates the control signal such that latched addresses (LA)01 thru LA07 (low byte) are shipped off during the row address strobe (RAS) portion of the memory cycle. LA08 thru LA14 (high byte) are shipped off during the column address strobe (CAS) portion of the memory cycle. Refer to any 16K-by-1 dynamic RAM spec sheet for details of the timing.

LA00 is not multiplexed into the memory address. It is stripped off and stored in U60. During a byte access, it is used to determine whether the high or low byte of the word is being requested. The low byte is accessed if LA00 is low and the high byte is accessed if LA00 is high. During a word access, it is insignificant, as it only specifies the difference between the high and low bytes of a given word stored in dynamic RAM.

The processor utilizes positive-true logic on the B'AD bus. Paks U45 and U59 are bidirectional drivers to route data in and out of the memory chips, which have very little drive capability. Pak U60 provides high byte to low byte conversion during DMA.

The ICP has four sockets for PROM (U88 to U91), which provide from 4K to 16K bytes. The sockets accommodate either 16K or 32K EPROMs or ROMs. This choice has no effect on system performance.

No parity checking is performed on PROM accesses. PROM starts at location 0 and goes up to 3FFFH (RAM starts at 4000H). Paks U88 thru U91 contain the PROMs, which in production contain the following firm/soft ware:

U88 Low-byte ICP software

U89 Low-byte ICP diagnostics

U90 High-byte ICP software

U91 High-byte ICP diagnostics

Paks U66 and U79 are bidirectional drivers wired for unidirectional operation. They drive the PROM data out onto the processor B'ADxx bus (the PROMs also have very limited drive capability). The ROM/PROM circuit is shown on page 14 of the ICP schematics.

Pak U72 is the PROM configuration selector. To use 4K-by-8 PROMs, connect ROM 0 (A to B) and ROM 1 (A to C). To use 2K-by-8 PROMS, connect ROM 0 (A to C) and ROM 1 (A to B). See the P/40 User's Manual for details about the jumpers.

Memory on the ICP is divided into three addressable areas:

Hex Address	Description
0000-3FFF	PROM space
4000-BFFF	Dynamic RAM space
C000-FFFF	Multibus space

4. SERIAL PORTS

The ICP has eight RS232C serial ports implemented with Z80A SIO2 USARTs. The Z8002 selects asyncronous baud rates from 50 to 38,400 by programming a Zilog Z80A CTC Counter/Timer. Character length, parity and number of stop bits are also programmable. See the Zilog 1981 Data Book for details. The RS232C pins supported by the ICP are shown below:

RS232C Pin		SIO Pin	I/O
2	Transmit Data	RxD	I
3	Receive Data	TxD	O
4	Clear to Send	CTS*	I
5	Request to Send	RTS	O
6	Data Set Ready	DTR	O
7	Signal Ground	Gnd	-
8	Data Carrier Detect	+12,DCD**	I/O
11	Second RTS	CTS*	I
15	Transmit Clock	TxC***	I
17	Receive Clock	RxC***	I
20	Data Terminal Ready	CTS*	I

----

* CTS is used as the auto transmitter enable. It can connect to one of three sources: pin 4, pin 11 or pin 20. To select one, see the Plexus P/25 User's Manual.

** Jumper allows Data Carrier Detect to be connected to +12v (ON) or to the DCD pin of the modem.

*** Jumpers allow use of internal or external clock.

NOTE: The pins for all terminals (and the printer) leave the ICP from connectors J1 and J2. These are shown in I/O Controller Section 13.

Each serial port has a DMA channel associated with it. The DMA moves data from local memory to the SIO transmitter. A DMA transfer is initiated by the SIO every time its transmitter buffer becomes empty while it is in active mode. A rotating priority scheme is used so that the most recently serviced SIO has lowest priority.

The DMA function is provided by three AMD9517 DMA chips (U106-U108). These have only an 8-bit data path; to get the contents of the appropriate memory location, each DMA chip latches the high-order byte of the address into an address latch (U80, U95 and U96) during the early part of the DMA transfer cycle. It then puts out the low-order byte of the address on its own address line. See the AM9500 Peripheral Products Interface Guide for operation and timing details on the DMA chips.

The DMA chip only has one cycle for both retrieving the memory data and writing the data to the SIO chip. It uses pak U115 as a fake port address generator. When DMA'WRITE is inactive (no DMA in progress), this multiplexor outputs LA00 and LA01 to MLA00 and MLA01. This is normal; it allows the Z8002 to address the CTCs, SIOs and PIO in the normal manner. However when DMA'WRITE is active (a DMA is in progress) the DMA chip takes over the address generation of MLA00 and MLA01 and addresses the appropriate SIO (or PIO).

The CTCs are included in the MLA00 and MLA01 circuit so that they cannot be selected accidentally with a simultaneous memory address and IORQ/ from the DMA. Note that ICP DMA only goes from local memory out to the SIOs; it writes to the terminals but doesn't read from the terminals. This is because there exists an order of magnitude more activity going out to the terminals than returning. Input from the terminals is limited by number of terminals times typing speed.

The combination of DMA and the sophisticated AMD9517 chips (U106, 7 and 8) protects the Z8002 from handling too many interrupts. The following discussion describes how:

> Assume that output to a 1000-lpm printer is equivalent to one terminal's output. Each ICP board must support nine of these. If all terminals are set to 19,200 baud (almost 2000 characters per second), then the total load to the ICP is 18000 characters per second, which translates to 18000 interrupts per second, or a 55-microsecond-per-interrupt service time.
>
> The Z8002 requires 2 microseconds per instruction (clocked at 4 Mhz). This allows only 28 instructions to decide which character to output and to output it. Even if the Z8002 could keep up, it would have time for nothing else (such as multiprocessing, screen formatting and special character

handling). The addition of DMA on the ICP lowers the number of character interrupts by 95%.

The AMD 9517 DMA chips are wired to provide three DMA channels each , so that three of them provide the 9 DMA required channels. The INTEL 8237 is a lookalike for this chip.

The DMA chips are programmed by the processor to transfer one byte and then release the buses. This assures that the Z8002 will get at least every other memory cycle. When an entire block of data has been transferred, the DMA generates a vectored interrupt to the processor through a CTC. This informs the processor that the DMA is done and the 9517 is ready for another task.

When a character is received by an SIO, a vectored interrupt is sent to the ICP processor.

Paks U102 through U105 are LSI SIO chips. Each chip has two channels on board. These are completely separate and only coincidentally share the same physical chip.

Both the processor and the DMA can program the SIO chips with terminal characteristics. The SIOs have a byte-wide data path and are programmed by a Z8002 OUT instruction to the appropriate port. Even though it is a word instruction, only the lower 8 bits are used.

5. PARALLEL PORT

The ICP has a single parallel I/O port that is used to connect an industry-standard (Centronics-type) line printer. The port is based on the Z80A PIO and is also DMA-supported. Control logic associated with the port generates all control and handshake signals necessary for the DMA to transfer data to a printer without intervention from the Z8002. The lines of the parallel port are shown below:

Signal Pin	Return Pin	Source ICP/Ptr	PIO Pin	Signal* Name
1	19	ICP	7	DATA Strobe/
2	20	ICP	27	DATA 1
3	21	ICP	28	DATA 2
4	22	ICP	29	DATA 3
5	23	ICP	30	DATA 4
6	24	ICP	31	DATA 5
7	25	ICP	32	DATA 6
8	26	ICP	33	DATA 7
9	27	ICP	34	DATA 8
10	28	Ptr	6	Acknowledge/
11	29	Ptr	14	BUSY
12	-	Ptr	13	Paper Out (PE)
13	-	Ptr	12	Online (SLCT)
14	-	-	-	Ground
16	-	-	-	Logic Ground
17	-	-	-	Chassis Ground
31	30	ICP	15	Input Prime (RESET)
32	-	Ptr	10	FAULT

* All signals are standard TTL levels terminated according to the _Centronics Interface Specification_. Pin numbers refer to Amphenol 57-30360 36 pin connector.

The port sends a vectored interrupt to the ICP processor if the printer asserts the FAULT line.

6. MULTIBUS ARBITER AND INTERFACE

The Multibus arbiter on the P/25 uses the parallel arbitration technique described in the _IEEE Proposed Microcomputer System Bus Standard_ (_P796 Bus_). The arbiter is really distributed; each Multibus board is responsible for doing part of the arbitration. The common arbitration circuitry resides on the P/25 backplane.

The ICP uses its block move capability to move data from the middle 32K of its memory space (local RAM) to the upper 16K which is mapped into system RAM. To do this, it programs a CTC timer circuit to request the Multibus every time it reaches a zero count (BUS'796'ARB'REQ at U99). When control of the Multibus is obtained, the arbiter locks 796'BUSY/ (U84) and interrupts the ICP processor (thru the CTC at U98 -- signal name BUS'ARB'INT'TRG). The interrupt informs the ICP processor that it can begin a transfer to or from main memory. When the transfer is complete, the ICP processor resets the arbiter and releases the Multibus by setting its multi-micro pin (U34).

The ICP Multibus interface is a master which supports only 16-bit memory read and write operations. It allows the Z8002 to assert one of the eight Multibus interrupt lines (INT0/ - INT7/). The interrupt line is selected by a jumper in a socket at U17. However, the value is not optional; it must be set as specified in the _Plexus P/25 User's Manual_.

The ICP can also be a Multibus slave. This happens when it is instructed by the CPU to execute an I/O write.

The interface initiates a Multibus memory cycle every time the ICP processor performs a memory operation in the upper 16K bytes of memory (C000H-FFFFH). Multibus addresses are formed by concatenating a six-bit port (U48) with the least-significant 14 processor address bits (U16, U64, U65) to form a 20-bit Multibus address. This port (U48) can be written by the ICP processor using standard I/O instructions.

7. INTERRUPTS

The ICP uses nonmaskable, vectored and nonvectored interrupts. The events that cause each type of interrupt are as follows:

Nonmaskable interrupt -- Because there are valid reasons for masking most interrupts, the nonmaskable interrupt is only used when a memory parity error or power failure is detected.

Vectored interrupt -- This is the most versatile interrupt, used for channel attention, DMA end of process, realtime clock, PIO cycle complete, PIO fault, SIO transmit, SIO receive, SIO external status and SIO special (break, lost carrier, etc.).

Nonvectored interrupt -- The nonvectored interrupt has only one purpose: to inform the Z8002 that its request for the Multibus has been accepted.

7.1. Nonmaskable Interrupts

Nonmaskable interrupts get their name from the fact that there are no instructions that the processor can execute to protect itself from them; when they occur, the processor is interrupted regardless of other conditions. Because there are valid software reasons for disabling most interrupts, the only nonmaskable interrupts in the ICP are caused by memory parity errors in the local ICP memory and by power failures.

7.2. Vectored Interrupts

Vectored interrupts interrupt the processor and then send it a vector which identifies the source of the interrupt. The interrupt routine depends on the vector, making these the most flexible interrupts.

The ICP uses a serial daisy-chain form of interrupt priority. The priority of service depends on location in the daisy chain, which is shown in the table below. The daisy chain is implemented with the interrupt enable in (IEI) and interrupt enable out (IEO) signals, which exist on each CTC, SIO and PIO.

The sequence of events is as follows: When a particular channel of a peripheral chip needs to interrupt the ICP processor, it first examines its own priority relative to other onboard channels by polling its IEI line. If the IEI is active (high), then no higher priority chip is interrupting and it may proceed. If the IEI is not active, then it must wait.

Assuming that IEI is active, the chip simultaneously pulls its IEO line (active low) inactive and VEC'INT/ (active low) active. This interrupts the ICP processor, causing it to issue an interrupt-acknowledge cycle. Upon receipt of the interrupt-acknowledge, the peripheral chip returns its interrupt vector (previously programmed into it) to the ICP processor and deactivates VEC'INT/. The peripheral chip then waits for the return from interrupt (RETI) signal. (This Z80 standard signal is produced by a special port, as described in the I/O Controller section 7.4.).

Upon receipt of RETI, the interrupt cycle ends. The chip returns its IEO to active, enabling other interrupts.

The following table shows the devices in the daisy chain and their priorities:

Interrupt	Priority	Device	Sources
HIGH	0	SIO 0	Chan A Serial Port 0
	1	SIO 0	Chan B Serial Port 1
	2	SIO 1	Chan A Serial Port 2
	3	SIO 1	Chan B Serial Port 3
	4	SIO 2	Chan A Serial Port 4
	5	SIO 2	Chan B Serial Port 5
	6	SIO 3	Chan A Serial Port 6
	7	SIO 3	Chan B Serial Port 7
	8	CTC 0	Chan 3 Channel Attention Port
	9	CTC 1	Chan 3 Bus Arbiter Interrupt
	10	CTC 2	Chan 2 DMA0 END OF XFER
	11	CTC 2	Chan 3 DMA1 END OF XFER
	12	CTC 3	Chan 2 Realtime Clock Ticks
	13	CTC 3	Chan 3 DMA2 END OF XFER
LOW	14	PIO	Chan A Parallel Port Status

7.3. Nonvectored Interrupts

These interrupts can be masked, but they do not return an interrupt vector. All nonvectored interrupts jump to the same interrupt handler, which takes a poll to find the source of the interrupt. The only nonvectored interrupt used by the ICP occurs when it receives control of the Multibus. This signal, which originates at U98, informs the Z8002 that it can begin a transfer to or from main memory.

7.4. Programming Information

A translation circuit makes the interrupt acknowledge produced by the Z8002 compatible with the Z80 peripheral chips. A similar translation circuit (see the ICP schematics, page 7) simulates the RETI instruction of the Z80 microprocessor.

When a peripheral chip sends an interrupt, the ICP processor issues an interrupt-acknowledge cycle (M1/ and IORQ/ active simultaneously). The peripheral chip responds with its own vector address on the low-order data bus (B'ADxx). The Z8002 processes the interrupt and then initializes the RETI (ED4D) opcode for which the peripheral chip is waiting, along with a special sequence of control lines (M1/ active and IORQ/ inactive). The peripheral chip sees these control lines and ED, followed by 4D on the bus. It interprets this as an interrupt reset and releases the interrupt-priority chain.

Because the Z8002 does not have the RETI opcode that the Z80 peripheral chips require, this signal is generated by writing to a special interrupt-reset port. The port outputs ED at the same time M1 goes active and IORQ/ goes inactive; then it outputs 4D. To the peripheral chips this circuit it transparent; the signal appears to come from the Z8002.

8. STATUS AND CONTROL PORTS

The ICP processor has access to several status and control ports using standard word I/O instructions. These are:

# Bits	R/W	Purpose
4	R	Switches for diagnostics and initialization
16	R	Switches and jumpers that select channel channel attention port I/O address and wake-up address
4	W	LEDs for diagnostics
1	W	Enable/disable parity generation and checking
1	W	Reset parity error
6	W	Latch holding six most significant Multibus address bits
8	R	Channel attention port
1	W	Assert one of eight Multibus interrupt lines
-	W	Daisy chain RETI port
1	W	Initialize PIO DMA request
1	R	Parity error vs power fail interrupt

The addresses of these ports are shown in the I/O Controller, section 9.

9. I/O SPACE

In addition to the status and control ports discussed in the previous part, the ICP processor has the following eight-bit LSI peripheral chips in its standard I/O space:

4	USARTs (2 SIOs each)
1	parallel printer port (PIO)
4	4-channel counter/timers (CTC)
3	4-channel DMA controllers (9517)

Although these are 8-bit devices, they are referenced only with word I/O instructions. This is a peculiarity of the Z8002; its ability to do byte-writes to certain ports is limited. The high byte is ignored.

A bus translator converts the Z8000 control bus to the Z80 control bus so that these peripheral chips can be used. A swap-byte buffer allows data transfers from the high bytes of memory.

A summary of the ICP port addresses and their functions is shown below:

Address	# Bits	R/W	Port	Function
4000	8	R/W	CTC0 - chn 0	Serial Port 0 Baud Rate
4001	8	R/W	CTC0 - chn 1	Serial Port 1 Baud Rate
4002	8	R/W	CTC0 - chn 2	Serial Port 2 Baud Rate
4003	8	R/W	CTC0 - chn 3	Channel Attn Interrupt
4010	8	R/W	CTC1 - chn 0	Serial Port 3 Baud Rate
4011	8	R/W	CTC1 - chn 1	Serial Port 4 Baud Rate
4012	8	R/W	CTC1 - chn 2	Serial Port 5 Baud Rate
4013	8	R/W	CTC1 - chn 3	Multibus Arbiter Interrupt
4020	8	R/W	CTC2 - chn 0	Multibus Arbiter Request
4021	8	R/W	CTC2 - chn 1	Realtime Clock - Level 1
4022	8	R/W	CTC2 - chn 2	DMA0 End of Xfer Interrupt
4023	8	R/W	CTC2 - chn 3	DMA2 End of Xfer Interrupt
4030	8	R/W	CTC3 - chn 0	Serial Port 6 Baud Rate
4031	8	R/W	CTC3 - chn 1	Serial Port 7 Baud Rate
4032	8	R/W	CTC3 - chn 2	Realtime Clock Interrupt
4033	8	R/W	CTC3 - chn 3	DMA2 End of Xfer Interrupt

4040	8	R/W	SIO0 - chn A Data	Serial Port 0
4041	8	R/W	SIO0 - chn B Data	Serial Port 1
4042	8	R/W	SIO0 - chn A Cntl	Serial Port 0
4043	8	R/W	SIO0 - chn B Cntl	Serial Port 1
4050	8	R/W	SIO1 - chn A Data	Serial Port 2
4051	8	R/W	SIO1 - chn B Data	Serial Port 3
4052	8	R/W	SIO1 - chn A Cntl	Serial Port 2
4053	8	R/W	SIO1 - chn B Cntl	Serial Port 3
4060	8	R/W	SIO2 - chn B Cntl	Serial Port 4
4061	8	R/W	SIO2 - chn B Data	Serial Port 5
4062	8	R/W	SIO2 - chn A Cntl	Serial Port 4
4063	8	R/W	SIO2 - chn B Cntl	Serial Port 5
4070	8	R/W	SIO3 - chn A Data	Serial Port 6
4071	8	R/W	SIO3 - chn B Data	Serial Port 7
4072	8	R/W	SIO3 - chn A Cntl	Serial Port 6
4073	8	R/W	SIO3 - chn B Cntl	Serial Port 7
4080	16	R	Switch jumpers	Wakeup Address
4090-F	8	R/W	8237 DMA0	Internal
4190-F	8	R/W	8237 DMA0	Internal
4290-F	8	R/W	8237 DMA0	Internal
40A0	8	R	Switch Register	D7 to D4-Diagnostic Switches D3-Parity Error
40B0	8	W	Outlatch 1	D7 to D4-Diagnostic LEDs D3-Initialize PIO DMA Request D2-Enable Parity (1=Enable) D1-Reset Parity (1=Reset) D0-Enable Multibus Interrupt
40C0	8	W	Outlatch 2	D0 to D5-6 MSBs of Multibus Address

40D0	8	R	Inlatch 1	Channel Attention Port
				D0=Reset
				D1=Start Command

				Execution D2=Clear Multibus Address D3 to D7-Unused
40E0	8	R/W	PIO0 - chn A Data	Printer Status
40E1			PIO0 - chn B Data	Printer Status
40E2			PIO0 - chn A Cntl	Printer Status
40E3			PIO0 - chn B Cntl	Printer Status
40F0	-	W		Daisy Chain Interrupt Reset

10. CHANNEL ATTENTION PORT

The channel attention port is at the slave address placed on the Multibus by a master to access an ICP. Each ICP has a unique address, determined by setting the switches in U68. This binary value serves a dual purpose: it is the ICP channel attention port address and it is the low-order byte of the wakeup address where the ICP seeks its first instructions during initialization. The switches represent a binary value between 0 and 255 (8 binary bits).

CAUTION: The value determined by U68 is NOT a customer option. Each ICP must be set according to its slot location (ICP 1 or 2). The proper switch settings are listed in the Plexus P/25 User's Manual.

A vectored interrupt is sent to the Z8002 whenever a Multibus master writes to the channel attention port. The port is 8 bits wide and is read by the Z8002 using I/O instructions. The action taken depends on the state of the first three bits. If bit 0 is high, it causes the ICP to reset itself. If bit 1 is high, the ICP performs a standard channel attention; it goes to the channel attention address, reads the instructions that the CPU has placed there and executes them. If bit 2 is high, the ICP's Multibus interrupt line (INTx/) is reset. This is used by the CPU processor to clear a Multibus interrupt posted by the ICP.

The remaining five bits are not used.

The CPU loads a block of commands into the channel attention address before it issues a channel attention to the ICP. Typically, the command block contains instruction to move data from one location to another (for example - "Move 300 words from system RAM to terminal 3 starting at address X and ending at Y").

10.1. Reset

The ICP can be reset by setting bit 0 in the channel attention port or by asserting the Multibus INIT/ line on connector P1. When the reset is released, the ICP executes its self-test, which takes about 15 seconds.

11. PSEUDO DMA

The ICP does not use true DMA to move blocks between local and main memory. Instead, it uses a combination of hardware and software.

Data is moved to and from main memory using block move instructions executed by the ICP processor. The upper 16K bytes of the ICP processor memory address space is mapped into main memory. To move data to main memory, the processor executes a block move from the middle 32K of memory (its local RAM) to the upper 16K. To initiate a transfer, the processor programs a CTC channel to provide a pulse every 0.5 microseconds. This pulse goes to the Multibus arbiter and causes it to request the Multibus. When control of the Multibus is obtained, the arbiter sends a vectored interrupt to the processor. The processor then initiates a block move of m words (m=~32). When the move is complete, the processor signals the arbiter to release the bus by resetting its multi-micro bit.

This technique allows a pseudo DMA task to run in the background without monopolizing either the Multibus or local bus. When no blocks remain in the DMA queue, the processor disables the CTC channel so that no more interrupts are generated. With the parameters mentioned above, the pseudo DMA can move 64K characters/sec using less than 10% of the processor bandwidth. This is fast enough to keep up with 16 terminals running at 19.2K baud plus two line printers at 300 lpm, which is the maximum load the P/25 will ever see.

The number of words in a block move is set in software using the LDIR (load direct) instruction.

12. PINS, CONNECTORS AND PINOUTS

The electrical level and timing interface for connector P1 is documented in the *IEEE Proposed Microcomputer System Bus Standard (P796 Standard)*. The following exceptions exist to this standard:

P1-77 Signal name = AACK/ used by the MPU and ICP

P1-78 Signal name = PWR'FAIL used by the MPU, ICP et al.

The ICP does not use connector P2; all the P2 pins on the ICP are open.

Connectors J1 and J2 connect the I/O pins on the ICP to the terminal connectors on the back panel. The electrical interface for all signals on connector J1 and connector J2 pins J2-1 to J2-38 is RS232C compatible at levels of +12 volts and -12 volts. Connector J2 pins J2-39 to J2-60 are TTL-compatible and are described in the *Centronics Interface Specification*.

12.1. Connector Pinouts

Connectors J1 and J2 are mounted on the front of the ICP. Ribbon cables (86 and 60 pins) connect them to the terminal connectors on the back panel. Pinouts for J1 and J2 are as follows:

Pin #	Signal Name	Pin #
J1- 1	Not Used	No Connection
J1- 2	Not Used	No Connection
J1- 3	Not Used	No Connection
J1 -4	Ground	T3- 7
J1- 5	B'RXCB.SIO1	T3-17
J1- 6	B'TXCB.SIO1	T3-15
J1- 7	B'DCDB.SIO1	T3- 8
J1- 8	B'DTRB.SIO1	T3- 6
J1- 9	B'RTSB.SIO1	T3- 5
J1-10	B'CTSB.SIO1	T3- 4
J1-11	B'CTSB.SIO1	T3-11
J1-12	B'CTSB.SIO1	T3-20
J1-13	B'RXDB.SIO1/	T3- 2
J1-14	B'TXDB.SIO1/	T3- 3
J1-15	Not Used	No Connection
J1-16	Ground	T2-7
J1-17	B'RXCA.SIO1	T2-17
J1-18	B'TXCA.SIO1	T2-15
J1-19	B'DCDA.SIO1	T2-8

J1-20	B'DTRA.SIO1	T2-6
J1-21	B'RTSA.SIO1	T2-5
J1-22	B'CTSA.SIO1	T2-4
J1-23	B'CTSA.SIO1	T2-11
J1-24	B'CTSA.SIO1	T2-20
J1-25	B'RXDA.SIO1/	T2-2
J1-26	B'TXDA.SIO1/	T2-3
J1-27	Not Used	No Connection
J1-28	Ground	T1-7
J1-29	B'RXCB.SIO0	T1-17
J1-30	B'TXCB.SIO0	T1-15
J1-31	B'DCDB.SIO0	T1-8
J1-32	B'DTRB.SIO0	T1-6
J1-33	B'RTSB.SIO0	T1-5
J1-34	B'CTSB.SIO0	T1-4
J1-35	B'CTSB.SIO0	T1-11
J1-36	B'CTSB.SIO0	T1-20
J1-37	B'RXDB.SIO0	T1-2
J1-38	B'TXDB.SIO0/	T1-3
J1-39	B'RXDA.SIO0/	T0-17
J1-40	Ground	T0-7
J1-41	B'TXCA.SIO0	T0-15
J1-42	B'DTRA.SIO0	T0-6
J1-43	B'DCDA.SIO0	T0-8
J1-44	B'RTSA.SIO0	T0-5
J1-45	B'CTSA.SIO0	T0-4
J1-46	B'CTSA.SIO0	T0-11
J1-47	B'CTSA.SIO0	T0-20
J1-48	B'RXDA.SIO0/	T0-2
J1-49	B'TXDA.SIO0/	T0-3
J1-50	Ground	T4-7
J1-51	B'RXCA.SIO2	T4-17
J1-52	B'TXCA.SIO2	T4-15
J1-53	B'DCDA.SIO2	T4-8
J1-54	B'DTRA.SIO2	T4-6
J1-55	B'RTSA.SIO2	T4-5
J1-56	B'CTSA.SIO2	T4-4
J1-57	B'CTSA.SIO2	T4-11
J1-58	B'CTSA.SIO2	T4-20
J1-59	B'RXDA.SIO2/	T4-2
J1-60	B'TXDA.SIO2/	T4-3
J2-1	Not Used	No Connection
J2-2	Not Used	No Connection
J2-3	Not Used	No Connection
J2-4	Ground	T7-7
J2-5	B'RXCB.SIO3	T7-17
J2-6	B'TXCB.SIO3	T7-15
J2-7	B'DCDB.SIO3	T7-8
J2-8	B'DTRB.SIO3	T7-6
J2-9	B'RTSB.SIO3	T7-5

J2-10	B'CTSB.SIO3	T7-4
J2-11	B'CTSB.SIO3	T7-11
J2-12	B'CTSB.SIO3	T7-20
J2-13	B'RXDB.SIO3/	T7-2
J2-14	B'TXDB.SIO3/	T7-3
J2-15	Not Used	No Connection
J2-16	Ground	T6-7
J2-17	B'RXCA.SIO3	T6-17
J2-18	B'TXCA.SIO3	T6-15
J2-19	B'DCDA.SIO3	T6-8
J2-20	B'DTRA.SIO3	T6-6
J2-21	B'RTSA.SIO3	T6-5
J2-22	B'CTSA.SIO3	T6-4
J2-23	B'CTSA.SIO3	T6-11
J2-24	B'CTSA.SIO3	T6-20
J2-25	B'RXDA.SIO3/	T6-2
J2-26	B'TXDA.SIO3/	T6-3
J2-27	Not Used	No Connection
J2-28	Ground	T5-7
J2-29	B'RXCB.SIO2	T5-17
J2-30	B'TXCB.SIO2	T5-15
J2-31	B'DCDB.SIO2	T5-8
J2-32	B'DTRB.SIO2	T5-6
J2-33	B'RTSB.SIO2	T5-5
J2-34	B'CTSB.SIO2	T5-4
J2-35	B'CTSB.SIO2	T5-11
J2-36	B'CTSB.SIO2	T5-20
J2-37	B'RXDB.SIO2/	T5-2
J2-38	B'TXDB.SIO2/	T5-3
J2-39	PDAT7	PAR-9
J2-40	PDAT6	PAR-8
J2-41	PDAT5	PAR-7
J2-42	PDAT4	PAR-6
J2-43	PDAT3	PAR-5
J2-44	PDAT2	PAR-4
J2-45	PDAT1	PAR-3
J2-46	PDAT0	PAR-2
J2-47	PBRDY/	PAR-1
J2-48	PBSTB/	PAR-10
J2-49	POSCXT	PAR-15
J2-50	PFAULT/	PAR-32
J2-51	PSLCT	PAR-13
J2-52	PPE	PAR-12
J2-53	PBUSY	PAR-11
J2-54	PINPPRM	PAR-31
J2-55	Ground	PAR-14
J2-56	Ground	PAR-16
J2-57	Ground	PAR-19
J2-58	Ground	PAR-20
J2-59	Ground	PAR-21
J2-60	Ground	PAR-22-30

13. DMA OPERATION

Refer to the ICP schematics, the block diagram and a copy of the AMD9517 specifications as necessary for this discussion.

The ICP uses true DMA to transfer data from local memory to the SIOs and PIO.

The DMA chips are programmed similarly to the SIOs, CTCs and PIO. They have a few different registers, but they use similar I/O commands and the hardware is identical. However, the actual process of transferring data from local memory to the SIOs and PIO is unlike anything on the board. It proceeds as follows:

(a) The ICP processor programs the DMA chip with a transfer type, a word count and a starting address.

(b) The DMA waits until one of the SIOs (or PIOs; they work similarly) sends it a RDYX.SIOY/ signal which informs the DMA that the transmitter buffer on the SIO is empty. This is the normal case for an SIO when it is not performing work and is the case every time it moves a character from the internal transmit buffer, where the DMA puts the character, to the shift register where the character is actually output onto the RS232C transmit line.

(c) When this READY signal originates at the SIO, it causes the DMA to fetch another character from local memory. The DMA activates its RQ line (which is passed through the DMA at U106), which goes to the BUSREQ/ input of the Z8002 and demands control of all the ICP buses. The Z8002 yields control of the buses to the DMA at the end of its current machine cycle, by tristating everything and activating BUSACK/.

(d) The DMA outputs its AEN signal to activate its high-order address latch. The DMA only has 8 address lines and it needs to provide a 16-bit address to memory, so it puts the high-order address bits on its data bus and latches them into the accessory address latches by activating its ADSTB signal.

(e) Besides latching the high-order address bits into the accessory latch, the ADSTB signal serves as the normal processor address strobe; it starts off the memory cycle along with the normal RAS and CAS signals. It is then used by pak U142 to generate a DMA'RDY signal that slows the DMA down to match the memory speed. After the prerequisite amount of time (about 500 microseconds after ADSTB goes active), the DMA generates its DACK signal to tell the SIO chip that there is a valid character on the B'ADxx bus.

(f) The DMA forces the ICP processor to tristate (give up control of) the bus by activating BUSREQ/. It then takes over, performing all activities necessary to write the character into the SIO chip. It releases its AEN and HRQ signals and waits for the SIO to request another character. During the time the DMA chip has control of the buses, the memory and peripheral chips function as they would if the ICP processor were in control.

14. ICP SOFTWARE CONCEPTS

This section provides an overview of the functions performed on the ICP board and the order in which they must be performed. This section is not intended to be exhaustive or complete. For details of the ICP software, see the Plexus software documentation.

The main strengths of the ICP are as follows:

(a) It is intelligent -- it has a Z8002 16-bit microprocessor on board.

(b) It has 32K bytes of DRAM memory on board, all for software use.

(c) It has pseudo-DMA access to the system memory without intervention of the CPU.

(d) It has true DMA from ICP local memory out to the nine ports.

To make use of these strengths, the ICP works in the following way: The Z8002 and the 32K bytes of DRAM allow the ICP to locally buffer large quantities of data (a screenful for each of eight CRTs), handle the character interrupts coming back from each terminal and notify the CPU only when an entire screenful of information is completed for a terminal.

NOTE: If it were necessary to put out a full CRT screen every time, this arrangement could cut the character-interrupt handling time by a factor of 1920. Because most information is output in smaller than screenful blocks, this arrangement typically cuts CPU interrupts by a factor of 20.

This basic idea is implemented in several steps. First, the CPU builds a command block in system memory at the ICPs channel attention address. This may take approximately 100 memory accesses, but does not occupy the Multibus. Then the CPU interrupts the ICP with an single I/O write over the Multibus. This interrupt is called a channel attention. It causes the ICP to go to the channel attention address and read the instructions there. This takes only 1 or 2 Multibus accesses of 64 bytes each.

NOTE: Pseudo DMA transfers across the Multibus are limited to 64 bytes, allowing the ICP to hold the Multibus for a maximum of 100 microseconds.

The ICP then interprets and executes the command block. For example, the command block may order the ICP to write a block of characters from memory to terminal 5. Assuming it is interpreted correctly, the ICP acknowledges the CPU by posting a command accepted status in system memory and interrupting (796'INT/) the CPU.

The ICP again uses the Multibus to access system memory. It retrieves the data from memory (the command block contains pointers) and transfers it to ICP local memory. Then the ICP begins to output the characters. Finally, after outputting all the characters, the ICP accesses the Multibus, transfers a command complete status into system memory and interrupts the CPU to tell it that the new status (command complete) is there.

This arrangement works both ways: the CPU can also order the ICP to write characters from a terminal to system memory. In this case, the ICP loads its local memory with characters from the terminal and transfers them to system memory over the Multibus.

NOTE: The entire block of characters has been output with only two interrupts to the CPU instead of the multiple interrupts which would have been required if the CPU had handled the transfer directly.

14.1. Timing Considerations

At 9600 baud, a character transfer from local memory to the terminal takes 1 milli~~micro~~second, most of which is spent printing the character at the terminal. This would require that the Z8002 process 1000 character interrupts per second to keep up with one terminal printing at full speed. Actually the Z8002 can process interrupts (including typical handling) at the rate of about 2000 per second. Without help, the Z8002 could not support 8 terminals running at full speed.

The Z8002 is assisted by the AMD9517 chips, which do for the Z8002 what the ICP does for the CPU: handle the interrupt received from each character output without interrupting the Z8002 until the entire message has been completed. There are nine DMA channels, one for each terminal. Instead of handling all the interrupts for each character, the Z8002 need only handle 1 to 5% as many interrupts, depending upon the number of characters per message.

14.2. Bus Arbitration

The ICP competes for the bus through a parallel priority encode (as opposed to a daisy chain). While the hardware places no restrictions on the duration of Multibus accesses, software limits each access to 100 microseconds.

The ICP uses pseudo-DMA to transfer data from its local memory to and from system memory, using the block move capability of the Z8002. While not true DMA (the Z8002 gets involved), it is better than conventional memory-to-memory moving because it does not require a software loop, and it requires only one instruction execution by the Z8002. Presently, it takes 13 Z8002 cycles to transfer one word, which gives an aggregate transfer rate of 308K-words per second. This translates to 30.8 words (approximatly 64 bytes) per 100 microseconds.

To keep all terminals running at full speed requires data transfers across the Multibus to the ICP 1000 times a second. To accomplish this, the ICP must send data, then release the bus every millisecond. Because the P/25 is arranged so that no controller occupies the Multibus more than 10% of the time, this allows 100 microseconds per transfer, which limits transfers to 64 bytes. This is arranged in P/25 software.

To accomplish a transfer, the ICP must:

(a) Program CTC2 (U99) channel 0 to request the bus.

(b) Do an MRES instruction to activate the MO/ line, which enables the bus request onto the physical bus.

(c) When the bus is granted, send a nonvectored interrupt to the Z8002. The interrupt handler should respond by transferring the data (maximum of 64 bytes).

(d) When the data has been transferred, issue an MSET instruction to clear the bus request.

This sequence of steps should be repeated, if necessary, for each 64-byte block, until the message has been completely transferred.

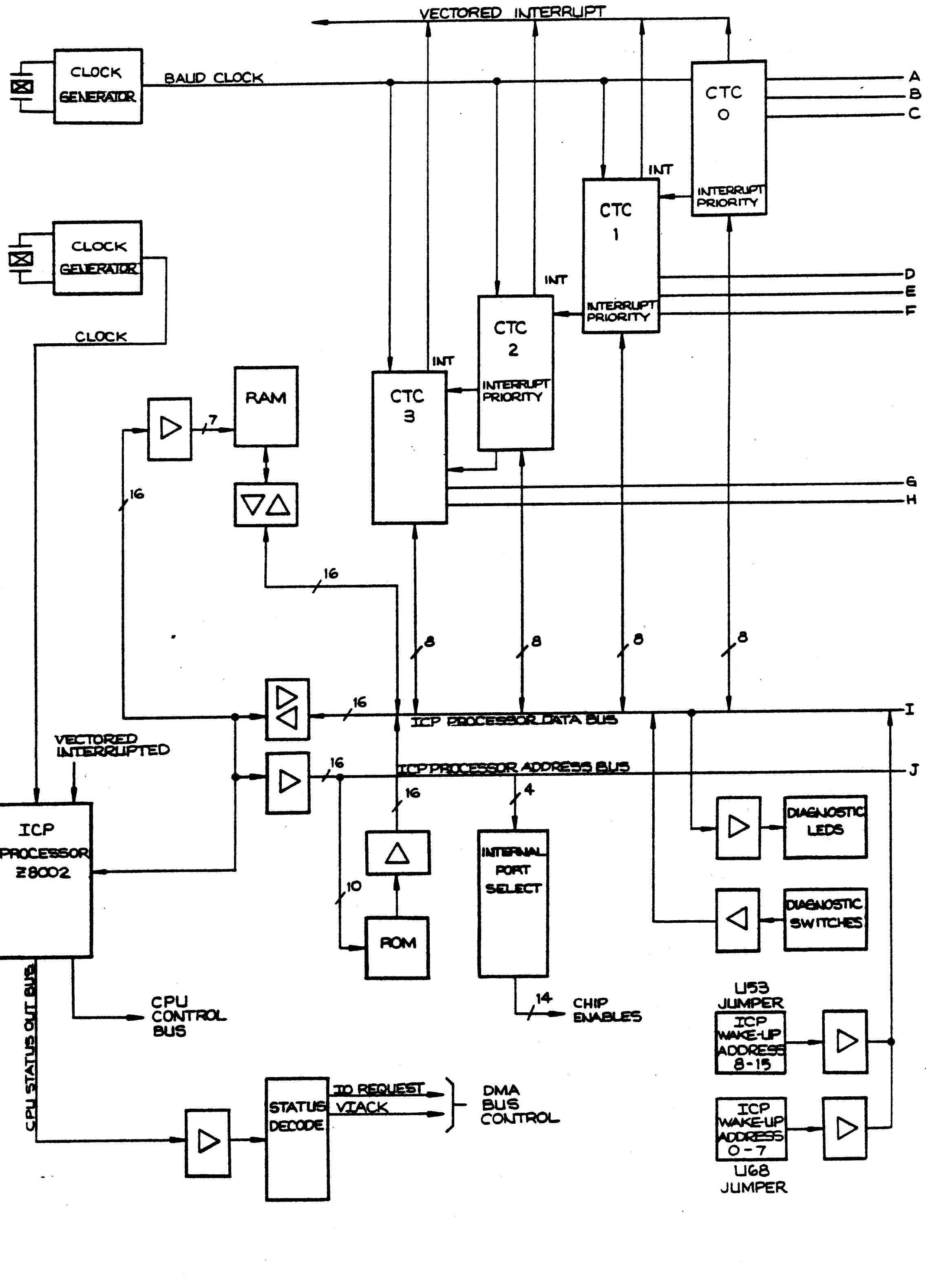


Figure 1. ICP Block Diagram -- Part A

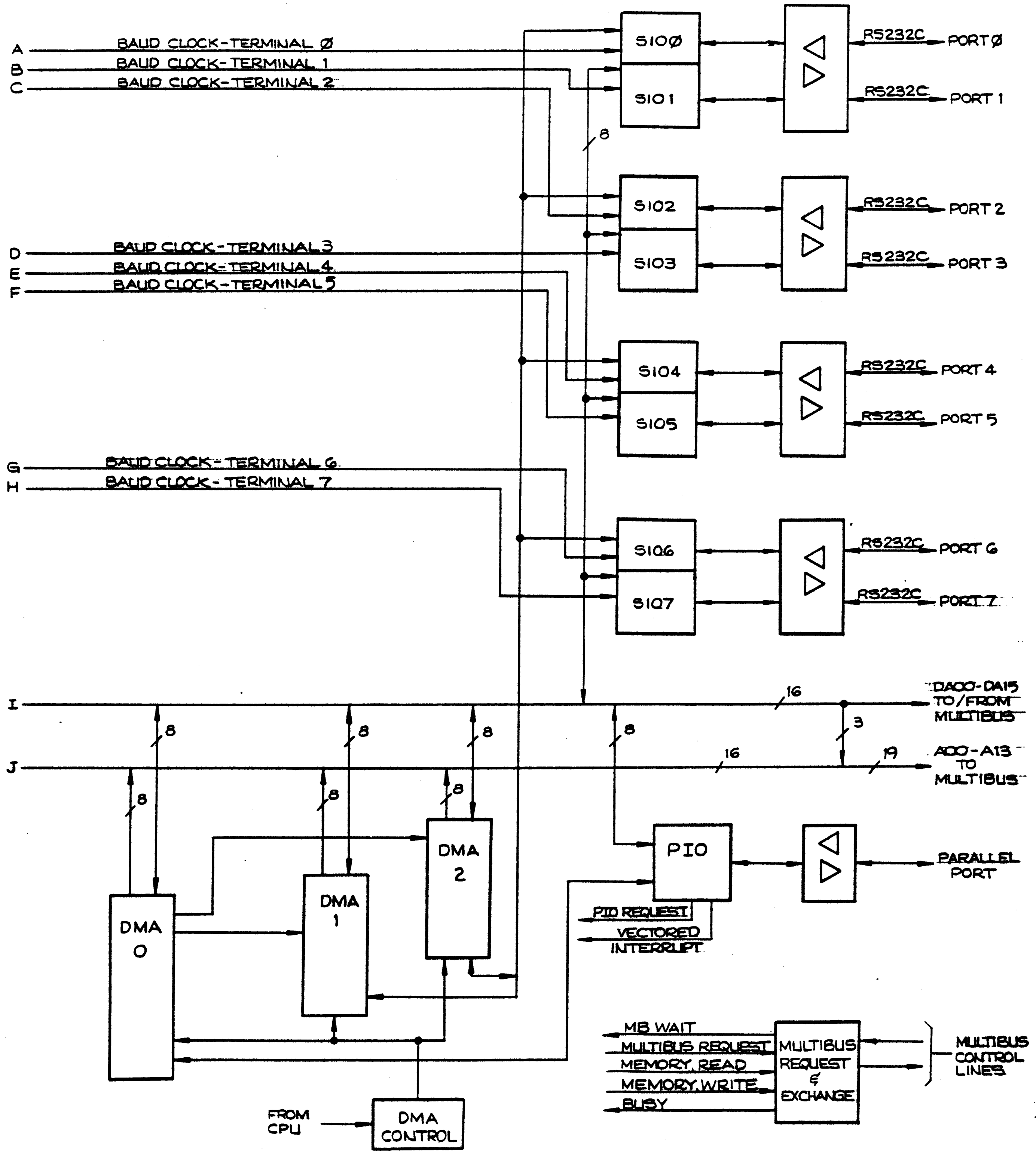


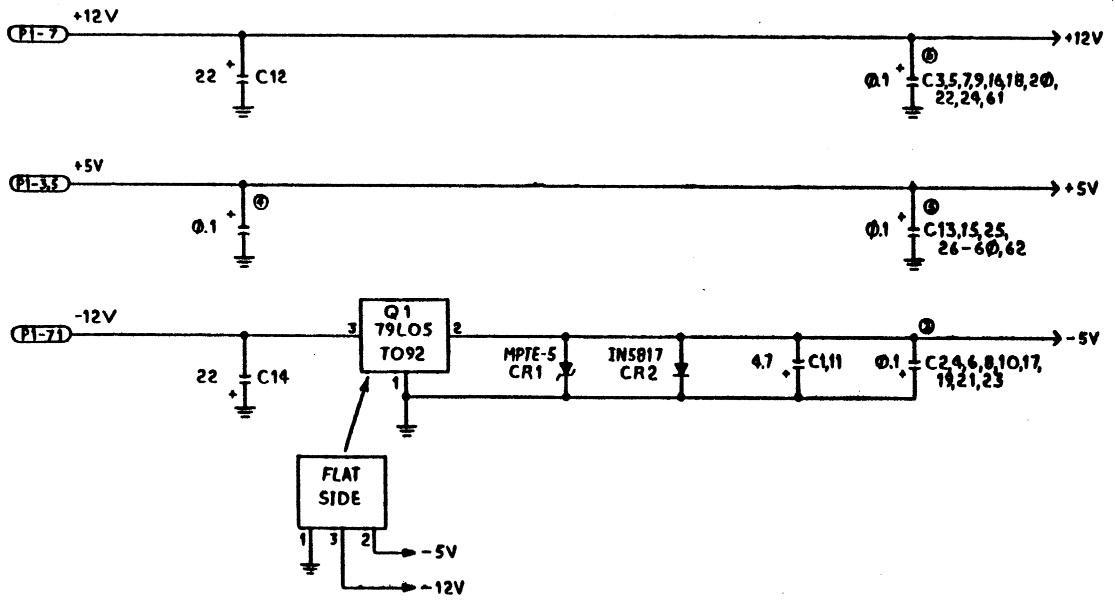
Figure 1. ICP Block Diagram -- Part B

REF. DES.	COMP I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. ID	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED
U1	LS139	3		U29	4116	4		U56	LS32	5-7-13		U83	SØ8	4-7-14		U111	1.228MHz	6		U136	75188	1Ø	
U2	LS74	13		U3Ø	4116	5		U57	LS373	2		U84	LS24Ø	8-13	4		OSC.			U137	75188	1Ø	
U3	LSØ4	3-4-8-7	1	U31	8287	12		U58	LS373	2		U85	LS373	8		U112	S1Ø9	6-13		U138	75188	1Ø	
U4	LS11	13-15-7		U32	8287	12		U59	LS245	4		U86	LS74	5-8		U113	LS244	11		U139	75189	1Ø	
U5	4116	4		U33	LSØØ	3-4-8		U6Ø	LS373	4		U87	74154	7		U114	LS244	11		U14Ø	LSØØ	3-7-15	
U6	4116	4		U34	S74	13		U61	S74	7		U88	2732	14		U115	S157	9		U141	74365	15	
U7	4116	4		U35	LS161	8		U62	LS27	4-15		U89	2732	14		U116	75189	11		U142	LS165	15	
U8	4116	4		U36	LS1Ø	4-7-8		U63	LS244	8		U9Ø	2732	14		U117	75189	1Ø					
U9	4116	4		U37	LS11	13		U64	LS24Ø	12		U91	2732	14		U118	75189	1Ø		RP1	PULL UP	2	
U1Ø	4116	4		U38	LS28Ø	5		U65	LS24Ø	12		U92	74154	7		U119	75189	1Ø			SIP.4.7K		
U11	4116	4		U39	LSØ2	8-13	1	U66	LS245	14		U93	LS373	7		U12Ø	75189	1Ø		RP2	PULL UP	14-15	
U12	4116	4		U4Ø	S157	2		U67	LS373	7		U94	LSØ4	9-15-13	2	U121	75189	1Ø			SIP.4.7K		
U13	4116	4		U41	LS245	2		U68	OCTAL	7		U95	LS373	15		U122	LSØ8	9		RP3	PULL UP	14-15	
U14	LS85	8		U42	Z8ØØ2	2			SW. PAK.			U96	LS373	15		U123	LS1Ø	15			SIP.4.7K		
U15	LS85	8		U43	LS245	2		U69	LSØ4	2-4-5-8		U97	CTC Ø	6		U124	LSØ4	2-15		RP4	PULL UP	8-15	1
U16	LS24Ø	12		U44	S157	2		U7Ø	LS366	2		U98	CTC 1	6		U125	LS11	3-9-7			SIP.470K		
U17	JMP. PAK.	8		U45	LS245	4		U71	74365	2		U99	CTC 2	6		U126	SØ4	3-4-13	1	RP5	PULL UP	2	1
U18	LS74	3-7		U46	LS86	3-5		U72	LS138	14		U1ØØ	CTC3	6		U127	LED PAK	8			SIP.4.7K		
U19	LS32	4-7	1	U47	LS28Ø	5		U73	LS2Ø	4-7		U1Ø1	PIO	9		U128	4 MHz	3		RP6	PULL UP	7	1
U2Ø	LS27	3-15	1	U48	LS373	12		U74	LSØØ	2-8-13		U1Ø2	SIO3/2	9			OSC.				SIP.4.7K		
U21	LS165	13		U49	LSØ4	4-15		U75	LSØ4	4-8-12		U1Ø3	SIO2/2	9		U129	75188	1Ø-11		RP7	PULL UP	7	1
U22	4116	5		U5Ø	LS2Ø	12		U76	S74	2-9		U1Ø4	SIOØ/2	9		U13Ø	PULLUP	11			SIP.4.7K		
U23	4116	4		U51	OCTAL	8		U77	LSØ8	15		U1Ø5	SIO1/2	9			PAK			RP8	PULL UP	8-15	2
U24	4116	4			SW. PAK			U78	LS32	5-7		U1Ø6	DMA Ø	15		U131	75189	1Ø-11			SIP.4.7K		
U25	4116	4		U52	LS373	8		U79	LS245	14		U1Ø7	DMA 1	15		U132	75189	1Ø					
U26	4116	4		U53	JMP. PAK.	7		U8Ø	LS373	15		U1Ø8	DMA2	15		U133	75188	11					
U27	4116	4		U54	LS373	7		U81	LS24Ø	15	6	U1Ø9	S Ø8	5-7-9-13		U134	75188	1Ø-11					
U28	4116	4		U55	LS21	5-9		U82	LS1Ø9	3		U11Ø	LS374	8		U135	75189	1Ø					

REV.	DESCRIPTION	DATE	BY
B	ENG. PRODUCTION RELEASE.	[illegible]	WAM
•	U88-91 WERE 2516.	•	•
C	REVISED TYPE OF IC'S: U[illegible] EN [illegible]	1-7-81	WAM
C	[illegible] TOGETHER EN [illegible]	4-1-82	WAM
D	CONNECT UNUSED INPUTS TO +5V EN 104	4-1-82	WAM
E	U142 WAS CONNECTED TO CLOCK EN [illegible]	4-1-82	WAM
F	EN 128 CENTRONIX PRINTER CHGS	[illegible]	WAM
F	EN 140 NO CIRCUIT CHGS	[illegible]	WAM
G	EN 154 CTC CHGS SHEETS 6,7,9	6-10-[illegible]	WAM

NOTES:
1. ALL CAPACITORS ARE IN MICROFARADS.
2. ALL RESISTOR VALUES ARE IN OHMS, ±5%, ¼ WATT.
3. BYPASS EVERY OTHER DRAM CHIP.
4. BYPASS EVERY EIGHT CHIPS.
5. BYPASS EVERY FOUR CHIPS.
6. BYPASS EVERY OTHER DRAM CHIP.

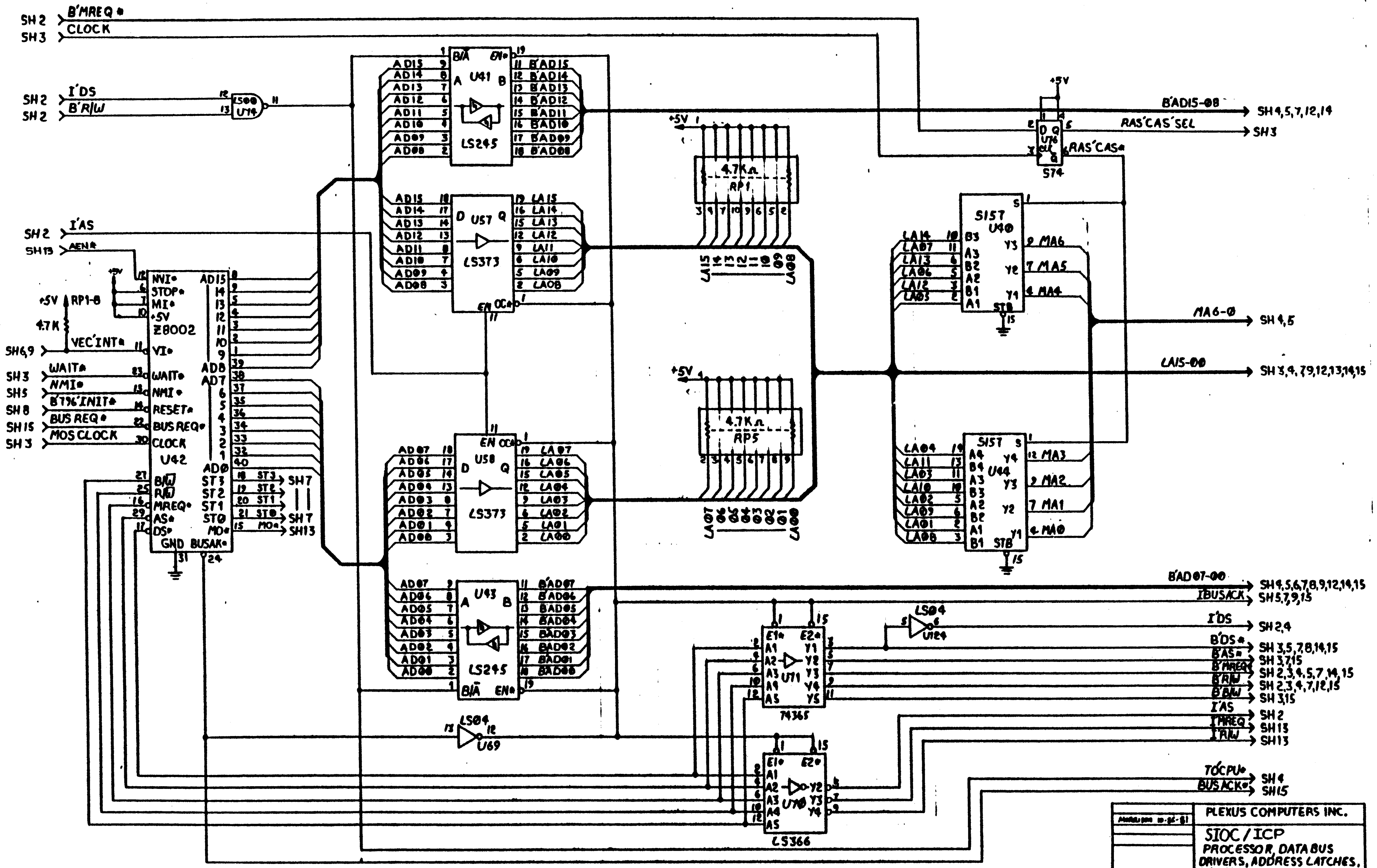
POWER SUPPLY



THIS REV. 'G' SCHEMATIC FOR USE WITH REV. 'L' BOARD ASSEMBLIES. (EN 154)

CONFIRM WITH PLEXUS DOCUMENT CONTROL DEPT. CURRENT REVISION LEVELS OF ASS'Y DRAWING, P.C. ARTWORK AND SOFTWARE.

[illegible] 4-81	PLEXUS COMPUTERS INC.
MORRISON [illegible]-81	PWB · SCHEMATIC · ICP
SIZE	6Ø·ØØØ79
SCALE —	SHT 1 OF 15
REV	G

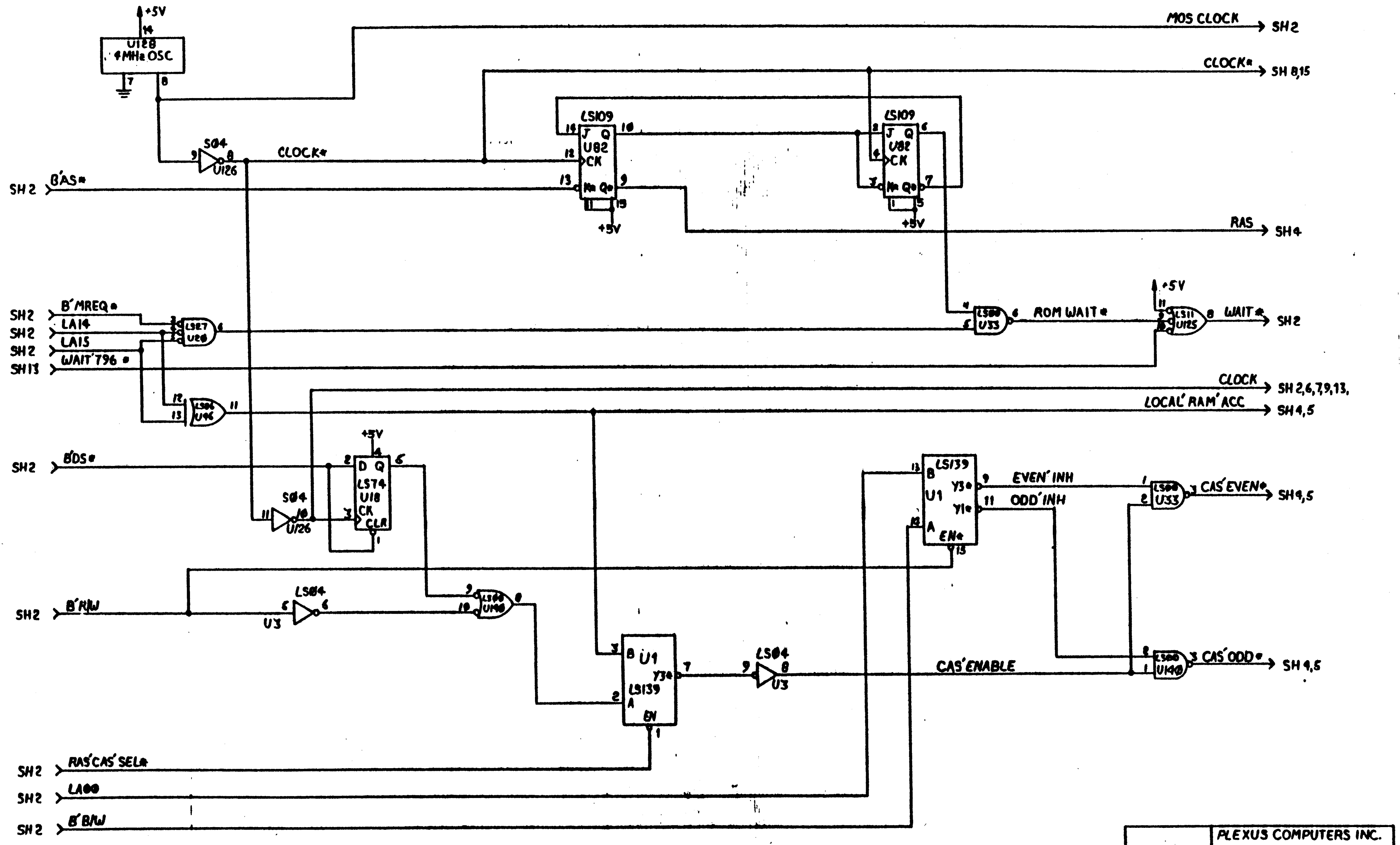

SH 2 B'MREQ*
SH 3 CLOCK
SH 2 I'DS
SH 2 B'R/W
LS00 U14
SH 2 I'AS
SH 13 AEN*
+5V RP1-8
4.7K
SH 6,9 VEC'INT*
SH 3 WAIT*
SH 5 NMI*
SH 8 BT%'INIT*
SH 15 BUSREQ*
SH 3 MOS CLOCK
Z8002
U42
NVI* STOP* MI* +5V VI* WAIT* NMI* RESET* BUSREQ* CLOCK B/W R/W MREQ* AS* DS* GND BUSAK*
AD15 AD8 AD7 AD0 ST3 ST2 ST1 ST0 MO*
ST3 ST2 SH7 ST1 ST0 SH7 MO* SH13
U41 LS245 B/A EN* A B
AD15 AD14 AD13 AD12 AD11 AD10 AD09 AD08
B'AD15 B'AD14 B'AD13 B'AD12 B'AD11 B'AD10 B'AD09 B'AD08
U57 LS373 D Q EN OC*
LA15 LA14 LA13 LA12 LA11 LA10 LA09 LA08
U58 LS373 D Q EN OC*
AD07 AD06 AD05 AD04 AD03 AD02 AD01 AD00
LA07 LA06 LA05 LA04 LA03 LA02 LA01 LA00
U43 LS245 A B B/A EN*
B'AD07 B'AD06 B'AD05 B'AD04 B'AD03 B'AD02 B'AD01 B'AD00
+5V 4.7KΩ RP1
+5V 4.7KΩ RP5
LA15 14 13 12 11 10 09 LA08
LA07 06 05 04 03 02 01 LA00
+5V D Q U76 Q̄ S74
B'AD15-08 SH 4,5,7,12,14
RAS'CAS'SEL SH 3
RAS'CAS*
S157 U40 S A4 B3 A3 B2 A2 B1 A1 STB Y3 Y2 Y1
LA14 LA07 LA13 LA06 LA12 LA05
MA6 MA5 MA4
MA6-0 SH 4,5
LA15-00 SH 3,4,7,9,12,13,14,15
S157 U44 A4 B4 A3 B3 A2 B2 A1 B1 STB Y4 Y3 Y2 Y1
LA04 LA11 LA03 LA10 LA02 LA09 LA01 LA08
MA3 MA2 MA1 MA0
B'AD07-00 SH 4,5,6,7,8,9,12,14,15
IBUSACK SH 5,7,9,15
LS04 U124
I'DS SH 2,4
B'DS* SH 3,5,7,8,14,15
B'AS* SH 3,7,15
B'MREQ SH 2,3,4,5,7,14,15
B'R/W SH 2,3,4,7,12,15
B'B/W SH 3,15
74365 U71 E1* E2* A1 A2 A3 A4 A5 Y1 Y2 Y3 Y4 Y5
I'AS SH 2
IMREQ SH 13
IR/W SH 13
LS04 U69
LS366 U70 E1* E2* A1 A2 A3 A4 A5 Y2 Y3 Y4
TOCPU* SH 4
BUSACK* SH 15

PLEXUS COMPUTERS INC.

SIOC / ICP

PROCESSOR, DATA BUS DRIVERS, ADDRESS LATCHES, AND MUX.

SIZE	SCALE: —	60-00079 SHEET [illegible] OF 15	REV G

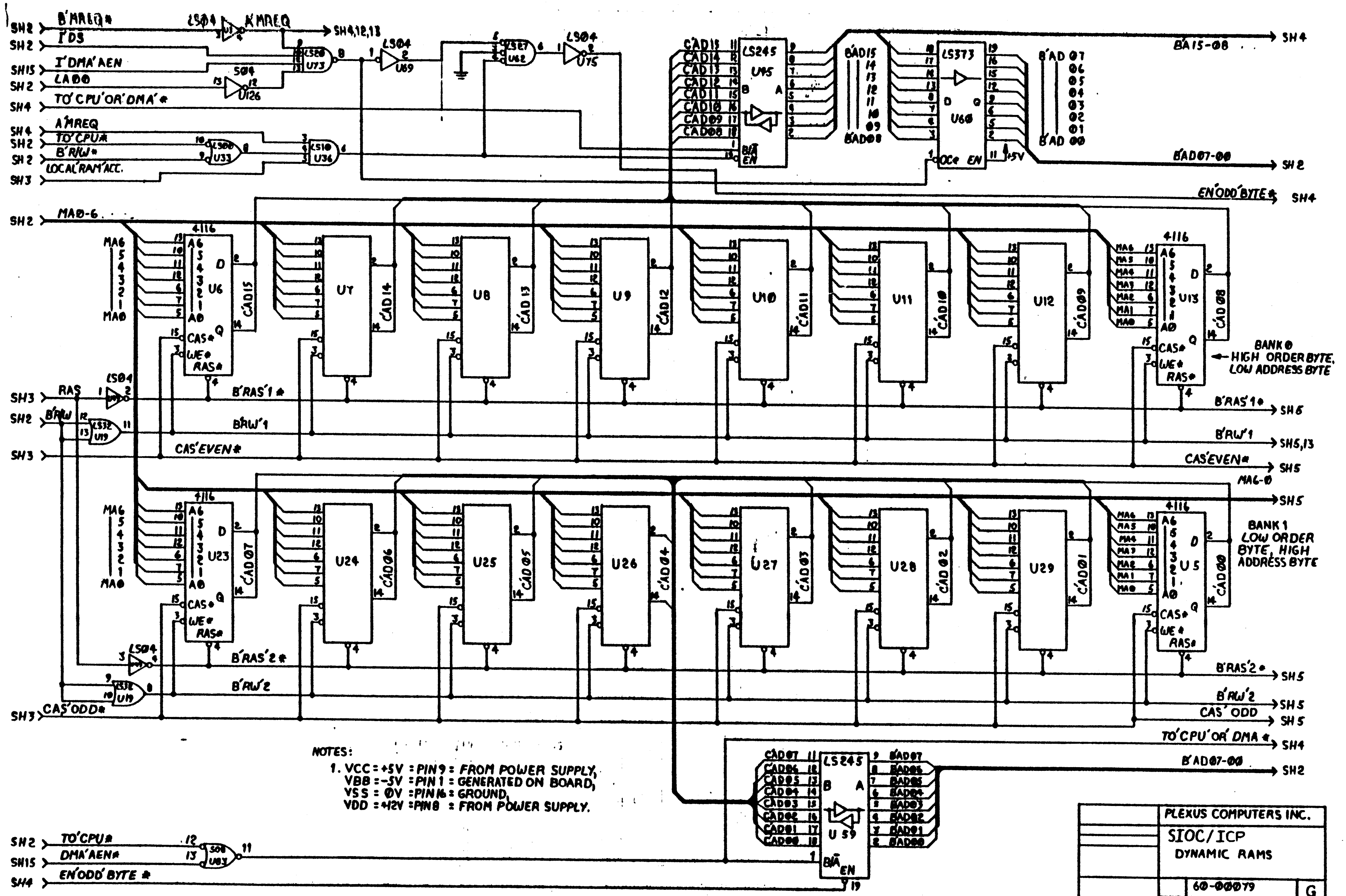


PLEXUS COMPUTERS INC.

SIOC/ICP
RAS,CAS, GENERATOR,
ROM WAIT GENERATOR

60-00079 | G.

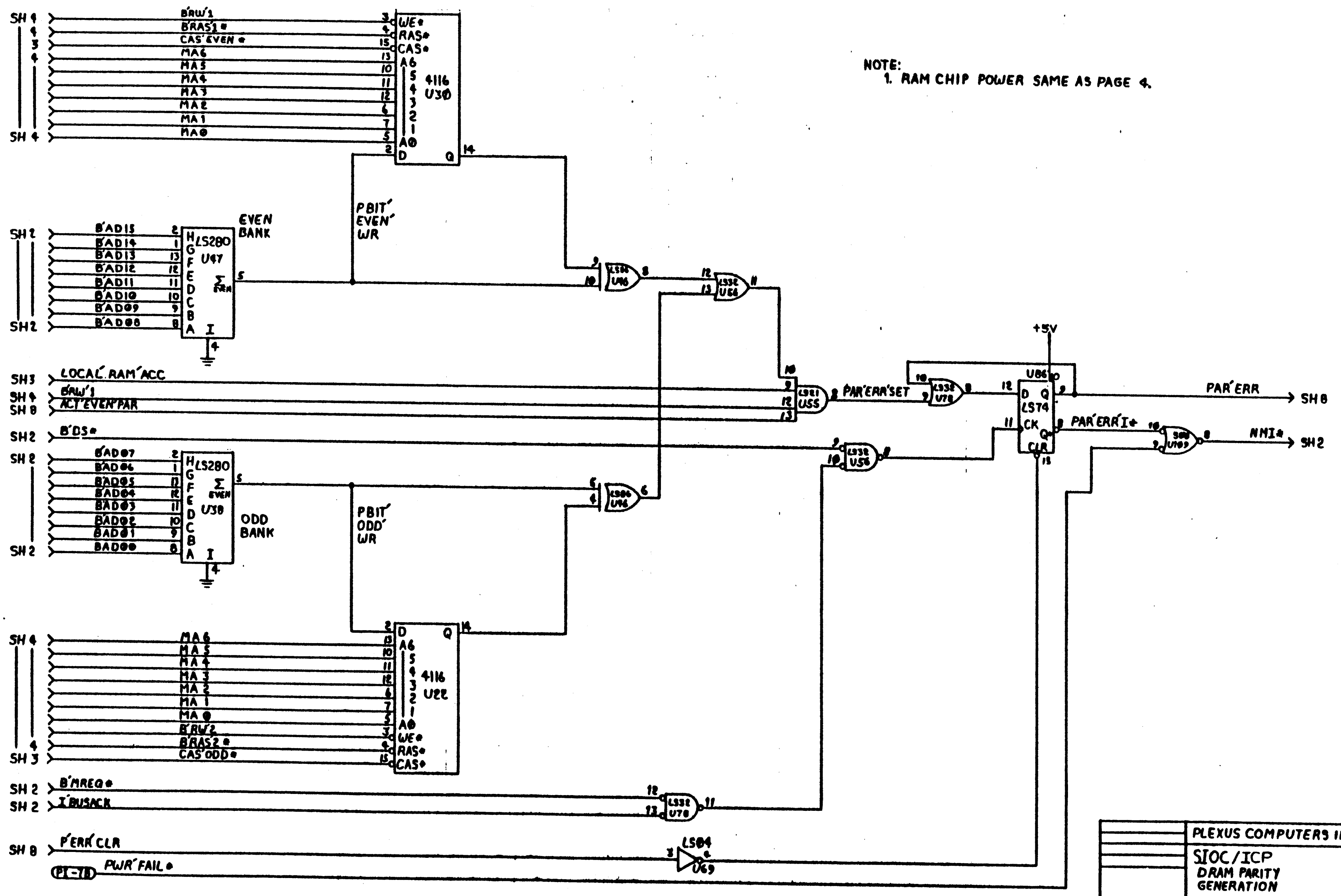
SIZE | SCALE: – | SH 3 OF 19 | REV.



NOTES:

1. VCC = +5V : PIN 9 : FROM POWER SUPPLY,
   VBB = -5V : PIN 1 : GENERATED ON BOARD,
   VSS = 0V : PIN 16 : GROUND,
   VDD = +12V : PIN 8 : FROM POWER SUPPLY.

PLEXUS COMPUTERS INC.	
SIOC/ICP	
DYNAMIC RAMS	
60-00079	G
SIZE SCALE: — SH 4 OF 15	REV.



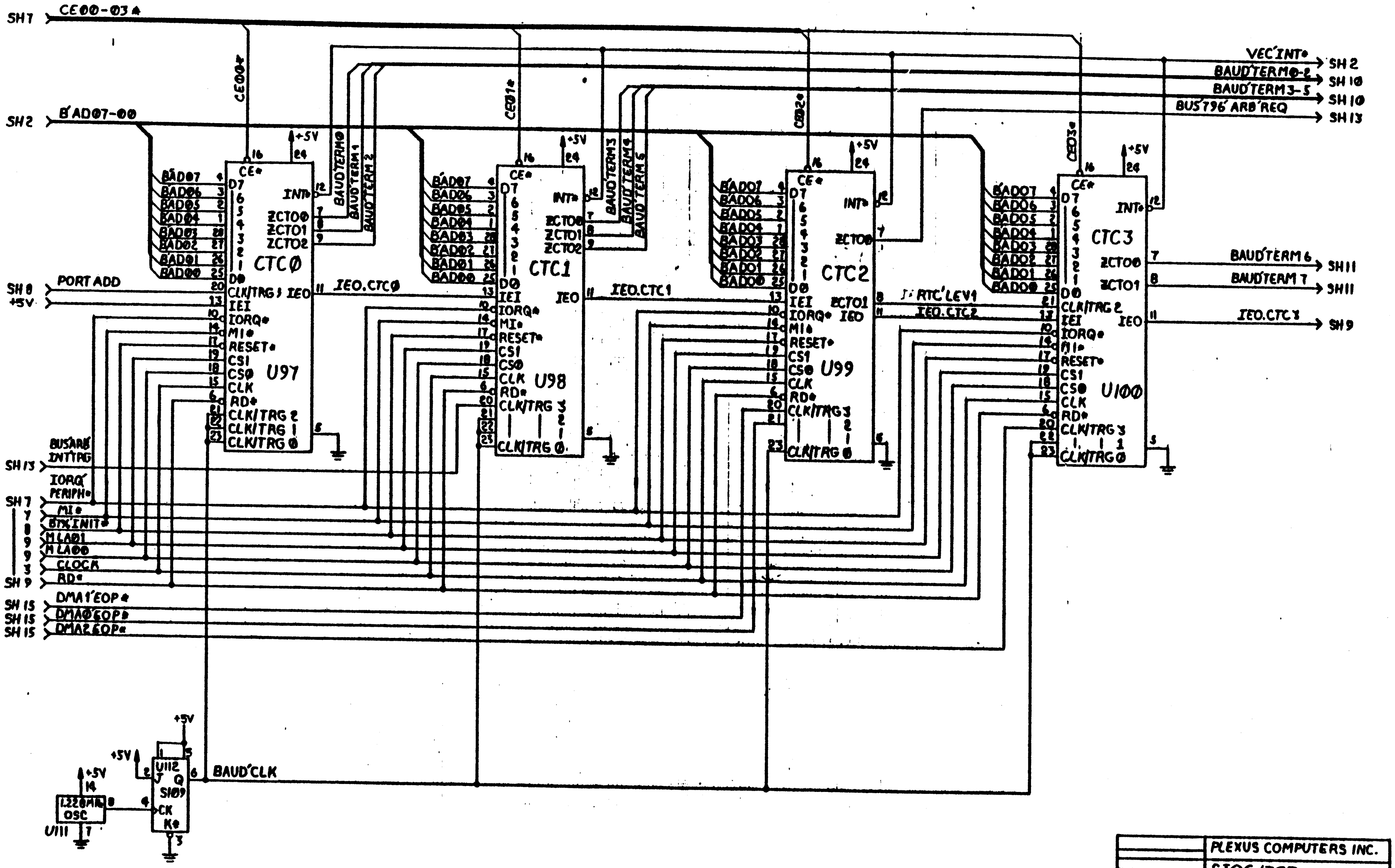
NOTE:
1. RAM CHIP POWER SAME AS PAGE 4.

PLEXUS COMPUTERS INC.

SIOC/ICP
DRAM PARITY
GENERATION

60-00079 G

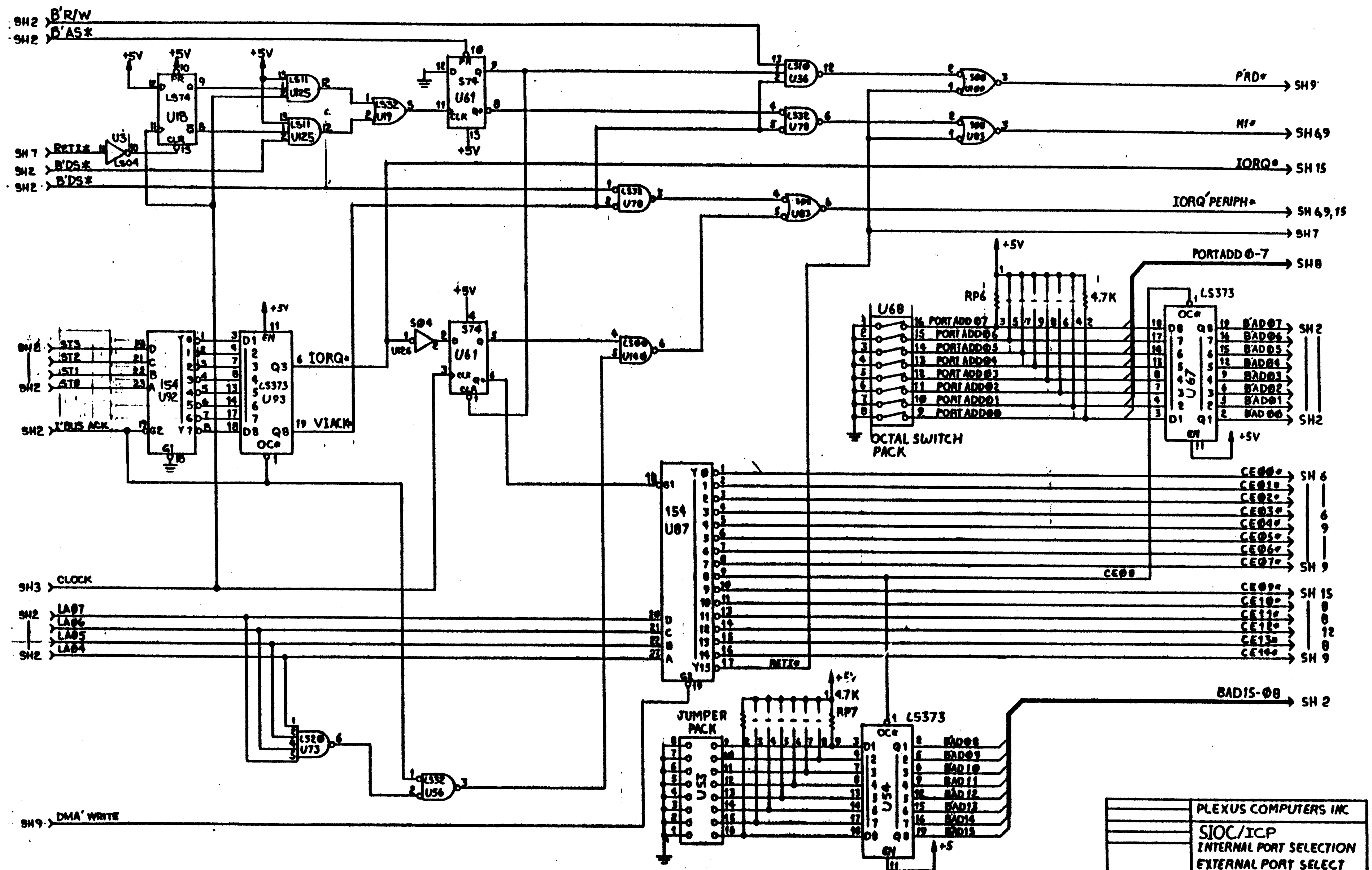
SIZE SCALE — SH 3 OF 13 REV



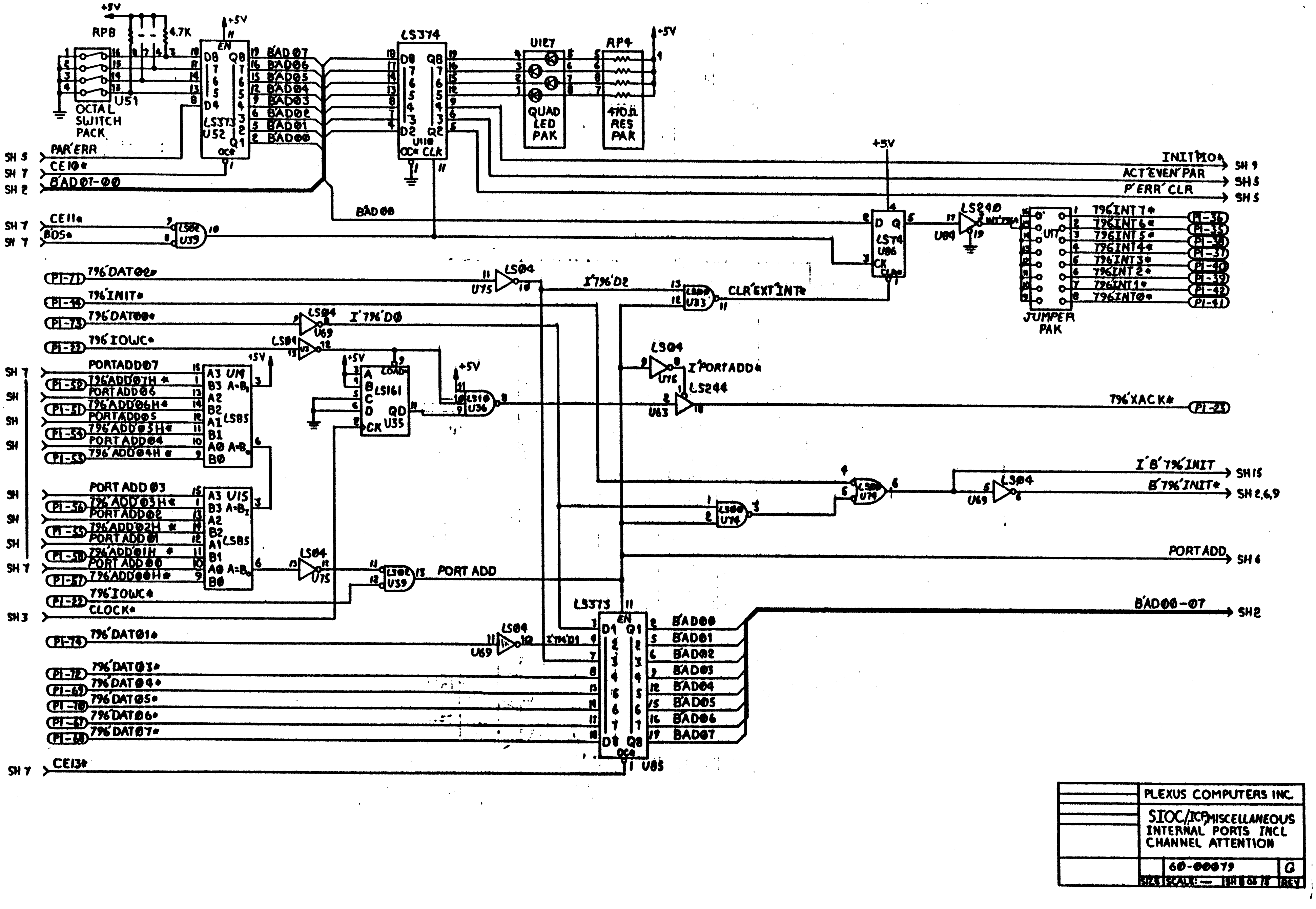
PLEXUS COMPUTERS INC.

SIOC/ICP
REAL TIME CLOCK
CONTROLLER/TIMER CRT'S

	60-00079		G
SIZE	SCALE: —	SH 6 of 15	REV



PLEXUS COMPUTERS INC
SIOC/ICP
INTERNAL PORT SELECTION
EXTERNAL PORT SELECT
6Ø-ØØØ79 — G
SIZE — SCALE — SHT — REV

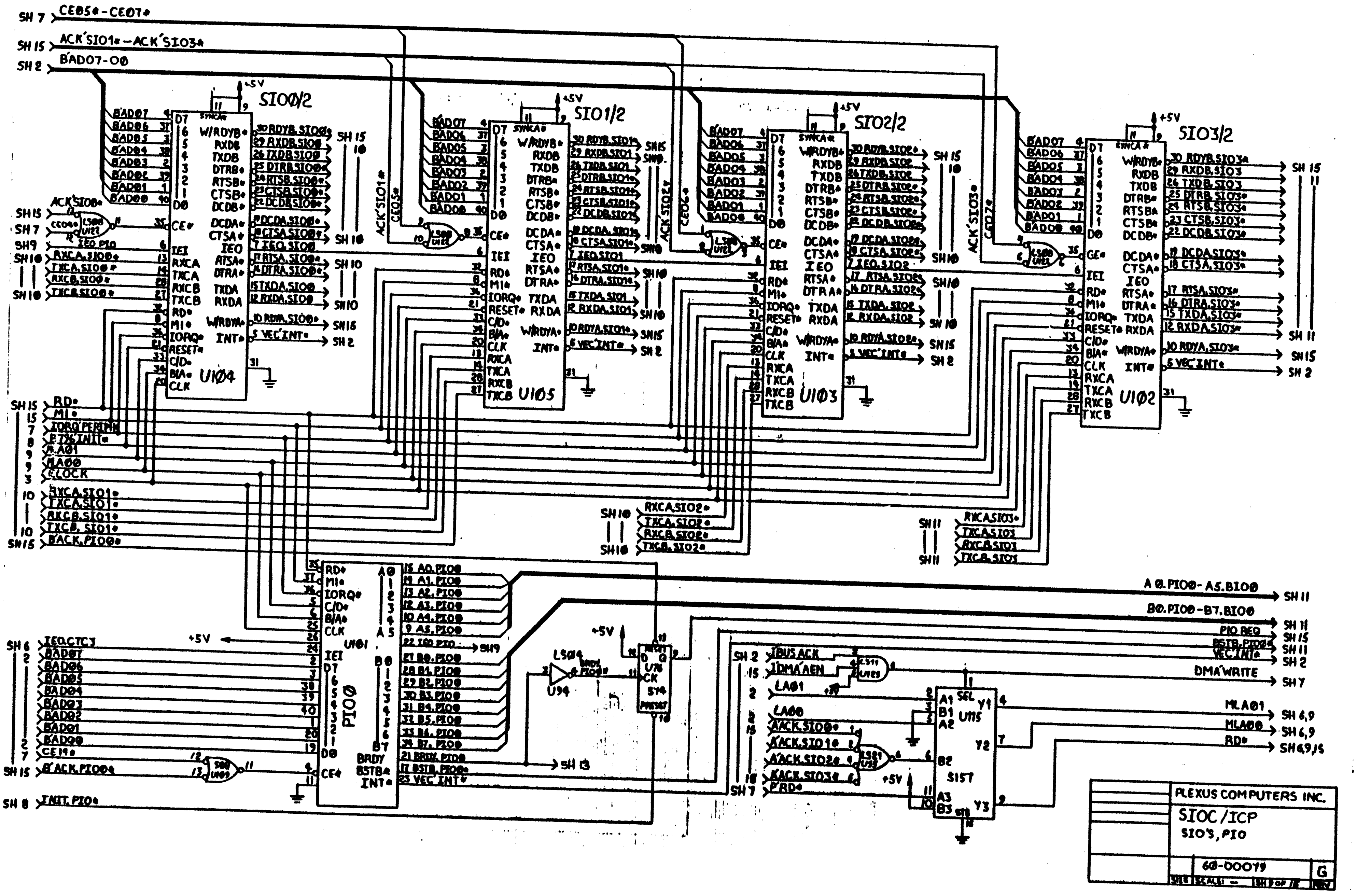

RP8
OCTAL SWITCH PACK
4.7K
LS373 U52
LS374 U110
U127 QUAD LED PAK
RP4 470Ω RES PAK
BAD07-00
PAR'ERR
CE10*
CE11*
BDS*
INIT'IO*
ACT'EVEN'PAR
P'ERR'CLR
LS74 U86
LS240 U84
JUMPER PAK
796'INT7* 796'INT6* 796'INT5* 796'INT4* 796'INT3* 796'INT2* 796'INT1* 796'INT0*
796'DAT02*
796'INIT*
796'DAT00*
796'IOWC*
I'796'D2
I'796'D0
CLR'EXT'INT*
I'PORTADD*
LS161 U35
796'XACK*
LS85 U14
LS85 U15
PORTADD07 796'ADD07H* PORTADD06 796'ADD06H* PORTADD05 796'ADD05H* PORTADD04 796'ADD04H*
PORTADD03 796'ADD03H* PORTADD02 796'ADD02H* PORTADD01 796'ADD01H* PORTADD00 796'ADD00H*
I'B'796'INIT
B'796'INIT*
PORT ADD
796'IOWC*
CLOCK*
796'DAT01*
796'DAT03* 796'DAT04* 796'DAT05* 796'DAT06* 796'DAT07*
LS373 U85
BAD00 BAD01 BAD02 BAD03 BAD04 BAD05 BAD06 BAD07
B'AD00-07
CE13*
SH 2 SH 3 SH 5 SH 6 SH 7 SH 9 SH 15 SH 2,6,9

PLEXUS COMPUTERS INC.

SIOC/ICP, MISCELLANEOUS INTERNAL PORTS INCL CHANNEL ATTENTION

60-00079 G

SIZE SCALE: — SH 8 OF 15 REV

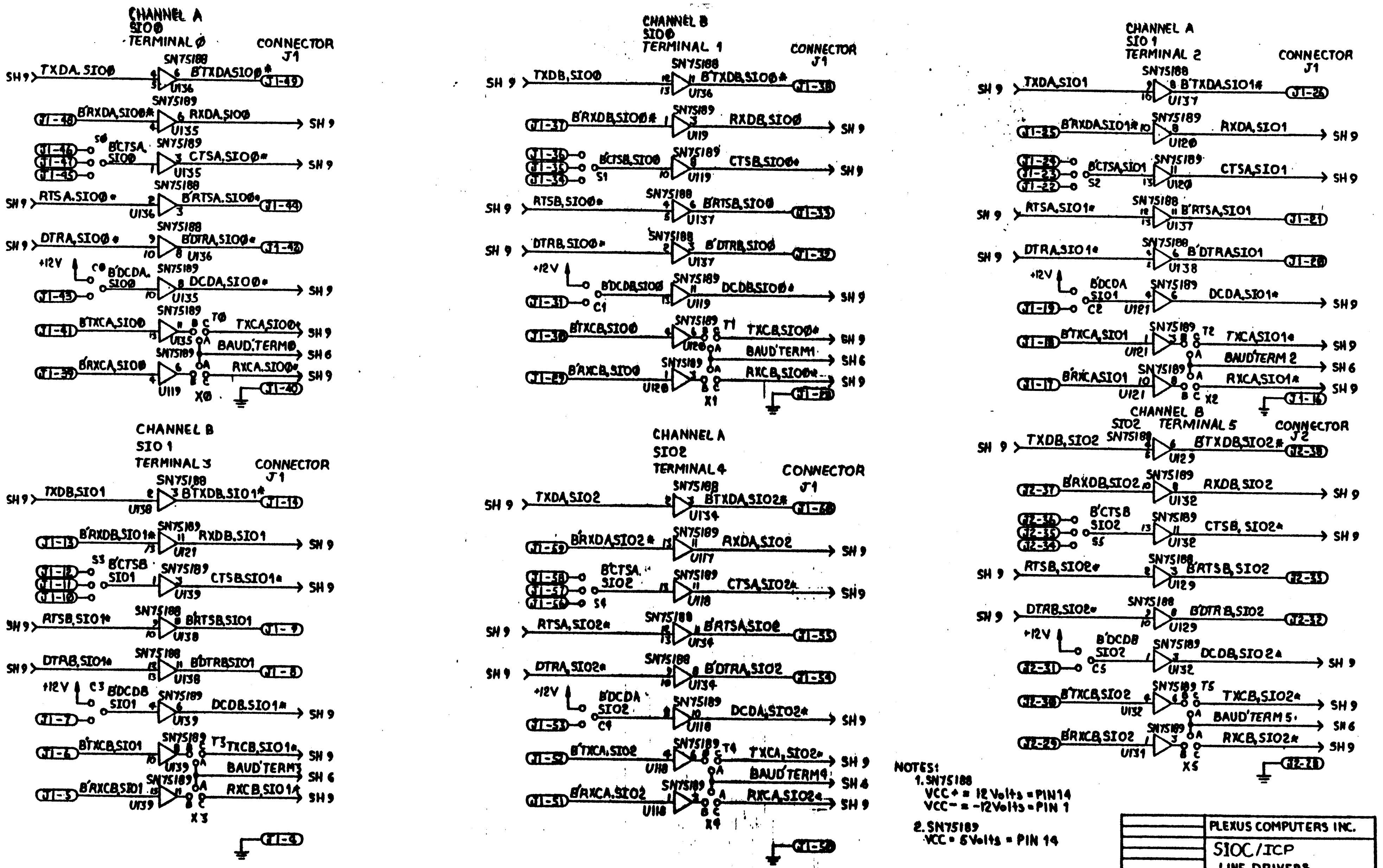

SH 7 CE05*-CE07*
SH 15 ACK'SIO1*-ACK'SIO3*
SH 2 BAD07-00
SIO0/2
SIO1/2
SIO2/2
SIO3/2
U104
U105
U103
U102
PIO0
U101
+5V
D7 6 5 4 3 2 1 D0
CE* IEI RXCA TXCA RXCB TXCB RD* M1* IORQ* RESET* C/D* B/A* CLK
W/RDYB* RXDB TXDB DTRB* RTSB* CTSB* DCDB* DCDA* CTSA* IEO RTSA* DTRA* TXDA RXDA W/RDYA* INT*
SYNCA*
30 RDYB.SIO0* 29 RXDB.SIO0 26 TXDB.SIO0 25 DTRB.SIO0* 24 RTSB.SIO0* 23 CTSB.SIO0* 22 DCDB.SIO0*
19 DCDA.SIO0* 18 CTSA.SIO0* 7 IEO.SIO0 17 RTSA.SIO0* 16 DTRA.SIO0* 15 TXDA.SIO0 12 RXDA.SIO0
10 RDYA.SIO0* 5 VEC'INT*
SH 15 SH 10 SH 2
ACK'SIO0* CE04* IEO.PIO LS00 U122
SH 15 SH 7 SH 9 SH 10
RXCA.SIO0* TXCA.SIO0* RXCB.SIO0* TXCB.SIO0*
ACK'SIO1* CE05* ACK'SIO2* CE06* ACK'SIO3* CE07*
SH 15 RD* 15 M1* 7 IORQ'PERIPH* 8 P.796 INIT* 9 A01 9 MLA00 3 CLOCK
10 RXCA.SIO1* TXCA.SIO1* RXCB.SIO1* 10 TXCB.SIO1* SH 15 BACK.PIO0*
SH 10 RXCA.SIO2* TXCA.SIO2* RXCB.SIO2* TXCB.SIO2*
SH 11 RXCA.SIO3* TXCA.SIO3 RXCB.SIO3 TXCB.SIO3
A0 1 2 3 4 A5 B0 1 2 3 4 5 6 B7 BRDY BSTB* INT*
15 A0.PIO0 14 A1.PIO0 13 A2.PIO0 12 A3.PIO0 10 A4.PIO0 9 A5.PIO0 22 IEO PIO
27 B0.PIO0 28 B1.PIO0 29 B2.PIO0 30 B3.PIO0 31 B4.PIO0 32 B5.PIO0 33 B6.PIO0 34 B7.PIO0
21 BRDY.PIO0 17 BSTB.PIO0* 23 VEC'INT*
A0.PIO0-A5.BIO0 SH 11
B0.PIO0-B7.BIO0 SH 11
PIO REQ SH 11
BSTB.PIO0* SH 15
VEC'INT* SH 11
SH 2
DMA'WRITE SH 7
SH 6 IEO.CTC3 2 BAD07 BAD06 BAD05 BAD04 BAD03 BAD02 BAD01 BAD00 2 7 CE14* SH 15 B'ACK.PIO0*
SH 8 INIT.PIO*
LS00 U103
LS04 U94 BRDY PIO0*
S74 U76 D Q CK PRESET RESET
SH 2 IBUS ACK 15 IDMA'AEN 2 LA01 LA00 AACK.SIO0* AACK.SIO1* AACK.SIO2* KACK.SIO3* SH 7 PRD*
LS11 U123
LS21 U52
S157 U115 SEL A1 B1 A2 B2 A3 B3 Y1 Y2 Y3
MLA01 SH 6,9
MLA00 SH 6,9
RD* SH 6,9,15
SH 13 SH 9

PLEXUS COMPUTERS INC.

SIOC/ICP
SIO'S, PIO

60-00079 G

SCALE: — SH 9 OF 18 REV

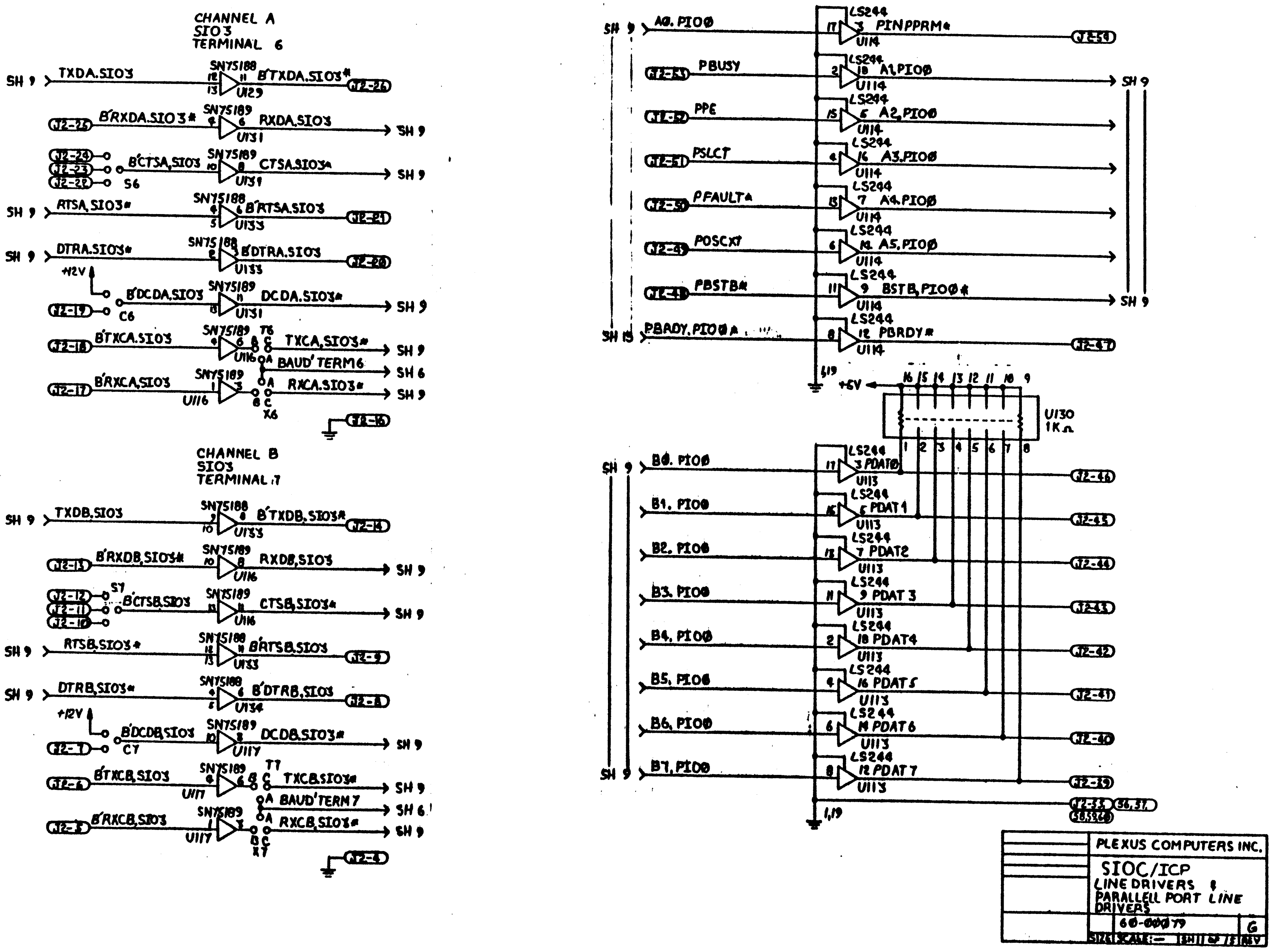


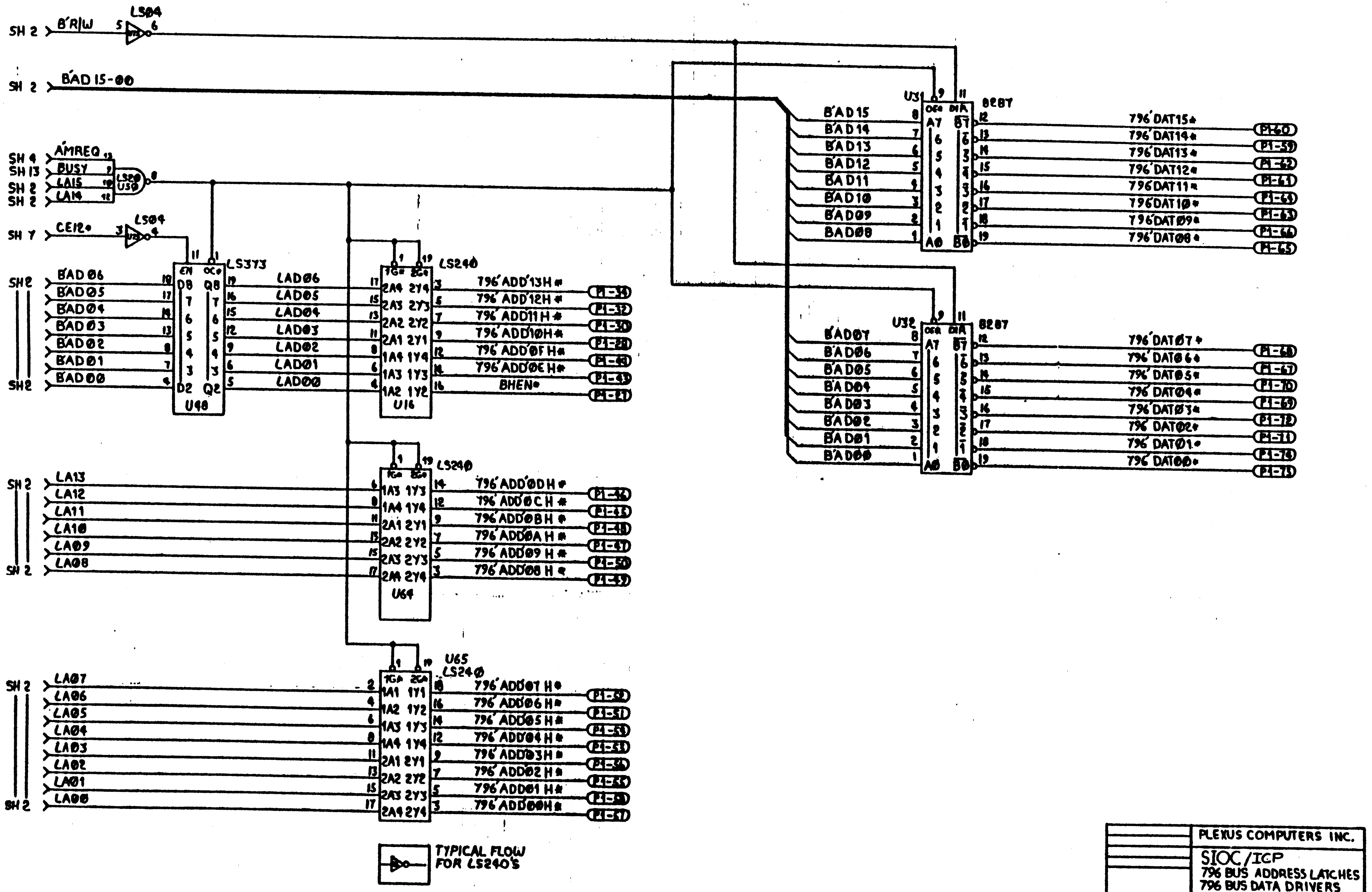
NOTES:
1. SN75188
VCC+ = 12 Volts = PIN 14
VCC− = −12 Volts = PIN 1
2. SN75189
VCC = 5 Volts = PIN 14

PLEXUS COMPUTERS INC.

SIOC/ICP
LINE DRIVERS

60-00079 G


CHANNEL A
SIO3
TERMINAL 6
SH 9 TXDA.SIO3 SN75188 U129 B'TXDA.SIO3* J2-26
J2-25 B'RXDA.SIO3* SN75189 U131 RXDA.SIO3 SH 9
J2-24 J2-23 J2-22 S6 B'CTSA.SIO3 SN75189 U131 CTSA.SIO3* SH 9
SH 9 RTSA.SIO3* SN75188 U133 B'RTSA.SIO3 J2-21
SH 9 DTRA.SIO3* SN75188 U133 B'DTRA.SIO3 J2-20
+12V
J2-19 C6 B'DCDA.SIO3 SN75189 U131 DCDA.SIO3* SH 9
J2-18 B'TXCA.SIO3 SN75189 U116 T6 TXCA.SIO3* SH 9
BAUD' TERM6 SH 6
J2-17 B'RXCA.SIO3 SN75189 U116 X6 RXCA.SIO3* SH 9
J2-16
CHANNEL B
SIO3
TERMINAL 7
SH 9 TXDB.SIO3 SN75188 U133 B'TXDB.SIO3* J2-14
J2-13 B'RXDB.SIO3* SN75189 U116 RXDB.SIO3 SH 9
J2-12 J2-11 J2-10 S7 B'CTSB.SIO3 SN75189 U116 CTSB.SIO3* SH 9
SH 9 RTSB.SIO3* SN75188 U133 B'RTSB.SIO3 J2-9
SH 9 DTRB.SIO3* SN75188 U134 B'DTRB.SIO3 J2-8
+12V
J2-7 C7 B'DCDB.SIO3 SN75189 U117 DCDB.SIO3* SH 9
J2-6 B'TXCB.SIO3 SN75189 U117 T7 TXCB.SIO3* SH 9
BAUD' TERM7 SH 6
J2-5 B'RXCB.SIO3 SN75189 U117 X7 RXCB.SIO3* SH 9
J2-4
SH 9 A0.PIO0 LS244 U114 PINPPRM* J2-54
J2-53 PBUSY LS244 U114 A1.PIO0 SH 9
J2-52 PPE LS244 U114 A2.PIO0
J2-51 PSLCT LS244 U114 A3.PIO0
J2-50 PFAULT* LS244 U114 A4.PIO0
J2-49 POSCXT LS244 U114 A5.PIO0
J2-48 PBSTB* LS244 U114 BSTB.PIO0* SH 9
SH 15 PBRDY.PIO0* LS244 U114 PBRDY* J2-47
1,19
+5V
U130 1KΩ
SH 9 B0.PIO0 LS244 U113 PDAT0 J2-46
B1.PIO0 LS244 U113 PDAT1 J2-45
B2.PIO0 LS244 U113 PDAT2 J2-44
B3.PIO0 LS244 U113 PDAT3 J2-43
B4.PIO0 LS244 U113 PDAT4 J2-42
B5.PIO0 LS244 U113 PDAT5 J2-41
B6.PIO0 LS244 U113 PDAT6 J2-40
SH 9 B7.PIO0 LS244 U113 PDAT7 J2-39
J2-35 36,37 38,39,40
1,19
PLEXUS COMPUTERS INC.
SIOC/ICP
LINE DRIVERS & PARALLELL PORT LINE DRIVERS
60-00079 G
SIZE SCALE:— SH 11 OF 15 REV



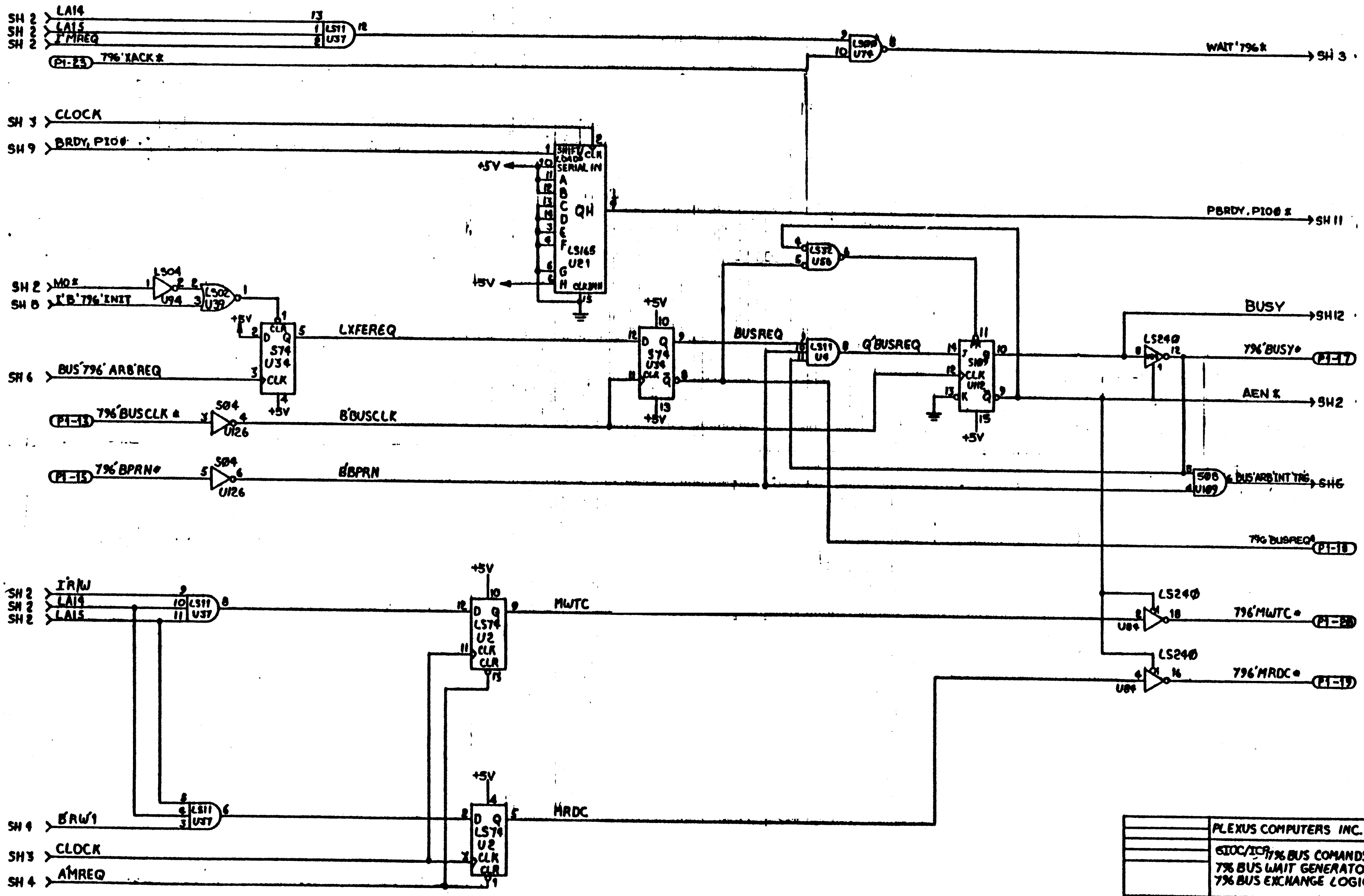
SH 2 B'R/W
LS04
SH 2 BAD 15-00
SH 4 AMREQ
SH 13 BUSY
SH 2 LA15
SH 2 LA14
LS20 U30
SH 7 CE12*
LS04
LS373
U48
BAD06 BAD05 BAD04 BAD03 BAD02 BAD01 BAD00
LAD06 LAD05 LAD04 LAD03 LAD02 LAD01 LAD00
LS240
U16
796'ADD13H* 796'ADD12H* 796'ADD11H* 796'ADD10H* 796'ADD0FH* 796'ADD0EH* BHEN*
LS240
U64
LA13 LA12 LA11 LA10 LA09 LA08
796'ADD0DH* 796'ADD0CH* 796'ADD0BH* 796'ADD0AH* 796'ADD09H* 796'ADD08H*
U65
LS240
LA07 LA06 LA05 LA04 LA03 LA02 LA01 LA00
796'ADD07H* 796'ADD06H* 796'ADD05H* 796'ADD04H* 796'ADD03H* 796'ADD02H* 796'ADD01H* 796'ADD00H*
U31
8287
BAD15 BAD14 BAD13 BAD12 BAD11 BAD10 BAD09 BAD08
796'DAT15* 796'DAT14* 796'DAT13* 796'DAT12* 796'DAT11* 796'DAT10* 796'DAT09* 796'DAT08*
P1-60 P1-59 P1-62 P1-61 P1-64 P1-63 P1-66 P1-65
U32
8287
BAD07 BAD06 BAD05 BAD04 BAD03 BAD02 BAD01 BAD00
796'DAT07* 796'DAT06* 796'DAT05* 796'DAT04* 796'DAT03* 796'DAT02* 796'DAT01* 796'DAT00*
P1-68 P1-67 P1-70 P1-69 P1-72 P1-71 P1-74 P1-73
TYPICAL FLOW FOR LS240'S

PLEXUS COMPUTERS INC.

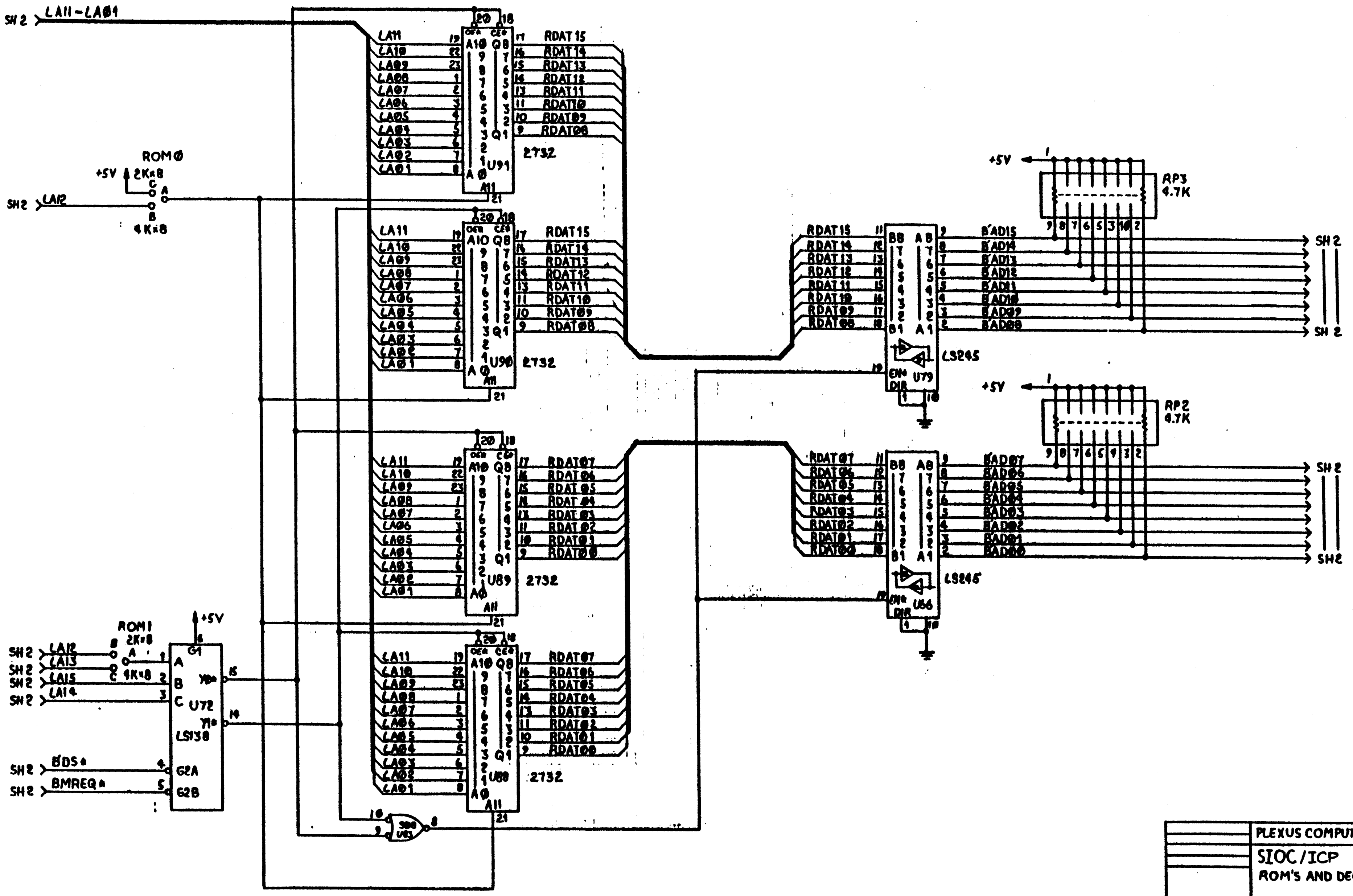
SIOC/ICP
796 BUS ADDRESS LATCHES
796 BUS DATA DRIVERS

60-00079 6

SIZE SCALE — SHEET 6 OF 11 REV



	PLEXUS COMPUTERS INC.
	SIOC/ICP 796 BUS COMANDS, 796 BUS WAIT GENERATOR, 796 BUS EXCHANGE LOGIC
	60-00079 G
	SIZE SCALE SH 13 OF 18 REV


SH2 LA11-LA01
ROM0
+5V 2K×8
4K×8
SH2 LA12
LA11 LA10 LA09 LA08 LA07 LA06 LA05 LA04 LA03 LA02 LA01
A10 A0 A11 Q8 Q1 OE CE
U91 2732
U90 2732
U89 2732
U88 2732
RDAT15 RDAT14 RDAT13 RDAT12 RDAT11 RDAT10 RDAT09 RDAT08
RDAT07 RDAT06 RDAT05 RDAT04 RDAT03 RDAT02 RDAT01 RDAT00
+5V
RP3 4.7K
RP2 4.7K
B8 B1 A8 A1 EN DIR
U79 LS245
U66 LS245
BAD15 BAD14 BAD13 BAD12 BAD11 BAD10 BAD09 BAD08
BAD07 BAD06 BAD05 BAD04 BAD03 BAD02 BAD01 BAD00
SH2
ROM1
2K×8 4K×8
SH2 LA12
SH2 LA13
SH2 LA15
SH2 LA14
SH2 BDS*
SH2 BMREQ*
G1 A B C G2A G2B Y0 Y1
U72 LS138
U83

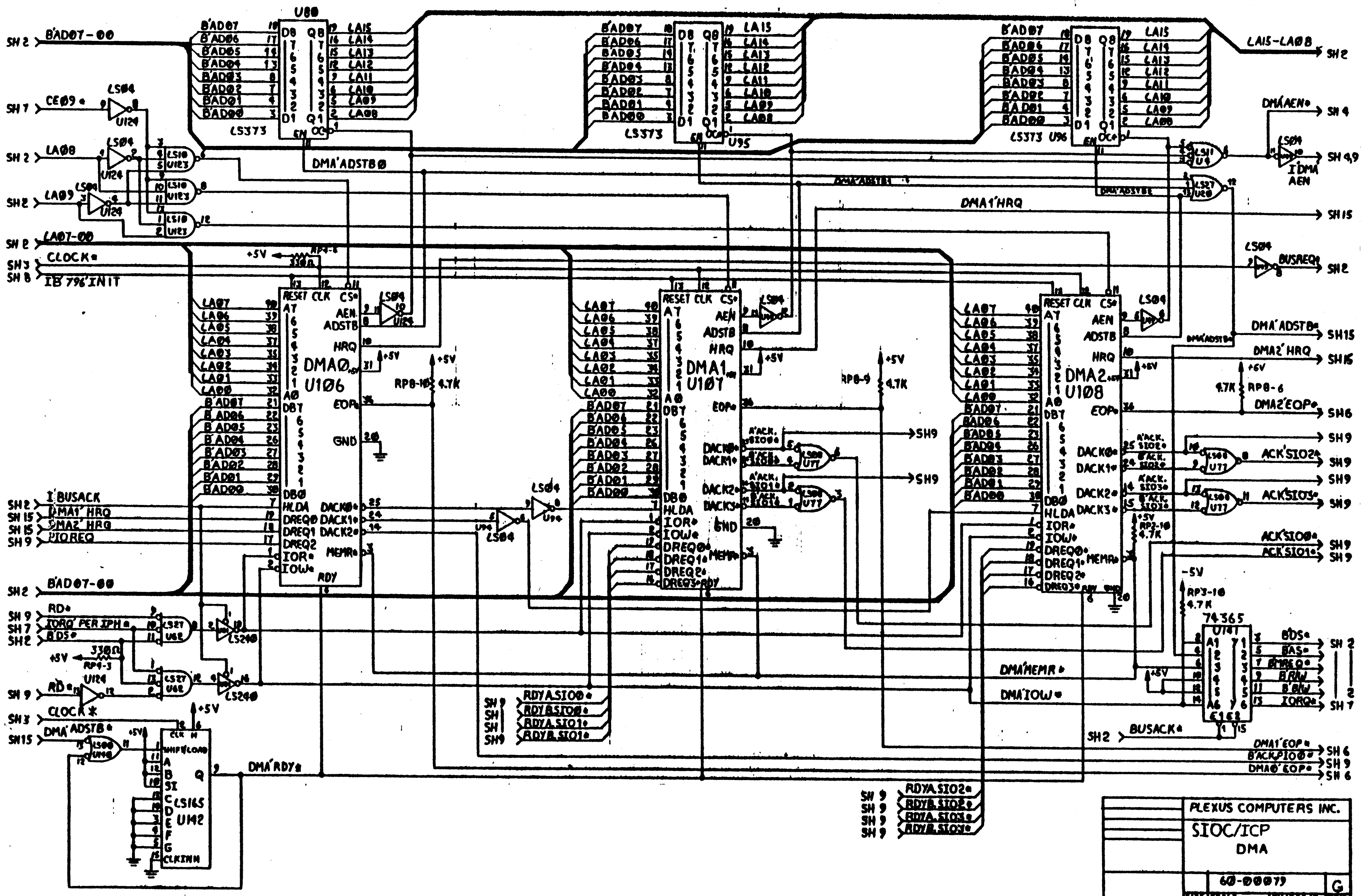
PLEXUS COMPUTERS INC.

SIOC/ICP

ROM'S AND DECODING

60-00079 G

SIZE SCALE — SHEET OF REV



PLEXUS COMPUTERS INC.

SIOC/ICP

DMA

60-00079

G

SIZE SCALE — SH 15 OF 15 REV

MEMORY

1. INTRODUCTION

The P/25 memory system contains 16K bytes of EPROM, 2K bytes of static RAM and from 512K bytes to 2M bytes of dynamic RAM. The EPROM and static RAM are located on the processor board, and the dynamic RAM is on separate memory boards.

All system memory, including EPROM, static RAM and dynamic RAM, features a dual-port memory access arrangement. All memory can be accessed by the onboard Z8001A or by another bus master via the Multibus interface.

The dynamic RAM includes an error detection and correction (EDC) feature and a refresh circuit, both located on the processor board.

Physical address locations for EPROM, static RAM and dynamic RAM are shown below:

EPROM	Segment 0, Offset 0000 to 7FFEH
Static RAM	Segment 0, Offset 8000 to 87FEH
Dynamic RAM	Segments 4 to 126, Offset 0000 to FFFEH

2. PRINCIPLES OF OPERATION

2.1. Physical and Logical Memory Space

The principle of physical and logical address space is essential to the P/25. While the dynamic RAM memory exists in a physical address space, it is never accessed by its physical address. To the rest of the P/25, data resides in a logical address space. The page map RAM (Figure 1, block 8), under control of the operating system, remembers the relationship between logical and physical memory. It acts like a file clerk, converting logical to physical addresses.

Each physical memory address is defined by 23 bits. This is shown in the Processor section, Figure 3. The 11 low-order bits are the offset, which pass through the page map RAM unchanged. The 12 high-order bits are manipulated by the page map RAM which uses them to select one of 32 map sets and one of up to 32 2K pages in the map set. The 11 low-order bits define the address within the 2K page.

For a detailed description of this process, see Processor section 2.1.1.

2.2. Dual-Port Memory

The dual-port memory system in the P/25 reduces the memory access overhead of the Z8001A, which shares common memory access with other Multibus masters. This dual-port arrangement allows the Z8001A and another Multibus master to access memory at the same time. The memory access cycles are interleaved and memory arbitration, performed by the local bus arbiter, is done on a single-word or byte basis.

2.3. Error Detection and Correction

The P/25 memory system includes error detection and correction (EDC) circuits, implemented with the AMD 2960 chip. These circuits, shown in Figure 1 as block 16, correct single-bit errors and detect double-bit errors.

The EDC generates checkbits based on a modified hamming code. For a 16-bit word, six checkbits are required. In a word-write operation, six checkbits are generated by the EDC and stored in the memory. These checkbits are read along with 16 data bits in a word-read operation. It then generates a new set of checkbits to compare with the checkbits read in. If a single-bit error is detected, the EDC corrects it and flags the ERROR line. If a double-bit error is detected, both the ERROR and MULTI-ERROR lines are activated. If no errors are detected, the EDC is transparent to the device accessing memory.

EDC errors cause the error status latches to save the address of the memory location that caused the error, the six-bit syndrome code generated by the EDC and several status bits describing the memory access. These are the same latches that store information about memory page attribute errors. The information format is shown in the Processor section, Figure 11.

When an EDC error occurs, the MMU goes into an error condition. Subsequent errors do not update the error status latches until the error condition is cleared by the processor, which performs a special write to port FFA0 (this address is one of the status latches).

The EDC is controlled by the processor through a special port at address FF80. It is used in internal control mode, and must be initialized before checking can be enabled. After initialization, checking is enabled by writing a logical one to bit A3 of the control port (FF80).

2.4. Circuit and Timing Analysis

This section analyzes the read/write operations for different types of memory cycles. These are:

(1) EPROM/static RAM read cycle

(2) Static RAM write cycle

(3) Dynamic RAM read cycle

(4) Dynamic RAM write cycle

(5) Dynamic RAM BYTE write cycle

Timing diagrams for the above memory cycles are shown in Figures 2 through 6.

2.4.1. Z8001A EPROM or Static RAM Read Cycle

The EPROM/static RAM read cycle is four clock cycles long. One wait-state is inserted to match the three-clock-cycle Z8001A running speed with the slow access time of EPROM and static RAM. The timing diagram is shown in Figure 2.

The read cycle begins in T1 when the address strobe (AS/) signal is asserted and the address is output. If the Multibus and refresh circuitry do not contend for local address and data buses, then ZGRANT/ and MEMRQ/ are pulled low sequentially, starting at the falling edge of clock cycle T1. MEMRQ/ is the signal to notify the timing sequencer (block 22) to start the timing program for a read cycle.

After receiving MEMRQ/, the timing sequencer synchronizes generation of the mapped physical address by asserting a PMREQ signal (active HIGH), which notifies the memory system that a physical address on connector P2 is valid. The memory address decoder (block 39) decodes the higher eight address lines to generate either an EPROM CS/ signal (see the processor schematics, page 4, U112 pins 6 and 8) or a static RAM CS/ signal (see the processor schematics, page 4, U109 pin 10). EPROM occupies the lower address space in segment 0. Static RAM occupies the upper space.

The WAIT/ line (block 40) is controlled by the wait-state generator. It is pulled low after the DRAM-EPROM/ signal (see the processor schematics, page 4, U3, pin 8) is activated. WAIT/ stays low for one clock cycle and is pulled back to high by the OESC/ line (block 22) which is controlled by the timing sequencer.

A memory cycle is then initiated. Maximum allowable access time for the parts used in the P/25 system is 450 nsec. The data read from EPROM/static RAM flows through data buffers (block 21) to the local memory/data bus and through another data buffer (block 5) to finally settle on the the Z8001A MA/D bus at the beginning of T3. This satisfies the 20-nsec minimum setup time before the falling edge of T3.

The timing sequencer is also responsible for generating a signal to indicate the end of a memory cycle. This signal, which is called CYCEND/, is a 66-nsec active-low pulse. The local bus arbiter repositions itself and starts arbitration for the next memory cycle.

2.4.2. Z8001A Static RAM Write Cycle

Figure 3 shows the timing for the Z8001A static RAM write cycle. Except for those pertaining to a write cycle, the circuit operations and timing analysis are the same as in the previous read cycle.

The most important timing consideration in a write cycle is the data setup time and data hold time. The 2114 static RAM (see the processor schematics -- U43 through U46) requires a minimum of

135-nsec data-setup time and 0-nsec data hold time. Both requirements are met as described in the next paragraph.

The write-enable (WE/) strobe is pulled low before the falling edge of T2 and pulled back to high after the falling edge of T3. Because a WAIT cycle is inserted, a minimum of 500-nsec active-low WE/ pulse is generated. This satisfies the minimum of 135-nsec WE/ pulse requirement. The valid data settles on the data inputs of the 2114 at the falling edge of T2. This guarantees a 500-nsec data setup time. The data is not removed before the WE/ signal is pulled to HIGH, so the 0-nsec data hold time is satisfied.

2.4.3. Dynamic RAM Read Cycle

The Z8001A runs at a frequency of 5 Mhz, which implies a 600-nsec memory cycle when no wait-state is inserted.

NOTE: This time accounts for memory mapping (necessary in a multi-user, multi-tasking environment) and error detection and correction.

The timing diagram for a read cycle is shown in Figure 4. The figure also shows the relative timing relationships between when the Z8001A initiates the cycle and when the Multibus initiates the cycle.

A read cycle begins with the assertion of PMREQ (active HIGH). The memory board with switch settings that match the highest five address lines on connector P2 is selected and a BRDSEL signal is generated. The corresponding row address strobe (RAS/) of the bank selected by the next three address lines is also activated, which latches in the row address. The row address consists of the lowest seven address lines, which are selected by the multiplexor.

The column address strobe (CAS/) and the signal which selects the column address are controlled by a delay line.

NOTE: Since the CAS/ and the column addresses are generated from two different paths, it is important to note that the skew problem does not exist in these two paths. This is shown in the following worst-case timing calculation:

Generation of MUX (Delay Line)	max	50 nsec
Multiplexors select COLUMN address (propagation delay from SEL input to data output)	max	15 nsec

```
    Column address buffers (74S240)          max     9 nsec
                                           ----------------
                                             max    74 nsec


    Generation of CAS/ (delay line)          min    75 nsec

    CAS/ buffers (74S241)                    min     3 nsec
                                           ----------------
                                             min    78 nsec
```

The above calculations show a 4-nsec minimum setup time for the column address before activation of the CAS/ signal.

After activation of CAS/, a maximum of 135 nsec is required to place the data on the output pins of the 64K memory chips.

2.4.3.1. EDC Timing

Latch Enable In (LEIN) controls latching of the input data. When LEIN is low, the data input latch and checkbit input latch are latched to their previous state. The LEIN (see the processor schematics, page 13) is pulled low when the 16-bit data and six checkbits read from the memory are stabilized. The EDC chip then generates six checkbits for the 16-bit data in the input latch and compares them with the six checkbits in the checkbits input latch. If one or more errors are detected, ERROR/ goes low. If two or more errors are detected, MUL-ERROR/ goes low. The data (corrected if a single-bit error is detected) is latched into the EDC output latch when LEOUT signal is pulled low.

NOTE: The output latches are enabled by OE0/ and OE1/ before LEOUT is pulled low.

If the Z8001A initiates the memory cycle, then the data is latched into the Z8001A buffers at the falling edge of clock cycle T3. If the cycle is initiated by the Multibus, then data is latched into U31 and U52 when CYCEND/ is asserted. In either case, the end of a memory cycle is signaled by the assertion of the CYCEND/ signal controlled by the timing sequencer.

2.4.4. Dynamic RAM Write Cycle

The timing diagram for the dynamic RAM write cycle is shown in Figure 5. The BRDSEL, RAS/, CAS/, ROW address and COLUMN address are generated in the same way as in the read cycle.

In a write cycle, the EDC chip is operated in the GENERATE mode. In this mode, six checkbits are generated to match the 16-bit data on the local data bus. The checkbits are present at the outputs of CB0 to CB5. The output latch in the EDC chip is disabled in a write cycle. Timing for the 16-bit data and six checkbits available in the RAM is calculated as follows:

Data available on the local bus (data from Multibus available sooner)	max	87 nsec
Checkbits generation	max	32 nsec
Data buffer delay (block 32, 74S240)	max	7 nsec
Data buffer delay (memory board)	max	7 nsec
	max	133 nsec

The above timing is referenced to the rising edge of clock cycle T2. The WE/ strobe in a dynamic RAM write cycle is generated approximately 180 nsec after the rising edge of T2. It stays low for about 130 nsec (mininum WE/ low requirement is 55 nsec). The later activation of WE/ ensures meeting the 0-nsec data setup time requirement for 64K dynamic RAMs.

Like the read cycle, the timing sequencer generates the CYCEND/ to end a memory cycle.

2.4.5. Dynamic RAM Byte-Write Cycle

The dynamic RAM byte-write operation requires special attention. It actually consists of a word-read and a word-write operation. When the byte-write operation is performed, a wait-state is inserted.

The word-read is performed because the six checkbits are generated for the corresponding 16-bit data. When a low byte or a high byte is going to be changed, the entire word with six checkbits must be read from memory. In the read cycle, the new byte (along with the byte which is not to be changed) are latched into the input latch in the EDC chip. A word-write operation is then performed. The EDC chip generates a new set of six checkbits for the new 16-bit data.

Figure 6 shows the timing diagram for the byte-write operation.

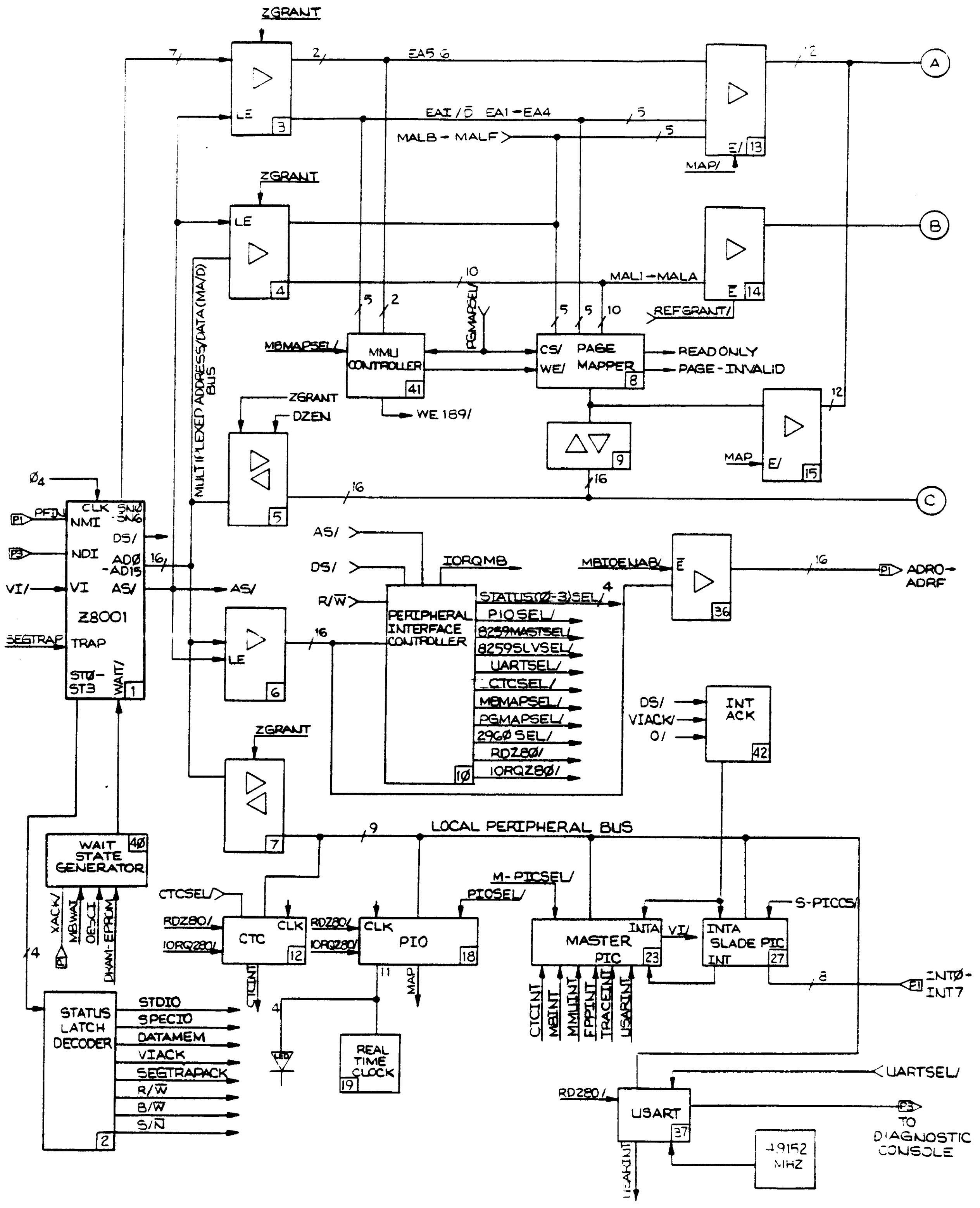


Figure 1. P/25 Processor Board Block Diagram -- Part A

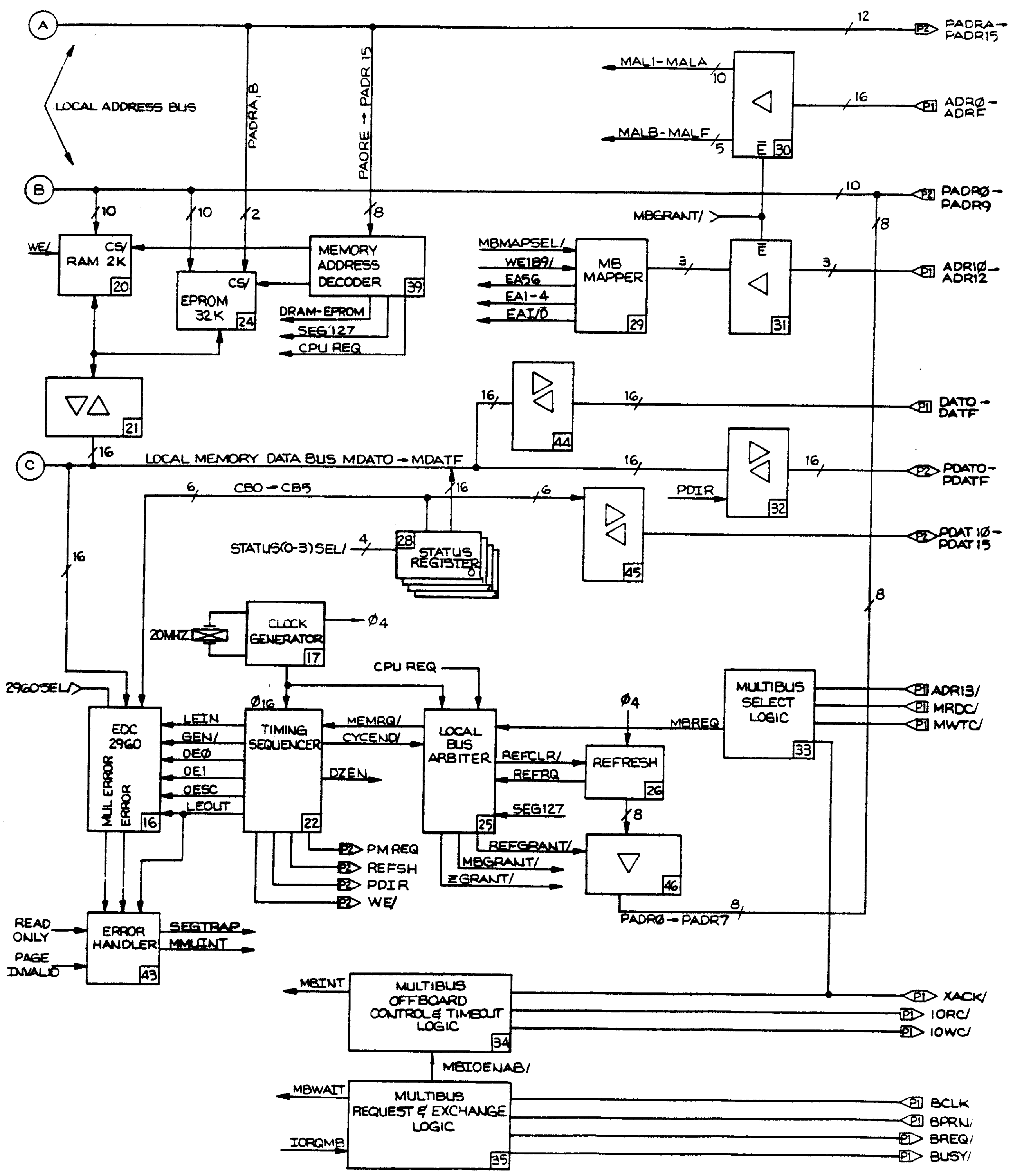


Figure 1. P/25 Processor Board Block Diagram -- Part B

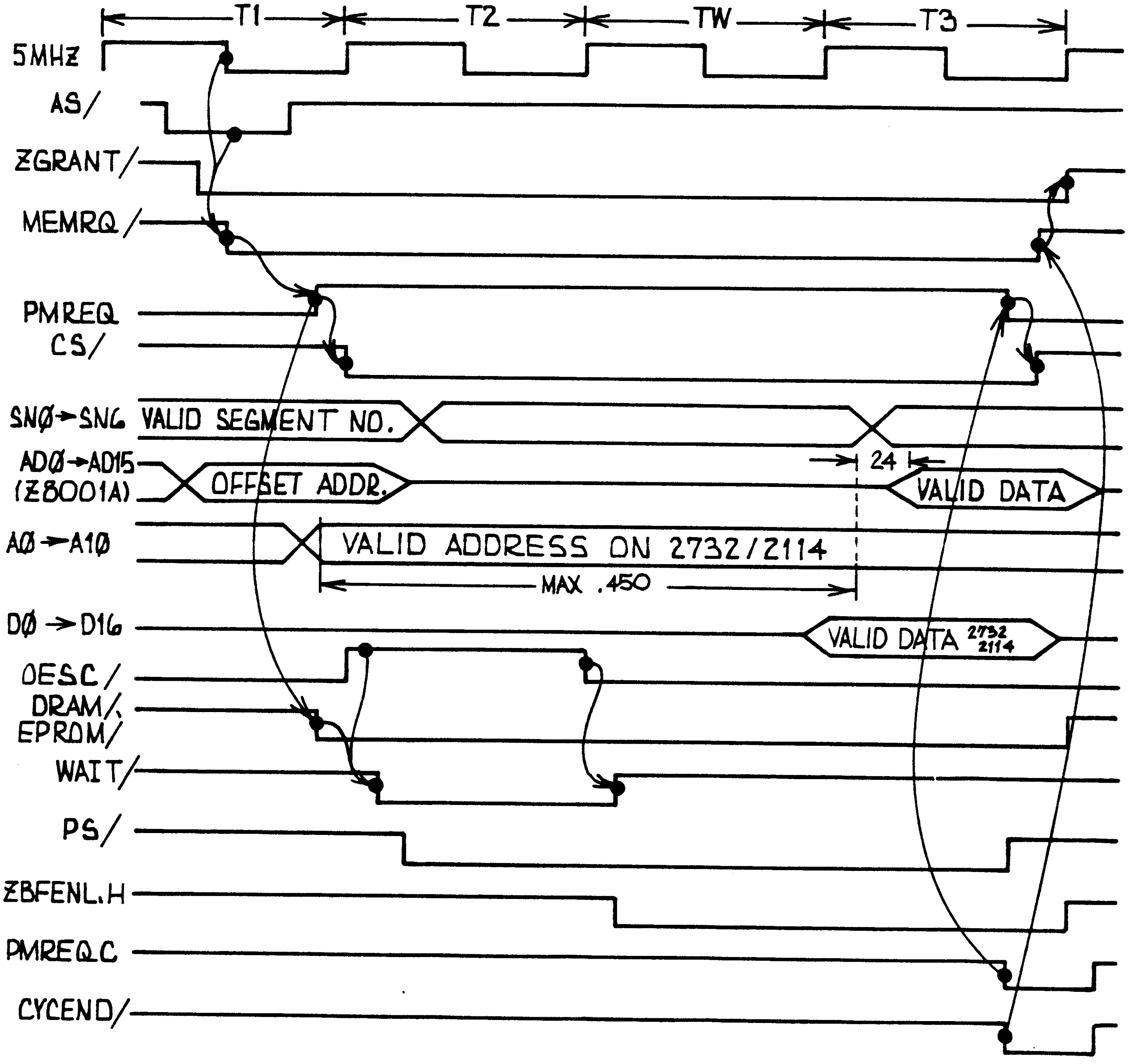


Figure 2. CPU-EPROM/Static RAM Read Cycle

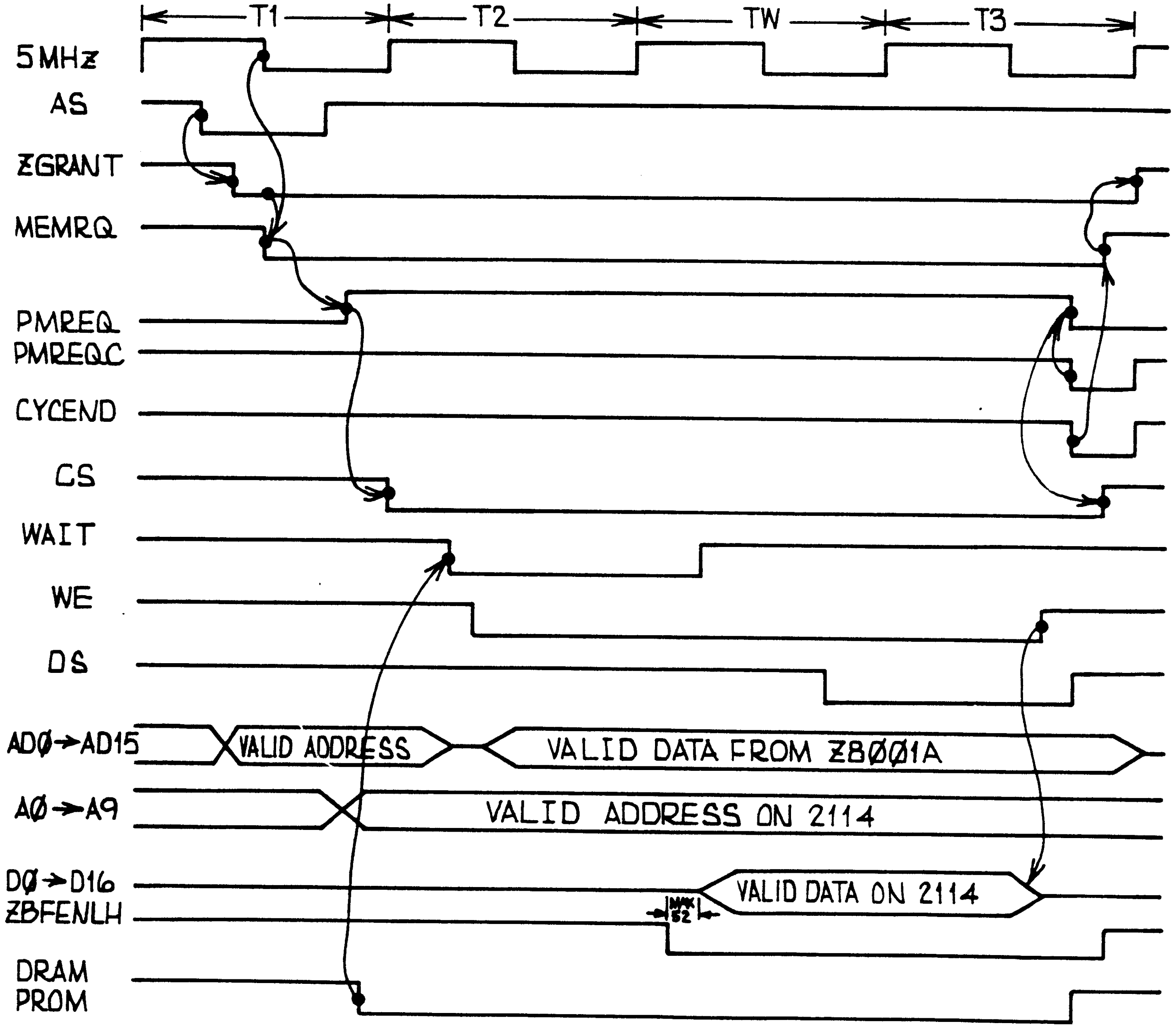


Figure 3. CPU-Static RAM Write Cycle

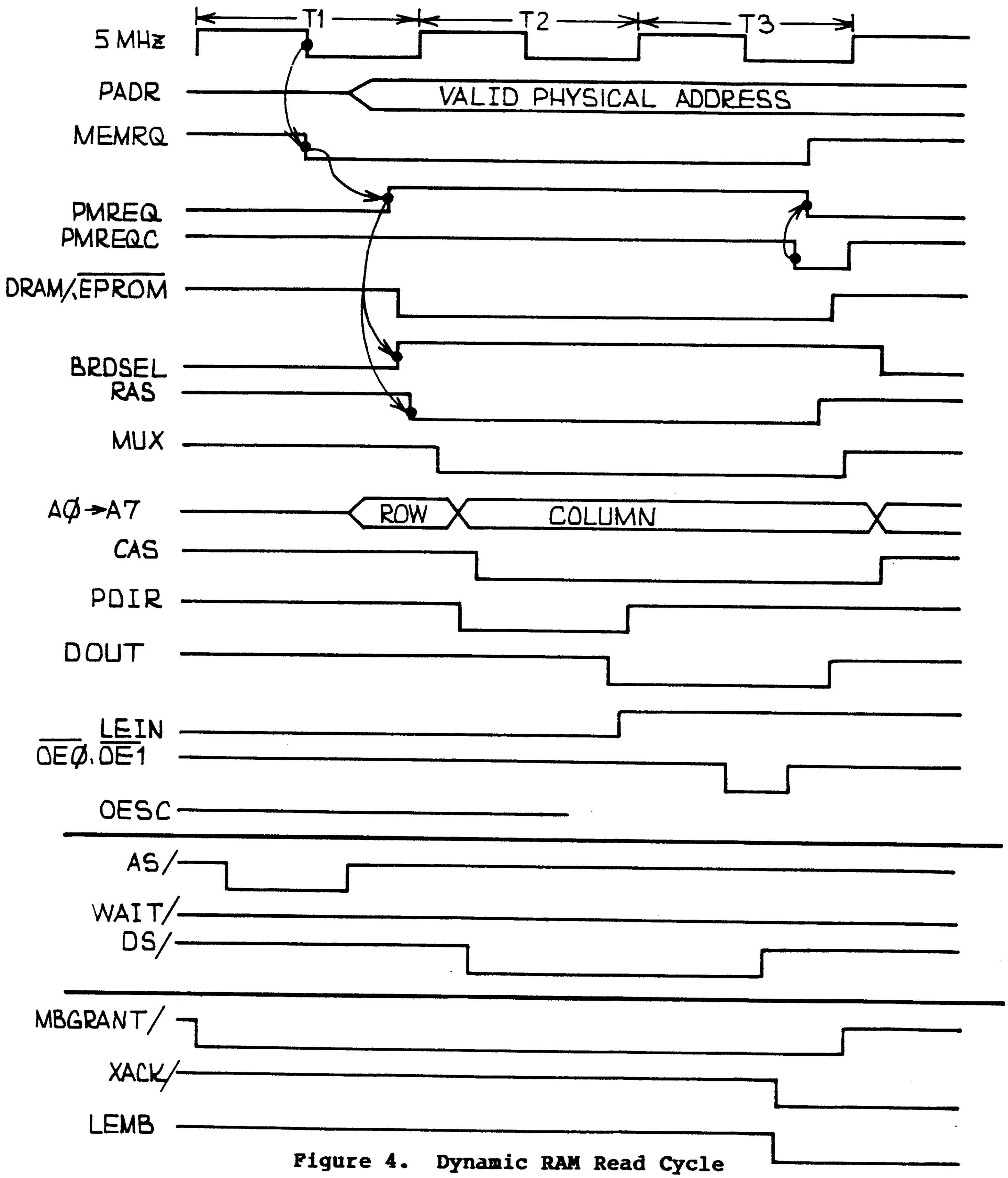


Figure 4. Dynamic RAM Read Cycle

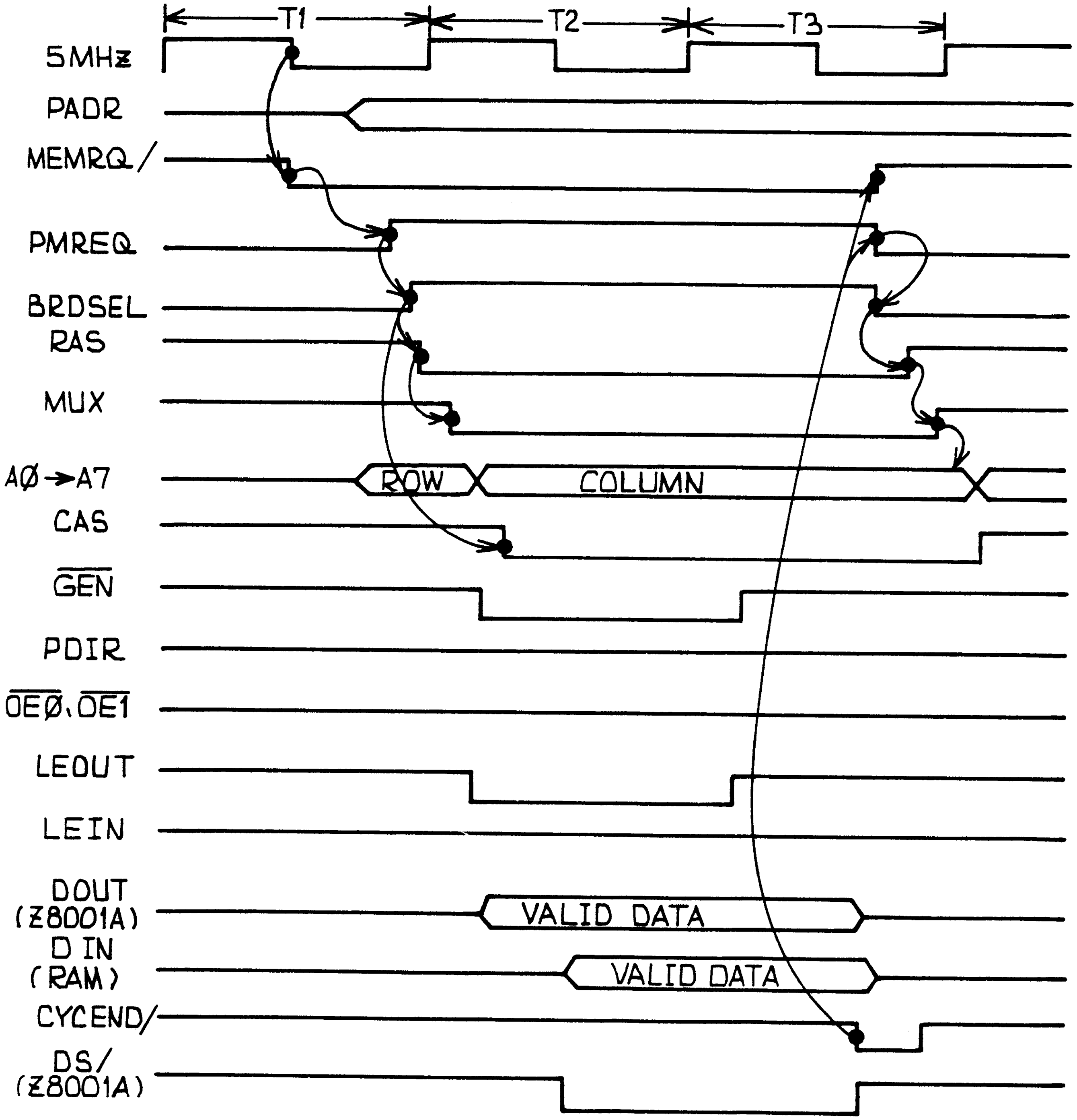
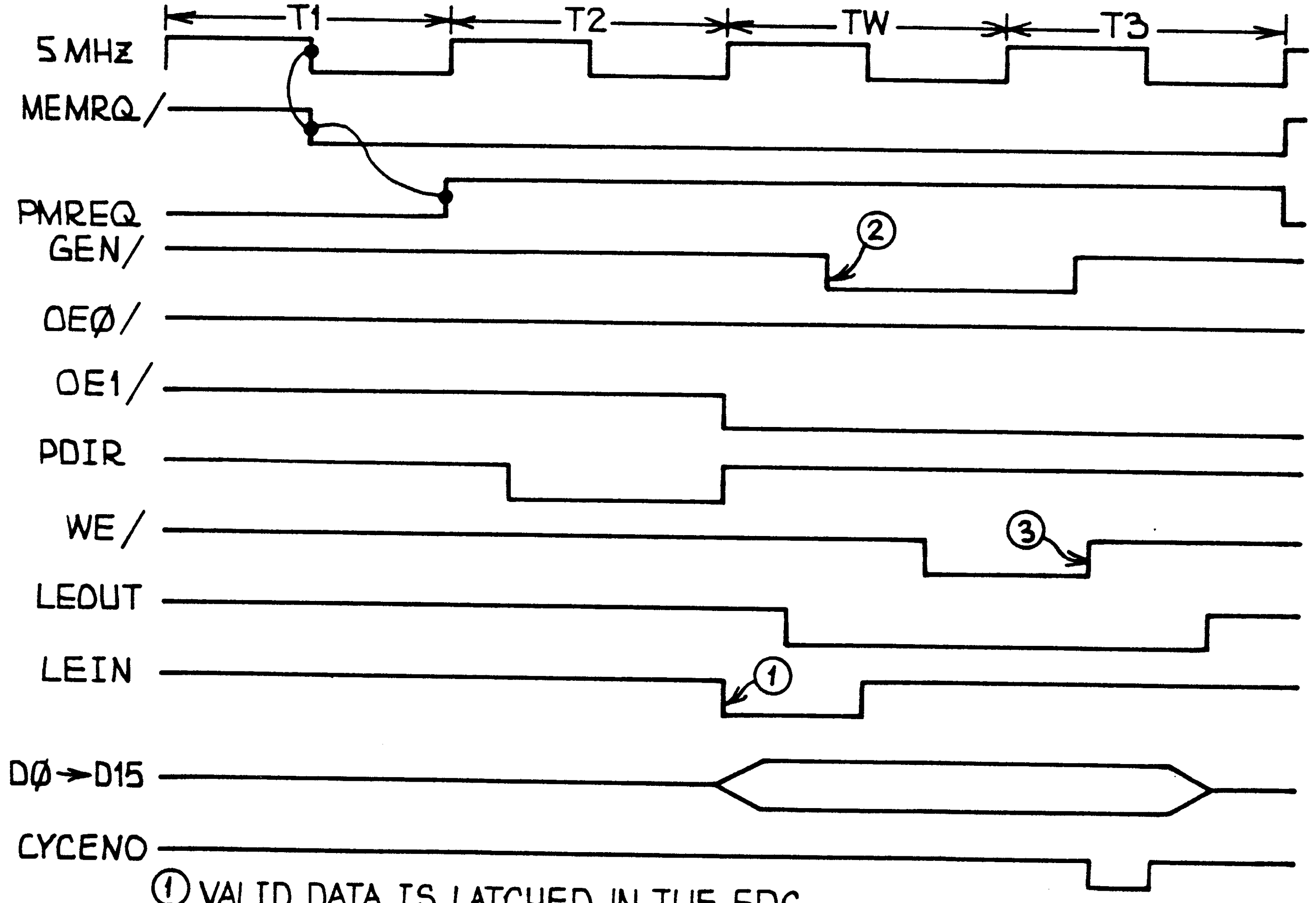


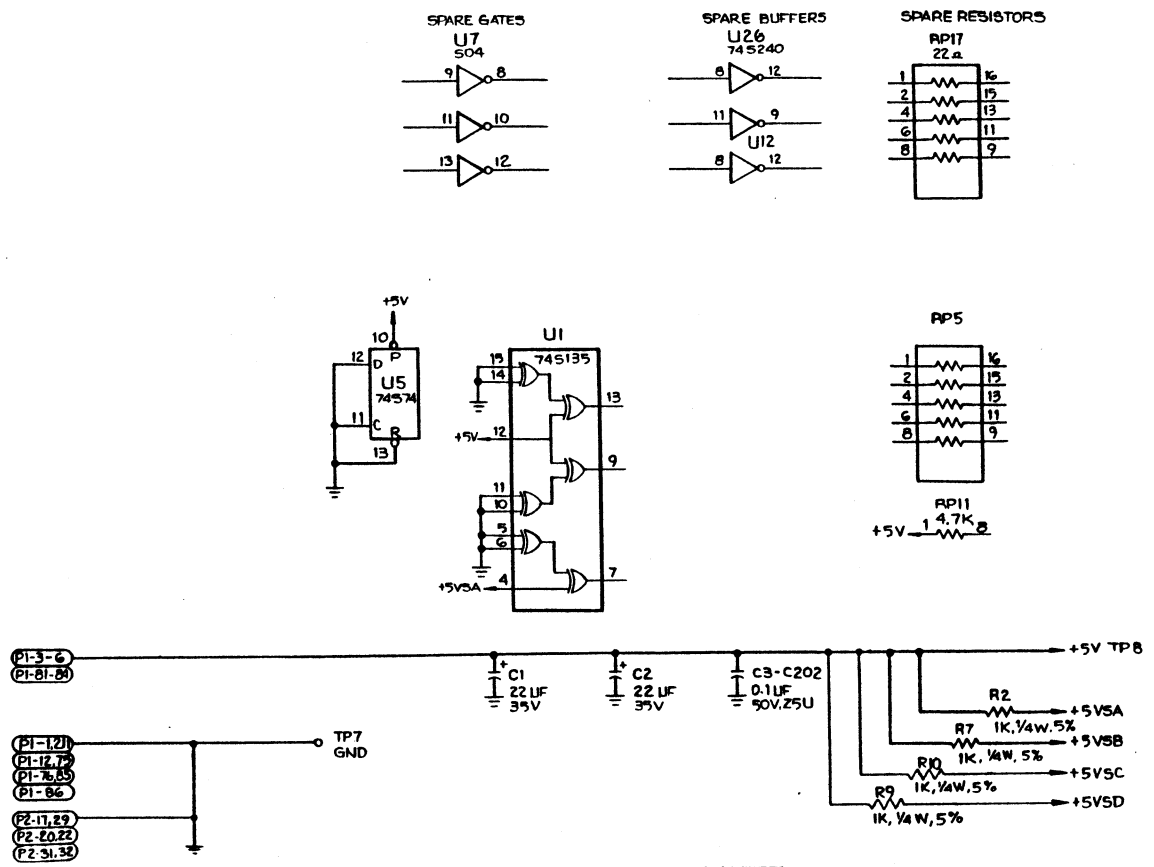
Figure 5. Dynamic RAM Write Cycle



① VALID DATA IS LATCHED IN THE EDC.
② 6 NEW CHECKBITS ARE GENERATED FOR NEW 16-BIT DATA.
③ DATA IS STORED IN THE MEMORY CHIP.

Figure 6. Dynamic RAM Byte Write

	REV	DESCRIPTION	DATE	APPV
AS 60-00088-X	A	ENGR. PROD. RELEASE, EN 84	1-12 82	WAM
AS 60-00089-X	A	EN145 P2 CONN CHANGED	5-21-82	WAM



REF. DES.	DEVICE TYPE	SHT. NO.	UNUSED GATES	+5V	GND
U1	S135	2		16	8
U2	S135	2		16	8
U3	S241	2		20	10
U4	S241	2		20	10
U5	S74	2			
U6	S00	2,4		14	7
U7	S04	4		14	7
U8	S30	2		14	7
U9	S241	2		20	10
U10	S157	2		16	8
U11	S157	2		16	8
U12	S240	4		20	10
U13	LS240	4		20	10
U14	S240	4		20	10
U15	S138	2			
U16	S241	2		20	10
U17	LS240	2		20	10
U18	LS240	2		20	10
U19	LS240	2		20	10
U20	LS240	2		20	10

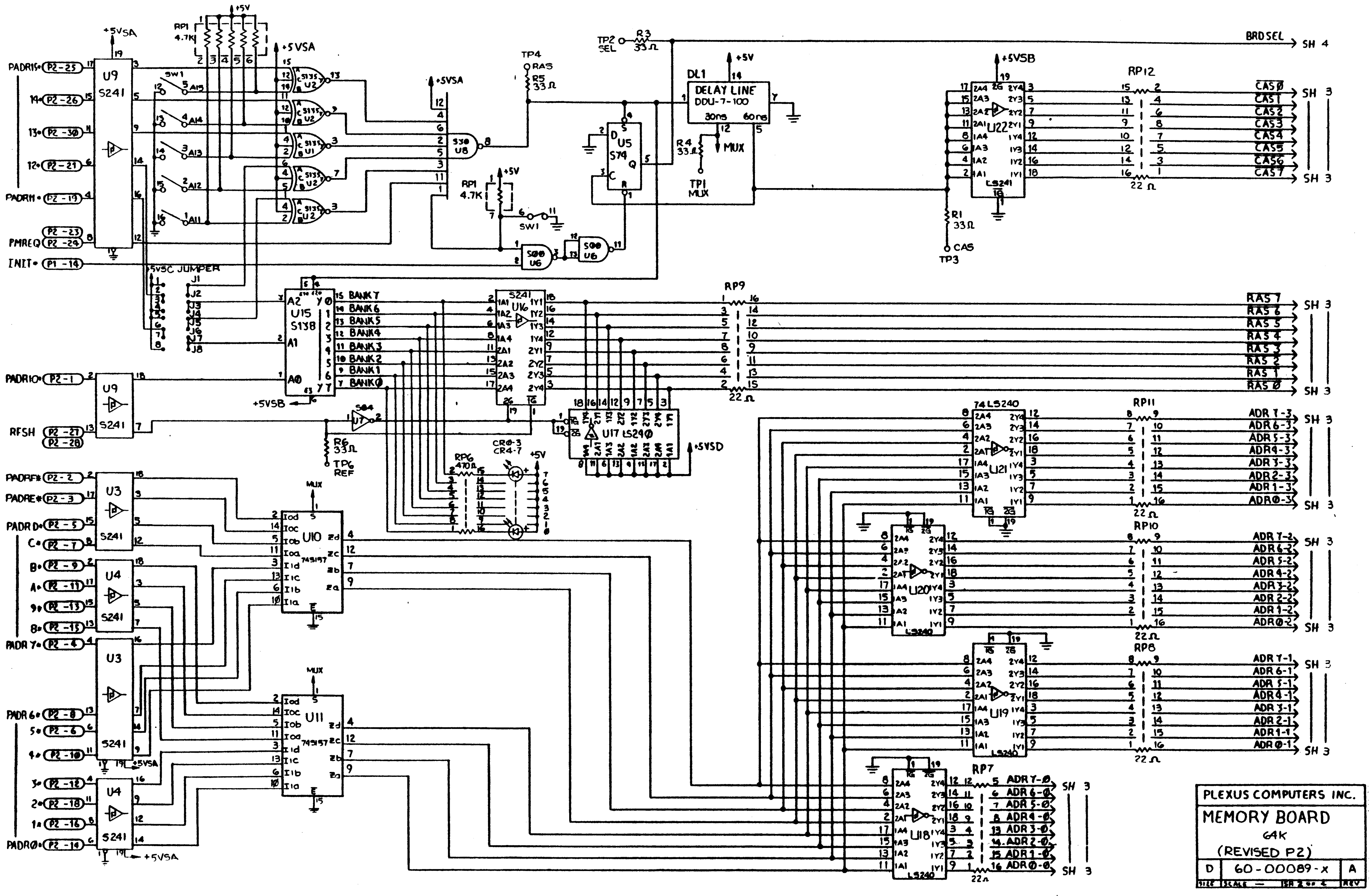
REF. DES.	DEVICE TYPE	SHT. NO.	UNUSED GATES	+5V	GND
U21	LS240	2		20	10
U22	LS241	2		20	10
U23	LS241	4		20	10
U24	LS240	4		20	10
U25	S240	4		20	10
U26	S240	4		20	10
U27-202	TMS4164-15	3		8	16
DL1	DDU-7-100	2		14	7

1)

DASH NO	CAPACITY
-1	1/4 MBYTE
-2	1/2 MBYTE
-4	1 MBYTE

NOTES:

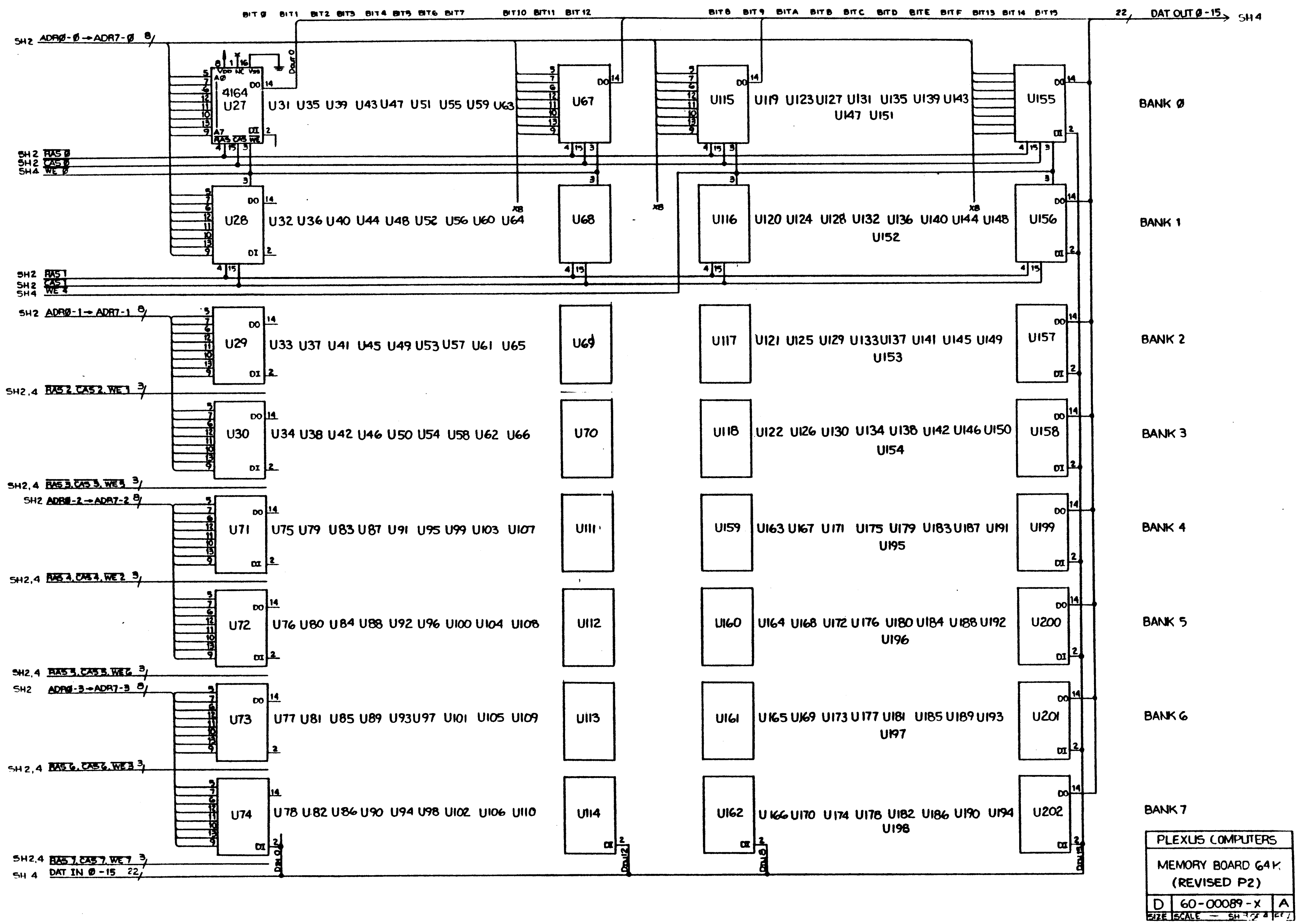
L. GAVKA 10 1/12/82	PLEXUS COMPUTERS INC.
[illegible] 5-21-82	MEMORY BOARD 64K (REVISED P2)
SIZE D	60-00089-X
SCALE: ~	SHT. 1 OF 4
REV	A



PLEXUS COMPUTERS INC.

MEMORY BOARD
64K
(REVISED P2)

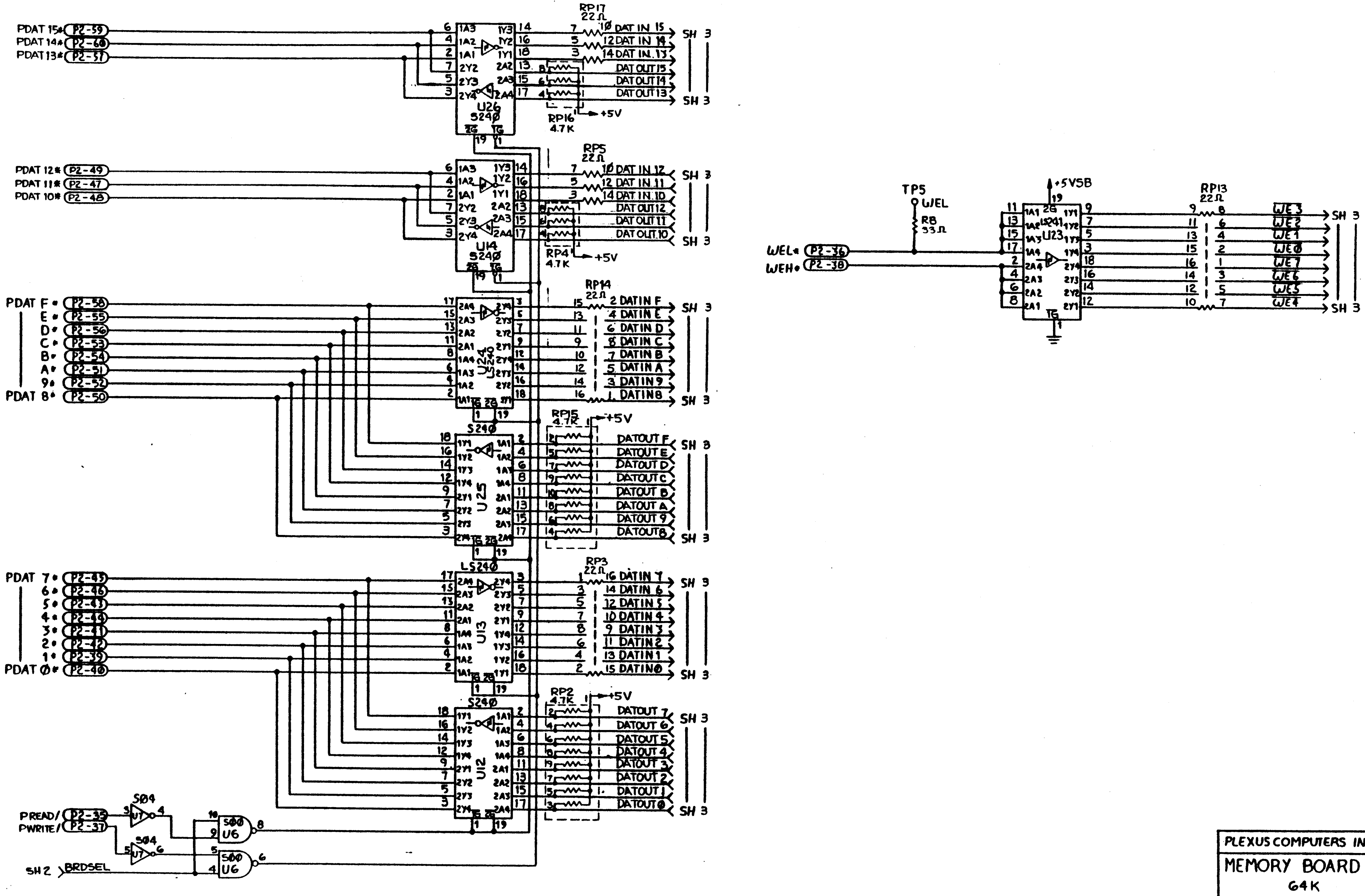
D	60-00089-x	A
SIZE	SCALE — SH 2 OF 2	REV


BIT Ø BIT1 BIT2 BIT3 BIT4 BIT5 BIT6 BIT7
BIT10 BIT11 BIT12
BIT8 BIT9 BITA BITB BITC BITD BITE BITF BIT13 BIT14 BIT15
22 DAT OUT Ø-15 SH4
SH2 ADRØ-Ø → ADR7-Ø 8
4164 U27
U31 U35 U39 U43 U47 U51 U55 U59 U63
U67
U115
U119 U123 U127 U131 U135 U139 U143 U147 U151
U155
BANK Ø
SH2 RAS Ø
SH2 CAS Ø
SH4 WE Ø
U28
U32 U36 U40 U44 U48 U52 U56 U60 U64
U68
U116
U120 U124 U128 U132 U136 U140 U144 U148 U152
U156
BANK 1
SH2 RAS 1
SH2 CAS 1
SH4 WE 4
SH2 ADRØ-1 → ADR7-1 8
U29
U33 U37 U41 U45 U49 U53 U57 U61 U65
U69
U117
U121 U125 U129 U133 U137 U141 U145 U149 U153
U157
BANK 2
SH2,4 RAS 2, CAS 2, WE 1 3
U30
U34 U38 U42 U46 U50 U54 U58 U62 U66
U70
U118
U122 U126 U130 U134 U138 U142 U146 U150 U154
U158
BANK 3
SH2,4 RAS 3, CAS 3, WE 5 3
SH2 ADRØ-2 → ADR7-2 8
U71
U75 U79 U83 U87 U91 U95 U99 U103 U107
U111
U159
U163 U167 U171 U175 U179 U183 U187 U191 U195
U199
BANK 4
SH2,4 RAS 4, CAS 4, WE 2 3
U72
U76 U80 U84 U88 U92 U96 U100 U104 U108
U112
U160
U164 U168 U172 U176 U180 U184 U188 U192 U196
U200
BANK 5
SH2,4 RAS 5, CAS 5, WE 6 3
SH2 ADRØ-3 → ADR7-3 8
U73
U77 U81 U85 U89 U93 U97 U101 U105 U109
U113
U161
U165 U169 U173 U177 U181 U185 U189 U193 U197
U201
BANK 6
SH2,4 RAS 6, CAS 6, WE 3 3
U74
U78 U82 U86 U90 U94 U98 U102 U106 U110
U114
U162
U166 U170 U174 U178 U182 U186 U190 U194 U198
U202
BANK 7
SH2,4 RAS 7, CAS 7, WE 7 3
SH 4 DAT IN Ø-15 22
x8

PLEXUS COMPUTERS

MEMORY BOARD 64K (REVISED P2)

SIZE		SCALE	SH
D	60-00089-X		A



PLEXUS COMPUTERS INC.

MEMORY BOARD
64K
(REVISED P2)

SIZE		REV
D	.60-00089-X	A
	SCALE — SH 4 OF 4	

REF. DES.	COMP. ID.	GATES NOT USED	REF. DES.	COMP. ID.	GATES NOT USED
U1	S135	3	U1Ø1-U116	4116	
U2	S135				
U3	S157		U117	LS24Ø	
U4	S157		U118-U157	4116	
U5	S241				
U6	LS24Ø		U158	S24Ø	
U7	LS241		U159-U174	4116	
U8	S138				
U9	S74	1	U175	S24Ø	
U1Ø	SØØ		U176-U199-	4116	
U11	SØ4	3			
U12	S3Ø		DL1	DDU4-5125	
U13	LS24Ø				
U14	LS24Ø		RP4-8,10		
U15	LS24Ø		,11,13	33Ω IND DIP	
U16	LS24Ø		RP1	4.7K SIP	
U17-U24	4116		R2,3	56Ω IND DIP	
U25	LS241				
U26-U41	4116				
U42	S24Ø				
U43-U58	4116				
U59	LS24Ø				
U6Ø-U99	4116				
U1ØØ	S24Ø				

60-00078-1

REV	DESCRIPTION	DATE	APPVD
A	PILOT RELEASE		
B	ENG. PRODUCTION RELEASE . EN 63	10-29-81	WMM
A	WAS: 60-00078-1 IS: 60-00087, EN	5-13-82	PZL

NOTES:

1. CAPACITORS ARE IN MICROFARADS, +80%, -20% 25VDC, TANTALUM.
2. RESISTANCE VALUES ARE IN OHMS. ±5%, ¼ WATT.
3. ~~DELETE~~(EN 125)

$\boxed{1}$ BYPASS EVERY OTHER CHIP IN RAM ARRAY (88 TOTAL) 11 PER BANK.

$\boxed{2}$ BYPASS EVERY 7 CHIPS IN RAM ARRAY (32 TOTAL) 4 PER BANK.

$\boxed{3}$ BYPASS EVERY NON MEMORY CHIP (24 TOTAL).

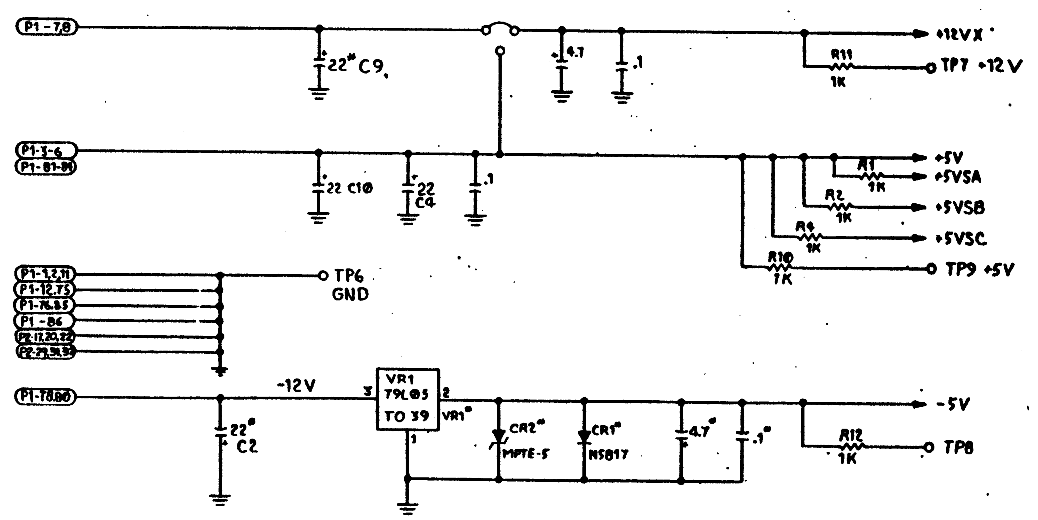
$\boxed{4}$ ~~DELETE~~ (EN 123)

$\boxed{5}$ ~~DELETE~~ (EN 123)

$\boxed{6}$ BYPASS EVERY 11 CHIPS IN RAM ARRAY (24 TOTAL) 3 PER BANK.

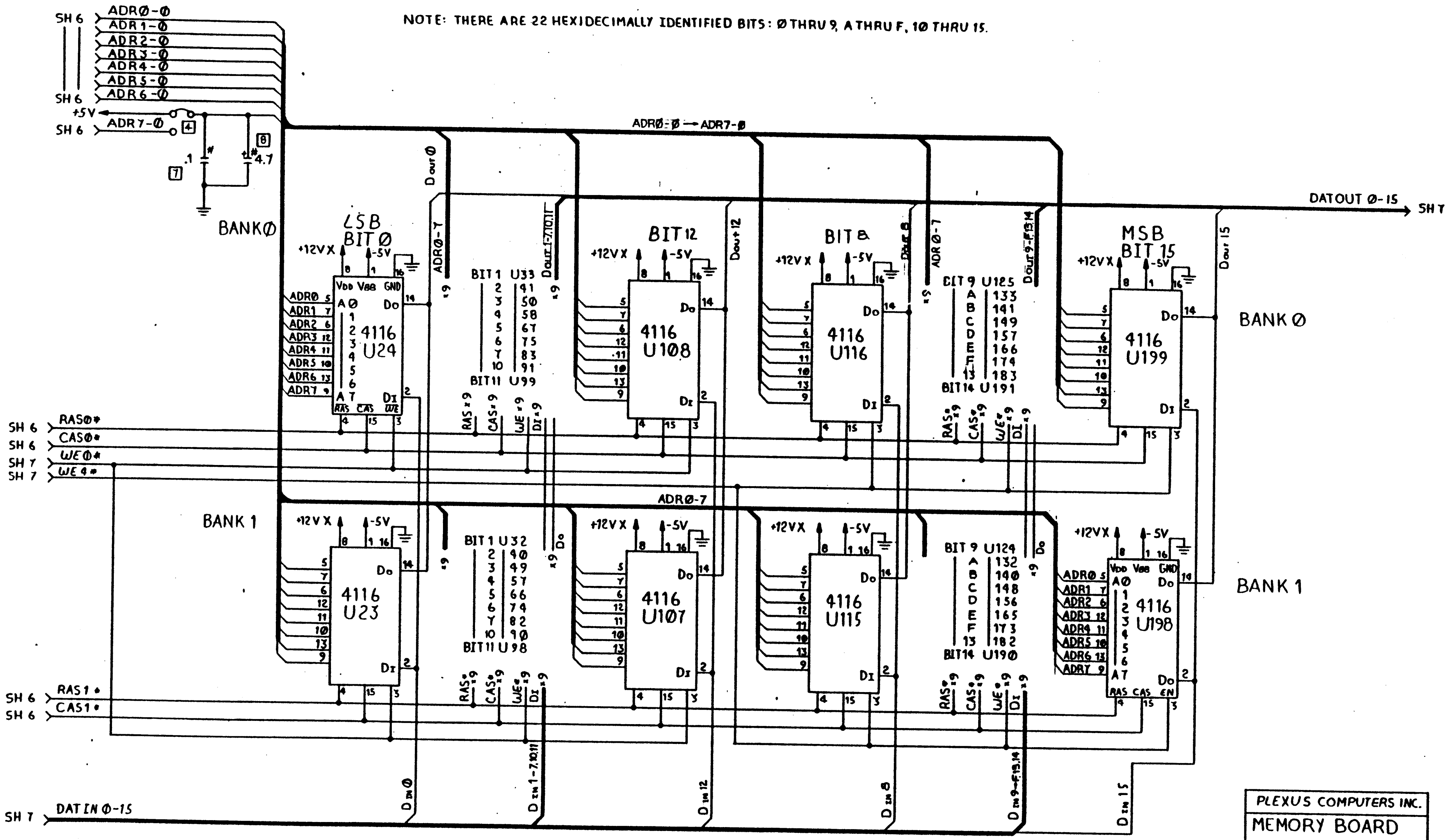
$\boxed{7}$ ~~DELETE~~ (EN 123)

$\boxed{8}$ BYPASS EVERY 22 CHIPS IN RAM ARRAY (8 TOTAL) 1 PER BANK.



[illegible] 5-25-81	PLEXUS COMPUTERS INC.
	MEMORY BOARD 16K
	D | 60-00087 | A
	SIZE | SCALE — | SH 1 OF 7 | REV

NOTE: THERE ARE 22 HEXIDECIMALLY IDENTIFIED BITS: Ø THRU 9, A THRU F, 1Ø THRU 15.

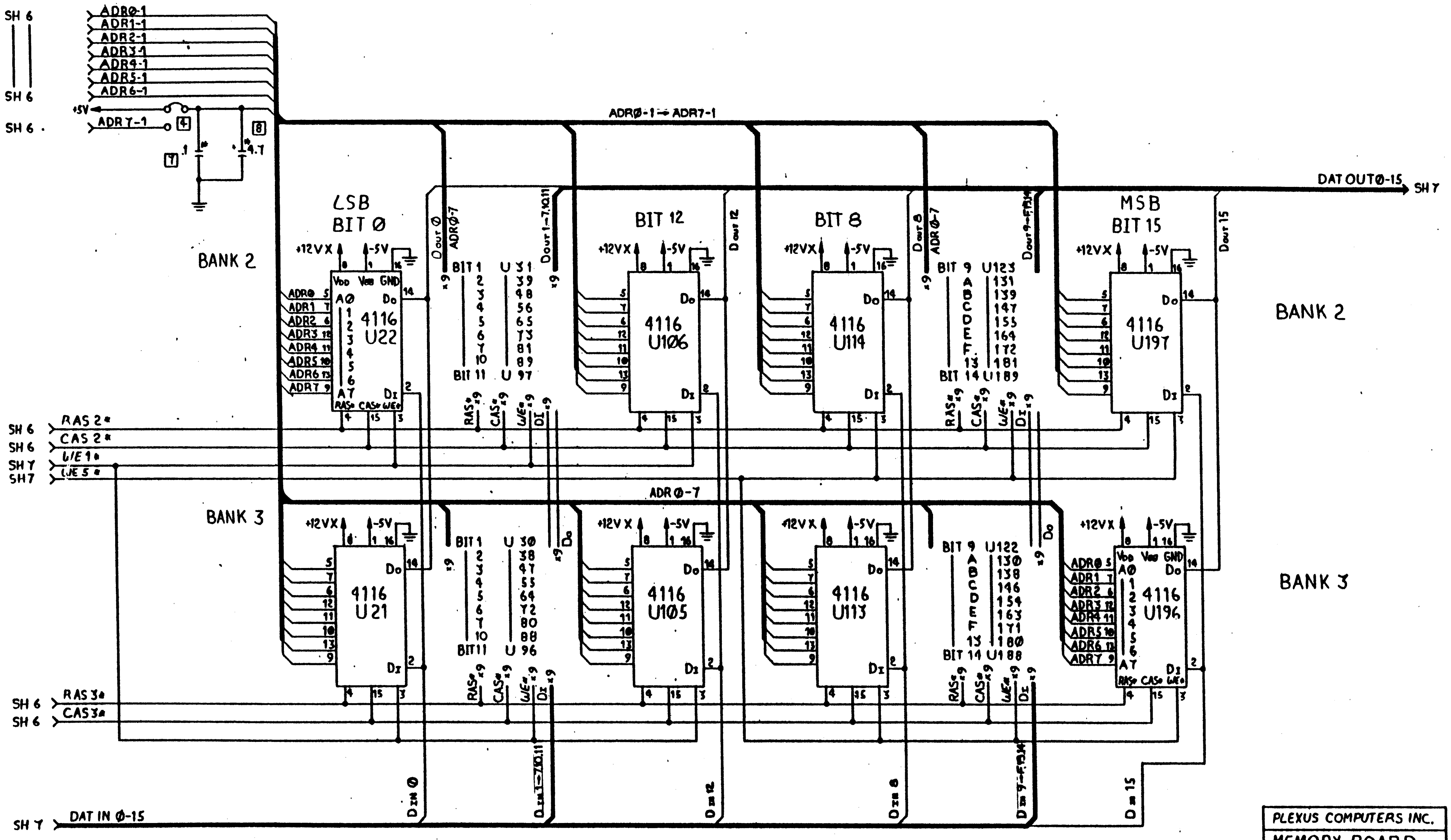


PLEXUS COMPUTERS INC.

MEMORY BOARD

16K

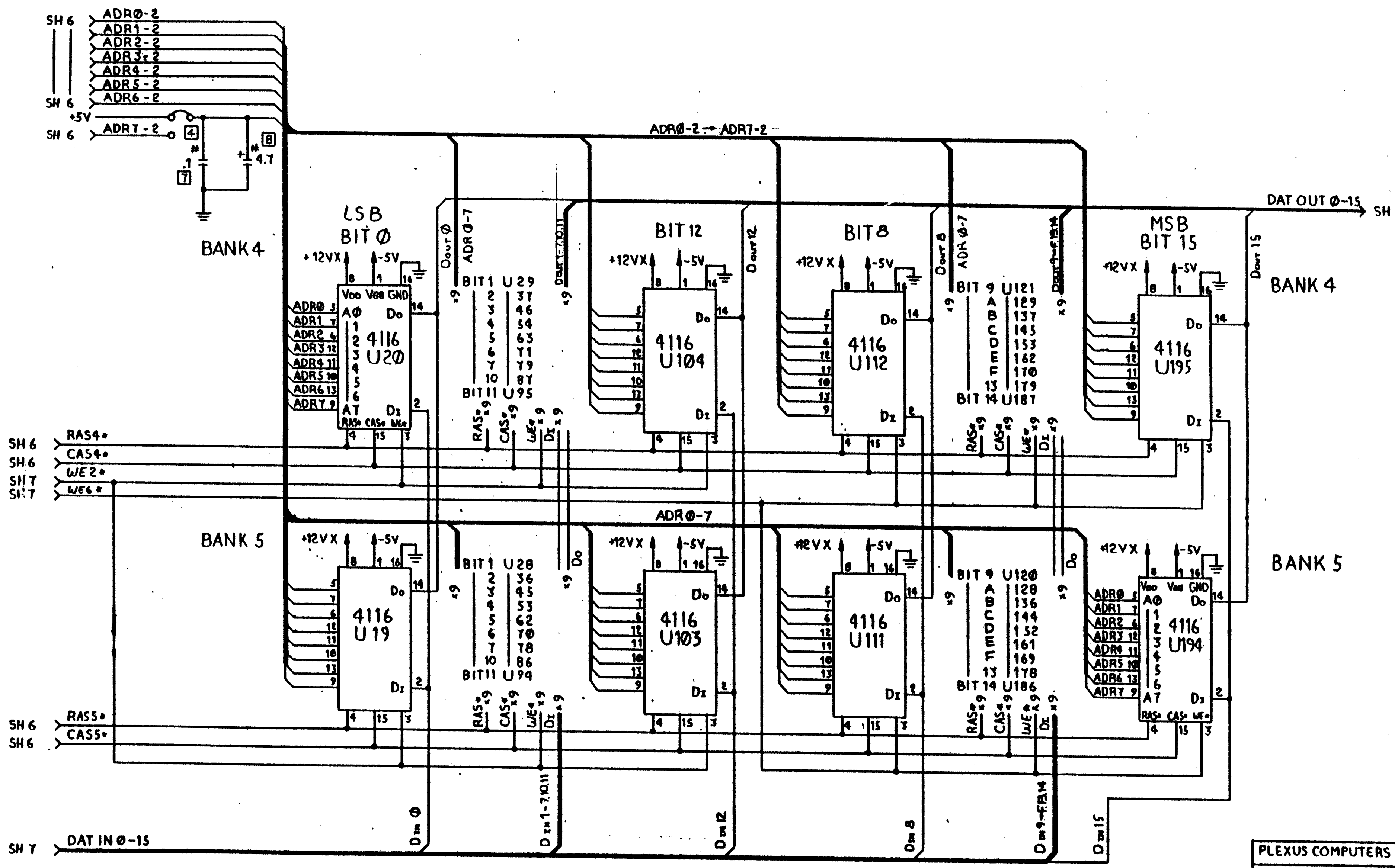
SIZE	SCALE —	SH 2 OF 7	REV
D		6C-00087	A



PLEXUS COMPUTERS INC.

MEMORY BOARD
16K

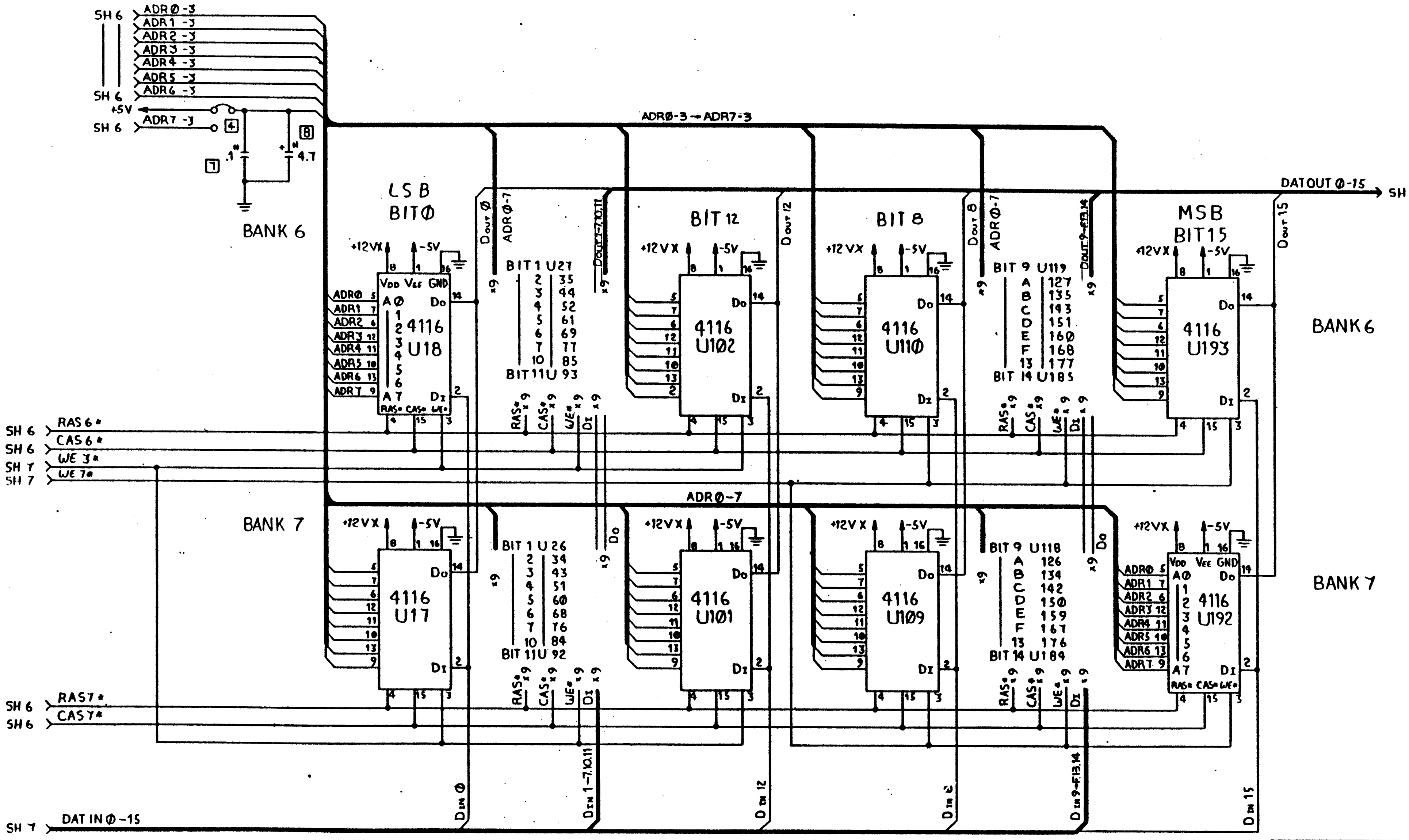
SIZE	SCALE —	SH 3 OF 7	REV
D	60-00087		A



PLEXUS COMPUTERS INC.

MEMORY BOARD
16K

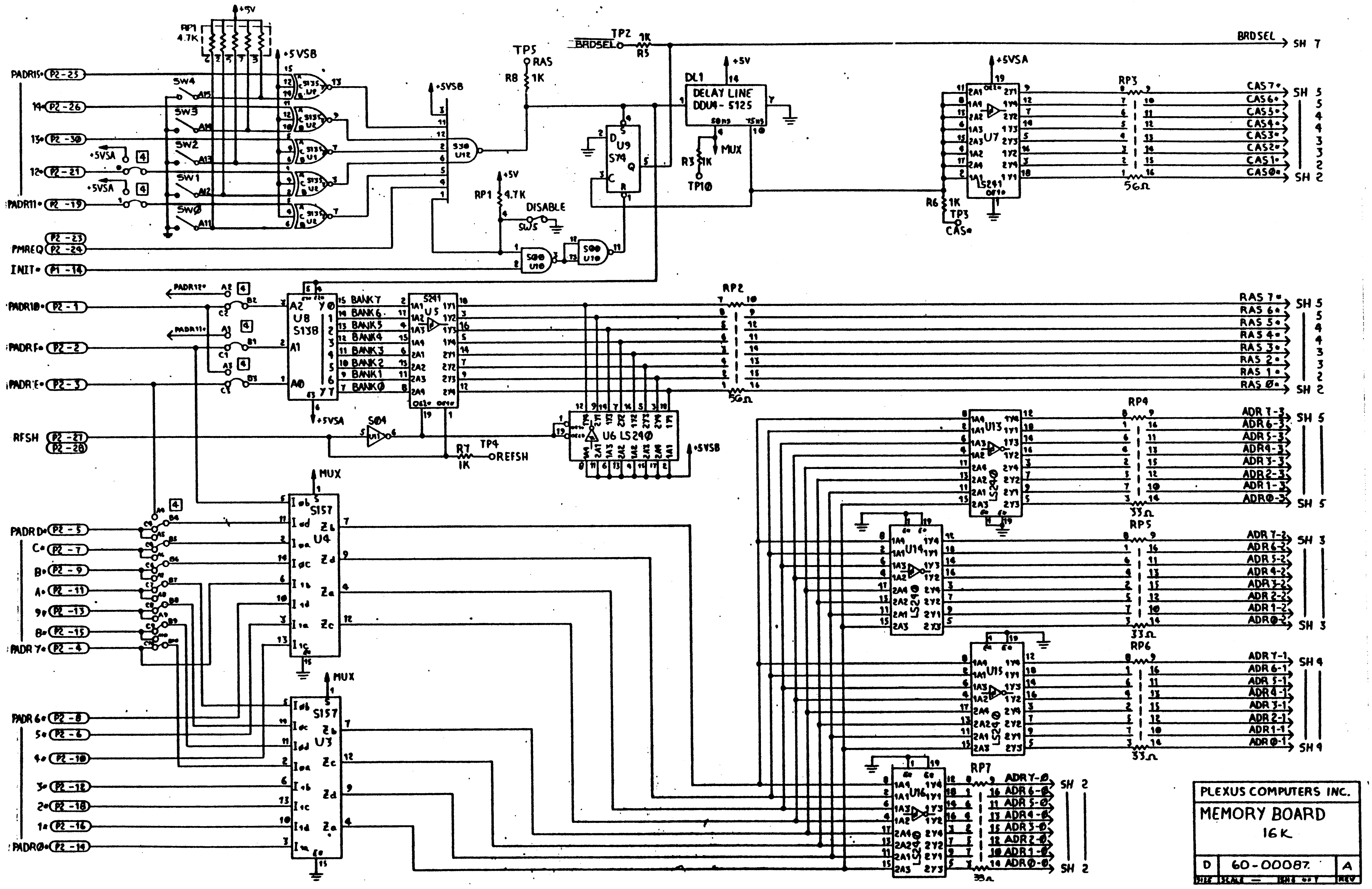
SIZE		SCALE — SH 4 OF 7	REV
b	60-00087		A



PLEXUS COMPUTERS INC.

MEMORY BOARD 16K

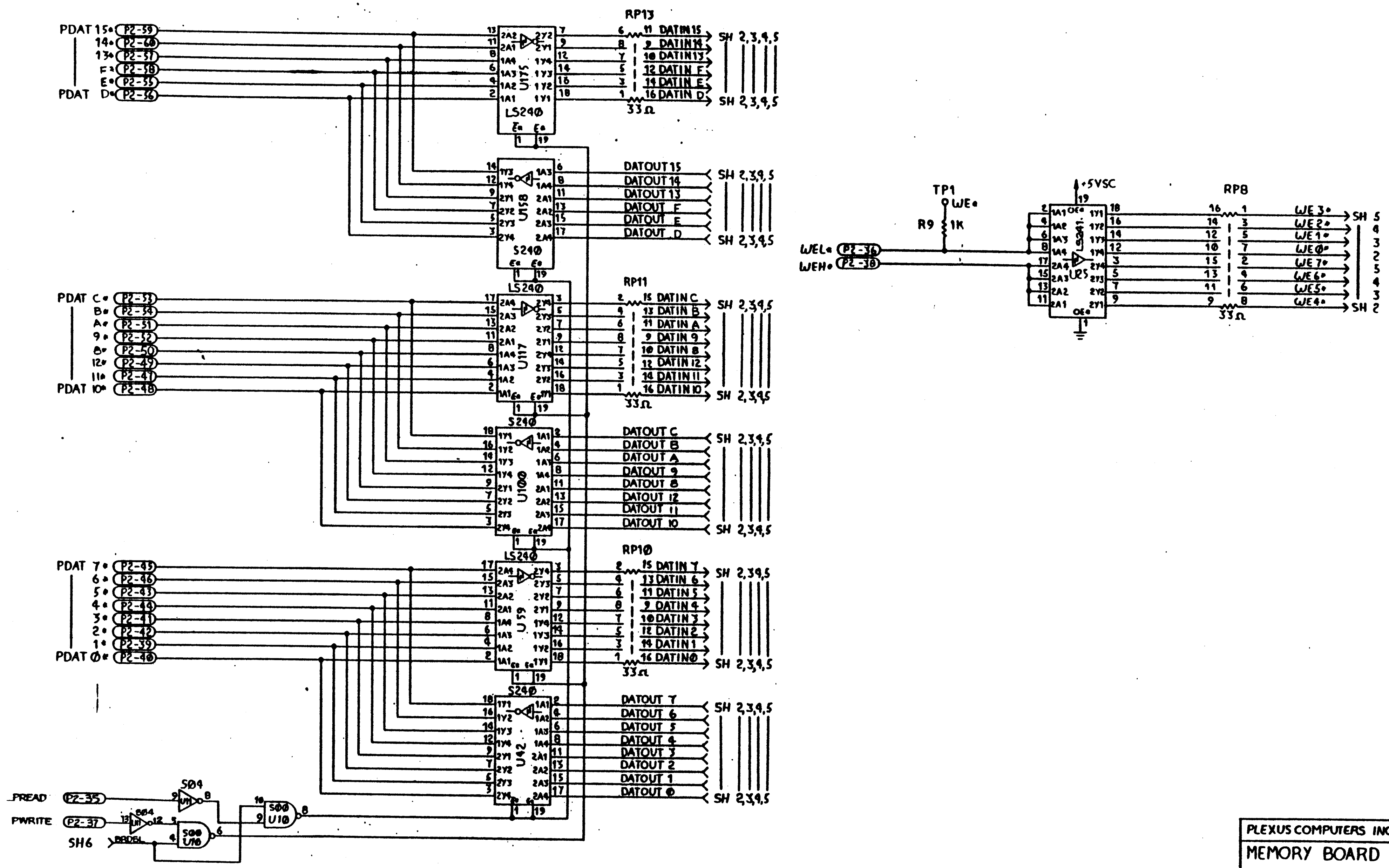
SIZE		SCALE —	SH 5 OF 7	REV
D	60-00087			A



PLEXUS COMPUTERS INC.

MEMORY BOARD 16K

SIZE	SCALE —	SHEET 1 OF 7	REV
D	60-00087		A



PLEXUS COMPUTERS INC.

MEMORY BOARD

16K

SIZE	SCALE	SH	REV
D	.60-00087	7 OF 7	A

BACKPLANE

1. MULTIBUS BACKPLANE AND INTERFACE

IEEE Proposed Microcomputer System Bus Standard (P796 Bus)

Schematics

1. MULTIBUS BACKPLANE AND INTERFACE

Most circuitry in the P/25 is located on cards that plug into a Multibus backplane. The backplane has 6 slots. It contains a processor board, one or two memory boards, one or two ICPs and an IMSC disk and tape controller. The processor and memory boards, which use connector P2, must be installed in slots 1 through 3.

A typical system may be configured as follows:

Slot	Contains
1	First Memory Board
2	Second Memory Board (Optional)
3	Processor
4	First ICP
5	Second ICP (Optional)
6	IMSC Disk and Tape Controller

Connector Pl contains the standard Multibus signals and power connectors. All connections to the system memory array are on connector P2.

The backplane also contains a bus priority arbitrator and a clock generator. The priority resolver arbitrates Multibus requests from bus masters using the parallel resolving method. It allows up to four bus masters on the backplane. The clock generator is an oscillator and driver that generates the 10-Mhz bus clock (BCLK/) used by all bus masters to synchronize their arbiters.

1.1. Multibus Interface

This section describes the Multibus interface used in the Plexus P/25 computer. Use the IEEE Proposed Microcomputer System Bus Standard (P796 Bus) in conjunction with this section for a complete description of the P/25 Multibus.

The standard modules which are connected to the Multibus in the P/25 are: a) the processor board, b) disk and tape controller, c) one or two serial I/O controllers and d) one or two memory array boards. Other user-optional Multibus I/O controllers may be added. The general Multibus interface is discussed below. Any restrictions specific to each module are noted.

1.1.1. Multibus Elements

(Refer to the P796 Standard, sections 2.1.1 - 2.1.2.)

In the P/25, the CPU acts as a bus master, the disk and tape controller and the ICP act as both bus masters and slaves, and the MMU acts as a bus slave. Other I/O controllers act as either bus masters and/or slaves.

NOTE: The MMU is physically located on the CPU (processor) board.

1.1.2. Multibus Signal Classes

(Refer to the P796 Standard, section 2.1.3 and section 3)

Most of the Multibus signal lines used by the P/25 are used as defined in the P796 Standard, section 2.1.3, and have the timing and electrical characteristics as defined in the P796 Standard, section 3.

1.1.2.1. Connector P1

The following table shows and describes the connector P1 Multibus signals. Those which are not identical to the pins in the P796 Standard are noted.

Control Lines

Signal Name	Symbol	Signal Generated by	Signal Used by
Bus Clock	BCLK/	MB Backplane	Any MB Module
Mem Write Cmd	MWTC/	Current Bus Master[1]	MB Memory Slave
Mem Read Cmd	MRDC/	Current Bus Master[1]	MB Memory Slave
I/O Write Cmd	IOWC/	Current Bus Master	Current MB Slave
I/O Read Cmd	IORC/	Current Bus Master	Current MB Slave
Transfer Ack Line[2]	XACK/	Current Bus Slave[3]	Current Bus Master
Initialize	INIT/	Power Supply Reset Button CPU	Any MB Module Any MB Module Any MB Module
Advanced Ack Line[4]	AACK/	MMU	Current Bus Master

Address and Inhibit Lines

Signal Name	Symbol	Signal Generated by	Signal Used by
Address Lines	ADR0/- ADR13/	Current Bus Master	Any MB Slaves
Inhibit 1 Line[2]	INH1/	Not Used	
Inhibit 2 Line[2]	INH2/	Not Used	
Byte High Enable	BHEN/	Current Bus Master	Bus Memory Slave
Constant Clock	CCLK/	Not Used	

Data Lines

Signal Name	Symbol	Signal Generated by	Signal Used by
Data Lines	DAT0/- DATF/	if write: Current Bus Master	Current Bus Slave
		if read: Current Bus Slave	Current Bus Master

[1]Multibus memory commands are not supported for the CPU bus master.

[2]XACK/ does not meet the P796 Standard requirement for INH1/ and INH2/ signal timing since these signals are not used by the P/25.

[3]If a timeout occurs, XACK/ must be generated by the Current Bus Master.

[4]Not a P796 Standard standard signal. Electrical characteristics same as XACK/.

Interrupt Lines

Signal Name	Symbol	Signal Generated by	Signal Used by
Interrupt Requests[5]	INT0/-INT7/	Any Bus Module	CPU Master
Interrupt Ack[5]	INTA/	Not Used	

Bus Exchange Lines

Signal Name	Symbol	Signal Generated by	Signal Used by
Bus Request	BREQ/	Any Bus Master	MB Backplane
Bus Priority IN	BPRN/	MB Backplane	Highest Requesting Bus Master
Bus Priority OUT[6]	BPRO/	Not Used	
Bus Busy	BUSY/	Current Bus Master	Any Non-Current Bus Master
Common Bus Request	CBRQ/	Not Used	

[5] Only non-bus-vectored interrupts are supported. (Refer to the P796 Standard, section 2.3).
[6] Daisy-chain priority schemes are not supported.

Power Fail/Restart Lines

Signal Name	Symbol	Signal Generated by	Signal Used by
AC Low	ACLO/	Not Used	
Memory Protect	MPRO/	Not Used	
Power Fail Int[7]	PFIN/	Power Supply	Any MB Module
Power Fail Sense	PFSN/	Not Used	
Power Fail Sense Reset	PFSR/	Not Used	

[7]This signal appears on a nonstandard pin (P1, pin 78).

1.1.2.2. Connector P2

Connector P2 connects the processor board to system memory. These pins are unique to the P/25; they are not Multibus standard. Pinouts for connector P2 are as follows:

> NOTE: The letter P in front of a signal name indicates that it goes to connector P2. For example, DAT0 on P1 is similar to PDAT0 on P2.

1	PADR10	31	Ground
2	PADRF	32	Ground
3	PADRE	33	not used
4	PADR7	34	not used
5	PADRD	35	PREAD
6	PADR5	36	WEL/
7	PADRC	37	PWRITE
8	PADR6	38	WEH/
9	PADRB	39	PDAT/
10	PADR4	40	PDAT0
11	PADRA	41	PDAT3
12	PADR3	42	PDAT2
13	PADR9	43	PDAT5
14	PADR0	44	PDAT5
15	PADR8	45	PDAT7
16	PADR1	46	PDAT6
17	Ground	47	PDAT11
18	PADR2	48	PDAT10
19	PADR11	49	PDAT2
20	Ground	50	PDAT8
21	PADR12	51	PDATA
22	Ground	52	PDAT9
23	PMREQ	53	PDATC
24	PMREQ	54	PDATB
25	PADR15	55	PDATE
26	PADR14	56	PDATD
27	RFSH	57	PDAT13
28	RFSH	58	PDATF
29	Ground	59	PDAT15
30	PADR13	60	PDAT14

1.1.3. Data Transfer Operations

(Refer to the _P796 Standard_, section 2.2)

P/25 data transfer operations work as described in the _P796 Standard_, section 2.2. The exception is that the inhibit operations (INH1/ and INH2/ -- _P796 Standard_, section 2.2.2.7) are not implemented.

1.1.4. Interrupt Operations

(Refer to the _P796 Standard_, section 2.3)

Only non-bus-vectored interrupts are supported.

1.1.5. Multibus Exchange

(Refer to the _P796 Standard_, section 2.4)

The P/25 bus exchange scheme is a subset of the parallel arbitration technique described in the _P796 Standard_, section 2.4.2.2.

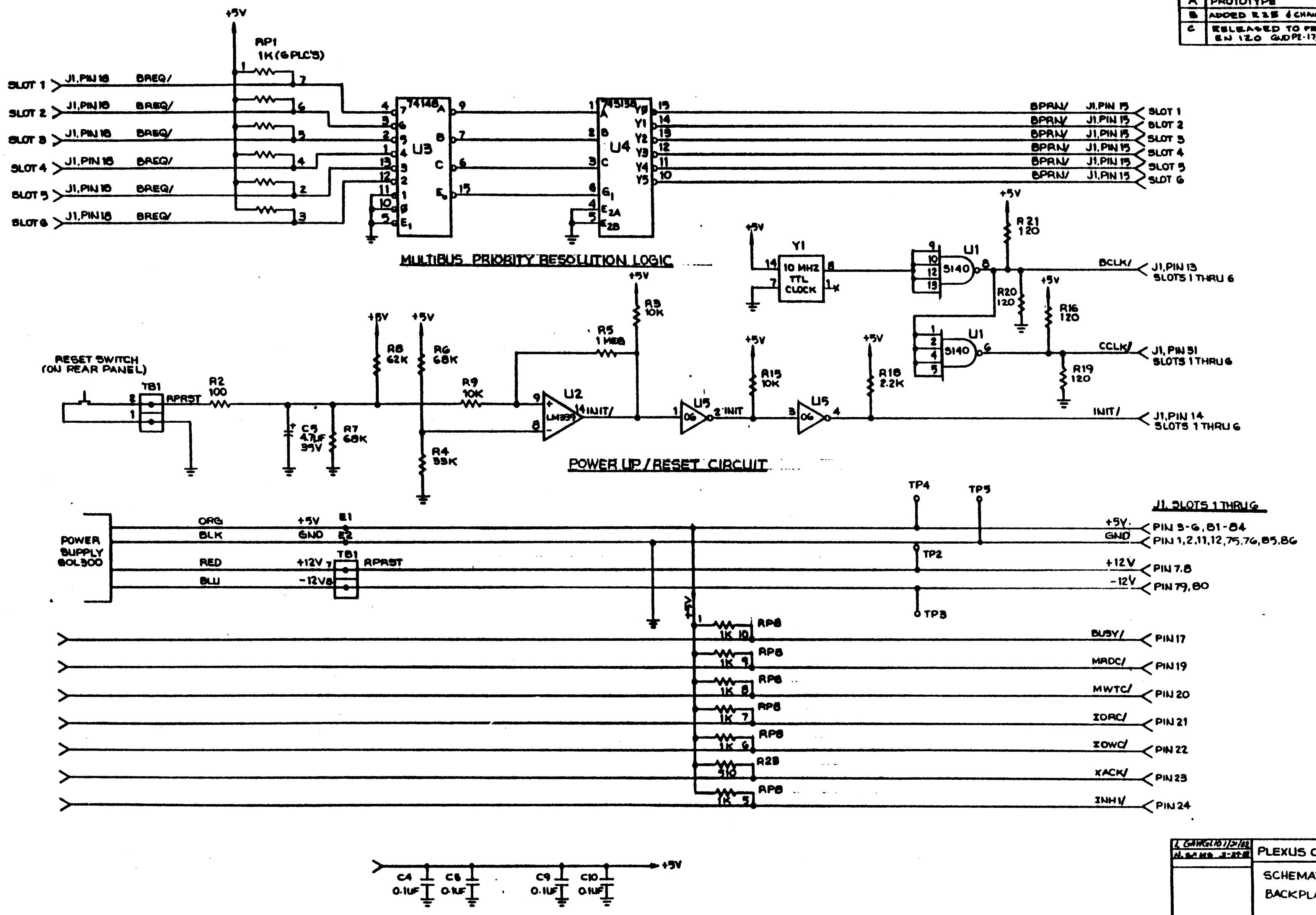
Once a master has been granted control of the bus (BPRN/), it maintains control by holding the busy line (BUSY/) active. Each bus master is only allowed to control the Multibus for 100 microseconds; after this, it must relinquish the bus by deactivating BUSY/. This time period allows transfers of 64 bytes maximum.

NOTE: The IMSC may be allowed to hold the Multibus longer if this can be implemented without degrading system performance.

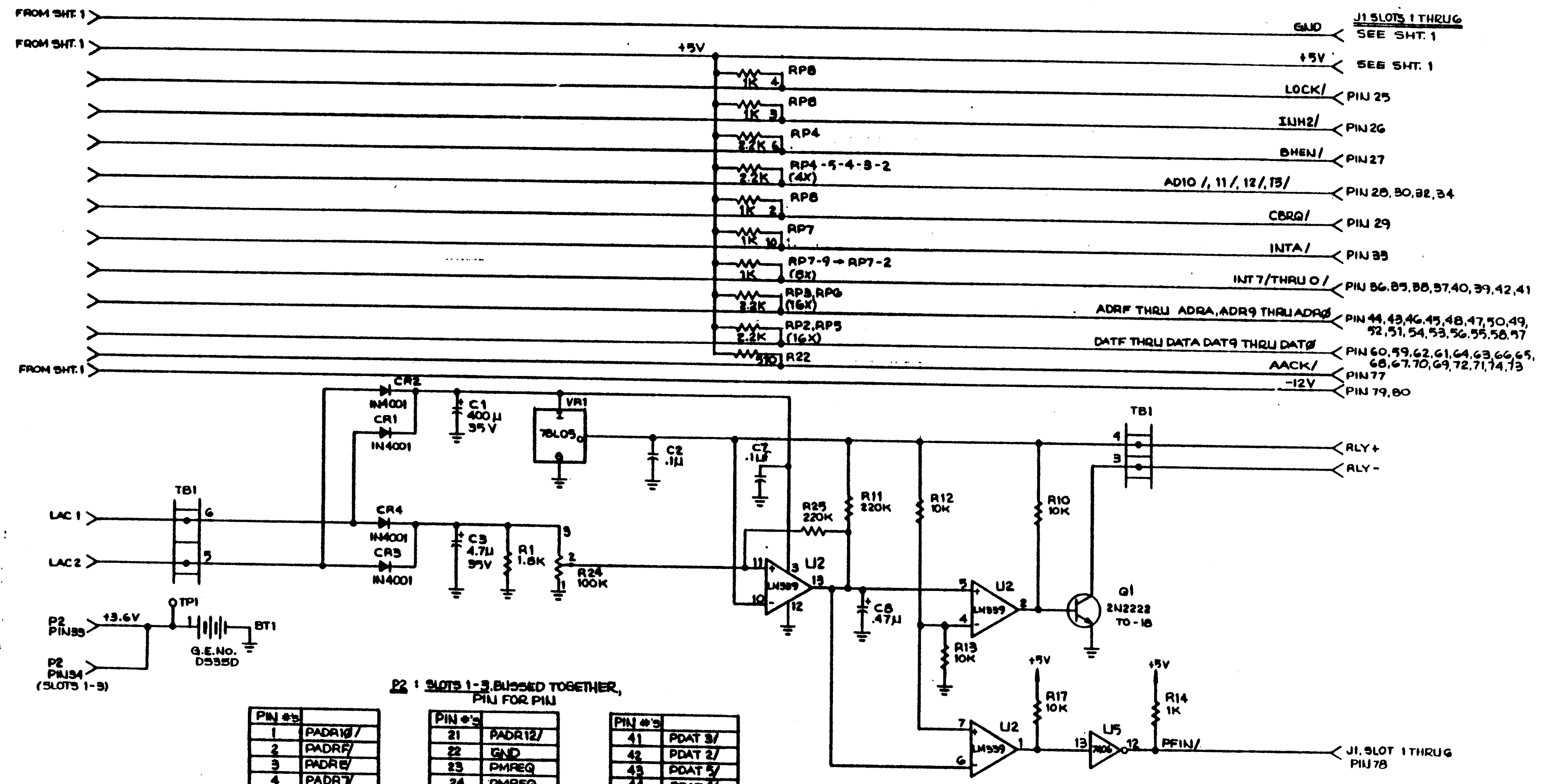
Bus Priority Out (BPRO/) and Common Bus Request (CBRQ/) are not used by the P/25 but may be used by user-supplied I/O controllers.

More detailed information about the P/25 Multibus is contained in the _IEEE Proposed Microcomputer System Bus Standard_ (_P796 Bus_) which is provided in this section of this manual.

REV	DESCRIPTION	DATE	APPV
A	PROTOTYPE		
B	ADDED R25 & CHANGED C-8	8-20-82	
C	RELEASED TO PRODUCTION EN 120 GND P2-17,20,22,23&24	3-29-82 4-7-82	[illegible]



L. GANGL(101) 5/1/82	PLEXUS COMPUTERS INC.		
N. [illegible] 3-29-82	SCHEMATIC: BACKPLANE, 6 SLOT, P-25		
SIZE D	60-00086		REV C
	SCALE: ~	SHT: 1 OF 2	



P2 : SLOTS 1-3, BUSSED TOGETHER, PIN FOR PIN

PIN #'s	
1	PADR1Ø/
2	PADRF/
3	PADRE/
4	PADR7/
5	PADRD/
6	PADR5/
7	PADRC/
8	PADR6/
9	PADRB/
10	PADR4/
11	PADRA/
12	PADR3/
13	PADR9/
14	PADRØ/
15	PADR8/
16	PADR1/
17	GND
18	PADR2/
19	PADR11/
20	GND

PIN #'s	
21	PADR12/
22	GND
23	PMREQ
24	PMREQ
25	PADR15/
26	PADR14/
27	RFSH
28	RFSH
29	GND
30	PADR13/
31	GND
32	GND
33	BATTERY+3.6V
34	BATTERY+3.6V
35	N/C
36	WEL/
37	N/C
38	WEH/
39	PDAT 1
40	PDATØ/

PIN #'s	
41	PDAT 3/
42	PDAT 2/
43	PDAT 5/
44	PDAT 4/
45	PDAT 7/
46	PDAT 6/
47	PDAT 9/
48	PDAT 8/
49	PDAT A/
50	PDAT B/
51	PDAT D/
52	PDAT C/
53	PDAT F/
54	PDAT E/
55	PDAT 11/
56	PDAT 10/
57	PDAT 13/
58	PDAT 12/
59	PDAT 15/
60	PDAT 14/

PLEXUS COMPUTERS INC.

SCHEMATIC :
BACKPLANE, 6 SLOT, P-25

SIZE		SCALE	SHT	REV
D	60-00086	~	2 OF 2	C

DISK AND TAPE CONTROLLER

1. IMSC OVERVIEW

The intelligent mass storage controller (IMSC) is a double-height Multibus card that controls interaction between the P/25 CPU and MMU and the disk and tape drives.

The IMSC communicates with the CPU and MMU via the Multibus. The CPU provides instructions in the form of command and status blocks, and the MMU is used to access memory. Disk storage is accessed over an industry-standard storage module disk (SMD) interface, and the tape drive is accessed via eight data lines and eight control lines.

The IMSC is equipped with a Zilog Z8001 microprocessor, 128K bytes of local memory and DMA. It normally controls one disk drive and one tape drive (P/25 standard configuration); however it is capable of controlling three additional disk drives.

The CPU controls the IMSC in almost the same way as it controls the ICPs. It builds a command block in the IMSC's wakeup address space, then interrupts the IMSC Z8001 with a channel attention. The IMSC receives and acknowledges the channel attention, then reads the command block, which contains its instructions.

The 128K bytes of local RAM is organized as five I/O data buffers, each capable of storing an entire disk track. These eliminate the slowdowns caused when two or more processes need constant access to disk. Instead of reading information from the disk one block at a time, the IMSC can read an entire track and store it in the buffers. When the process is ready for the next block, the buffer provides it. The process appears to have the entire disk to itself.

The P/25 uses an Archive Intelligent Tape Drive, which performs many of the functions normally handled by the controller. This simplifies the IMSC's dealings with the tape drive.

The functional blocks of the IMSC board are shown in Figure 1.

1.1. References

For a complete description of the disk drive operation, see the _Plexus Winchester Disk Manual_ for the required disk size. For a description of the tape drive operation, see the _Plexus Cartridge Tape Drive Manual_.

2. PROCESSOR

The IMSC is equipped with a Zilog Z8001 microprocessor which runs at 4 Mhz. The Z8001 runs in system and normal mode, and uses only standard I/O space (special I/O space isnot used). It is described in the Zilog Z8000 CPU Technical Manual.

The Z8001 uses five of its 64 memory segments. Segments 0 and 17 access the local PROM bank. Segment 3 accesses the low-order local RAM, and segment 5 accesses the high. Segment 9 is mapped into the Multibus memory; it is used for performing block moves between local RAM and the system RAM array.

The Z8001 uses all three of its interrupt capabilities; vectored, nonvectored and nonmaskable interrupts. These are described in Disk and Tape Controller, section 6.

The Z8001 is helped by two other onboard processors -- a Z80 DMA chip and the disk sequencer. The DMA chip handles many of the details of passing data to and from the tape drive with minimum participation by the Z8001, and the disk sequencer performs a similar function for the disk drive(s).

2.1. Bus Structure

The IMSC processor address/data pins (A0 through A15) are buffered by U88 through U91. These latch addresses to the LADR bus and data to and from the B'AD bus. These two busses go throughout the board, often latching to the other buses. The IMSC bus structure is described in Disk and Tape Controller, section 8.

2.2. Local Ports

The Z8001 makes extensive use of local ports to communicate with the various local peripheral devices. This is necessary to support the concurrent operations that it must perform constantly; it downloads instructions and expects the addressed device to implement them. It also reads status and makes decisions based on the status and/or passes the status to the system CPU.

To write to the local I/O ports, the Z8001 places an address on the address bus (LA00 to LA15). This causes the addressed port to input data from the B'AD bus. The Z8001 does this sequentially: it activates the address strobe (AS) to latch the address onto the address bus, then it activates its data strobe and R/W line to latch in the data. The R/W line is inverted into B'W/R to latch the data onto the XCVR bus when necessary.

The addresses, data, and functions are shown in the following table:

Port	Signal Description	Bits	Bit Description
00	Load disk sequencer memory address counter	0 - 14	RAM address bits
		15	RAM segment select bit
02	Host memory offset register	0 - 3	Host memory address bits
		4	Reserved. Must be HIGH
04	LED/interrupt/parity control	0 - 6	LED bits. 0 = ON
		7	1 = Enable parity
		8	1 = generate interrupt to host
		9	0 = parity clear
06	Load disk sequencer byte register	0 - 7	Byte count
10	Load utility register	0	Burst error processor (BEP) control request 0 = sequencer 1 = Z8001
		1	1 = enable memory holding register
		2 - 4	C0 to C2 of BEP
		5	Disk read/write bit 0=Write to memory 1=Read from memory
		6 - 7	Clock select (binary)
		8	Disk select tag
		9	Tag 1
		10	Tag 2
		11	Tag 3

Source BITCLK	SOCLK
00=Disk	125CLK
01=Z8001	Z8001
10=External	External
11=125CLK	125CLK

		12	Tape in read mode
		13	Tape in write mode
		14	Tape on-line (Active LOW)
		15	Disk data path MUX 1=From sequencer 0=From Z8001
20	Z8001 to sequencer single-step		
22	Sequencer reset		
24	Load sequencer command	0 - 2	Command code
		3	1K-byte sector select
		4	Reserved
		5	Command acknowledge preset by Z8001; cleared by sequencer
26	Load sequencer ID and counter	0 - 7	Iteration counter
		8 -15	Sector ID
30	Disk select register	0 - 9	Disk data bus
		10	Reserved
		11-14	Disk drive select
		15	Tape reset
40	Z8001 to BEP clock		
50	BEP read port	0	BEP error bit
		1	BEP EP bit
		2	BEP AE bit
		3	Unused
		4 - 7	BEP LP 0 to 3
		8 -15	BEP OUT (0 to 7)
60	Tape data bus write	0 - 7	Data
70	Reset tape request		
80	Generate Z80 RETI signal	0 - 7	Store with 0ED and 4D (hex)
90	Read board address strap	0 - 7	DIP switches (ON=0)
		8 -15	Jumpers
A0	Diagnostic switches and sequencer counter	0 - 7	Diagnostic switches
		8 -13	Iteration counter

14	Parity error bit (Active LOW) (0=not power fail)
15	Tape exception (Active LOW)

3. LOCAL MEMORY

3.1. Memory Organization

The IMSC board includes 8K words of PROM, 128K bytes of dynamic RAM, and 64K bytes of memory mapped into the Multibus memory space. The PROM occupies segments 0 and 17 of the IMSC Z8001 address space. The low half of the dynamic RAM (64K bytes) occupies segment 3, and the high half occupies segment 5. The 64K bytes of Multibus address space occupy segment 9.

The PROM memory is used to store diagnostic and bootstrap programs for the disk and tape drives. The dynamic RAM is used to buffer information from the disk and tape drives before passing it on to the system memory, and to store large blocks of Unix disk programs. The Multibus memory space is used to pass information over the Multibus to and from system memory.

3.2. Local Dynamic RAM Access

The 128K bytes of dynamic RAM on board is shown on page 2 of the IMSC schematics.

The RAM chips (U1 through U8 and U21 through U28) are 64K-by-1 bits wide, and are wired for parallel operation. The 16 bits of any one word are stored at identical addresses in all 16 chips. The MEMBUS has one line attached to the input and output of each RAM chip. The addresses appear on the multiplexed address bus (MADR 0 to 7), and are latched into memory by the proper combination of B'RAS/ 1 and 2, CAS EVEN and ODD/, and B'RW/ 1 and 2.

The RAS/CAS operation is typical. When CAS is enabled, the data on the MADR bus is interpreted as a column address; when RAS is enabled, it is interpreted as a row address. Each column and row address enables a single bit in all 16 chips.

The CASEVEN/ and CASODD/ lines can be used to select a byte-only access to either the high- or low-order byte of a word. The CAS line is deactivated for the unused byte.

Signals B'RW 1 and B'RW 2 control the direction of data flow for the high- and low-order bytes. Provided with a row and column address, and a direction of data flow, each RAM chip outputs (or inputs) its one bit of data on (or from) the MEMBUS. Data from the MEMBUS is latched onto the B'AD bus by a combination of paks U45, U46 and U48.

NOTE: The 4164 RAM chips are able to operate properly with their inputs wired to the outputs because they sense the proper direction of data flow and tristate their gates when necessary (during input operations).

To arrive on the MADR bus, the multiplexed address bits first must be latched from the Z8001 outputs to the address bus by U88 and U89. Since this is a 16-bit data path and the CAS and RAS addresses are only 8 bits each, they must be multiplexed by U32 and U33.

3.2.1. Parity

The parity circuit is shown on page 4 of the IMSC schematics.

U9 and U29 are 4164 RAM chips wired into the memory circuit in the same manner as U1 through U8 and U20 through U28. The RAS, CAS and B'RW signals perform exactly the same as in the RAM circuit. The only difference is that the data inputs and outputs of U9 and U29 are connected to the EVEN outputs of U44 and U50, which are parity generation chips.

During write operations, the data on the B'AD bus is fed into the parity generation chips, and the EVEN outputs are stored in U9 and U29. During read operations, the data on the B'AD bus is again fed into the parity generation chips, which produce an EVEN output. This is compared with the output of U9 and U29 by an exclusive OR (U10). Any difference produces an output, which activates PAR'ERR and NMI/.

3.3. Memory Arbitration

The IMSC processor, the disk sequencer, and the tape DMA all compete for access to local RAM. This is not a direct, one-on-one competition. The disk sequencer and the IMSC processor compete with each other, and the tape DMA fills in for the IMSC processor when it is active.

The IMSC processor and the disk sequencer access local RAM by taking control of the MADR bus, and the memory control lines (CAS, RAS and WE). The final stage in the contention is U97, which forces the IMSC processor into a wait state when the disk sequencer is in control. When both devices require access, the cycles are interleaved; first one then the other is granted a memory cycle.

The tape DMA looks exactly like the IMSC processor when it is active. It performs a formal DMA request and forces the IMSC processor to give up control by tristating its outputs. It then mimics the IMSC processor by generating and receiving exactly the same signals.

3.4. Local PROM

The local PROMs occupy segments 0 and 17 in the IMSC Z8001 address space. They contain diagnostic and bootstrap programs for the disk and tape drives.

The local PROM circuit consists of four 4K-by-8 PROMs (U54 and U57, and U71 and U72), an 8-bit latch for each PROM, and the ENROMBK0/ line. The PROM circuit contains 8K words (16K bytes).

If the bits on LA01 to LA12 match an address in the PROM circuit, both PROMs in the set output their data into the buffers, which must be enabled by ENROMBK0/. When the latches are enabled, they latch data onto the B'AD bus. Because all PROM accesses are for entire words, bit LA00 is not used.

The IMSC processor selects between segments 0 and 17 with its SN 0 to SN 5 lines. These are read as a binary number; SN 0 alone accesses segment 0 and SN 0 with SN 4 accesses segment 17. This logic is shown in the IMSC schematics, page 5.

Because the PROMs and latches are 8-bit devices, two PROM and latch circuits are required for each set. Both PROMS receive the same address, and both latches are enabled by the same signal (ENROMBK0/).

This circuit is shown in the IMSC schematics on pages 5 and 22.

4. MULTIBUS INTERFACE

The IMSC performs all its interaction with the rest of the system over the Multibus. It can act as both a bus master and a bus slave. The B'AD bus and the address bus (LADR 0 to 15) both connect to the Multibus. This circuit is shown on page 18 of the IMSC schematics.

4.1. Bus Arbitration

The IMSC competes for the Multibus exactly like any other Multibus master in the P/25. When it requires the Multibus, it enables a CTC output to generate a bus request every millisecond.

After the bus request is activated, two steps must be completed before the IMSC can receive the Multibus.

First, the bus request enters the bus priority logic circuit on the backplane. This circuit returns BPRN, based on the IMSC's bus priority.

Once the IMSC receives BPRN, it polls its BUSY line to see if any other master is on the bus. As soon as BUSY goes inactive, it pulls BUSY high to keep other bus masters off, then proceeds with its bus transaction.

The amount of time any master can keep the Multibus is controlled by a timing circuit in the master. In this case, an MRES instruction releases the Multibus by releasing BUSREQ, which also causes it to release BUSY. The IMSC is allowed to keep the Multibus long enough to transfer 256 bytes (400 microseconds).

The instructions to set this time are downloaded into the CTC chip by software.

4.2. Data Transfer

Data is passed between Multibus lines DAT0 to DAT15 and the B'AD bus by U14 and U15, which are bidirectional transceivers. When the output of U140 pin 8 and U20 pin 8 both go low, data flows from the Multibus to the B'AD bus. When U20 pin 8, which is an inverted version of B'R/W, goes high, data flows from the B'AD bus onto the Multibus. Any other combination causes both directions to assume the high impedance state.

The low-order byte of the Multibus data lines (DAT0 to DAT7) is also latched into U34, which is enabled by LD MOFF/. U34 latches this data into U12, which outputs BA'D00 to B'AD04 onto the Multibus as BHEN and AD10H to AD13H. U12 is enabled by the output of U140 mentioned above.

The low-order data byte from the Z8001 (DAT00 through DAT07) can also be latched onto the B'AD bus by the combination of U13 and U35. This transfer is enabled by the CEMULTIBUS signal from U144.

4.3. Address Transfer

The address bus lines (LA00 to LA15) are latched onto the Multibus by U16 and U17, which are enabled by U140 pin 8. These addresses access system memory through the MMU (located on the processor board).

When the CPU wants to communicate with the IMSC, it places the IMSC's wakeup address on ADD00 to ADD07. These pins feed into U37 and U38, which are 4-bit comparators. This byte is compared with the levels set by SW2, which defines the IMSC's wakeup address. If they are identical, U37 generates an output which is ANDed with IOWC/ in U96 and used to start the channel attention sequence.

5. IMSC CONTROL

5.1. CPU Controlling IMSC

The CPU controls the IMSC the same way it controls the ICPs; it builds a command block in its memory at the IMSC's wakeup/channel attention address, then issues a channel attention to the IMSC, which reads the command block and executes the instructions there.

Typically, the command block contains a pointer to another block of data, and instructions for handling that data. For example, the command block might say (in data structures) "go to memory location 50,128H and write the next 40 blocks onto the disk".

The following is an example of a CPU-to-IMSC command sequence:

> NOTE: This example describes a memory-to-disk write mentioned in the preceding paragraph.

a) The CPU writes the data in system memory and puts a pointer and a command block in the IMSC's wakeup address.

b) The CPU issues a channel attention to the IMSC.

c) This causes a local vectored interrupt to the IMSC processor.

d) The IMSC requests and (eventually) receives the Multibus. The signal used to request the bus is 796'BUSREQ. The IMSC is notified that it has the bus when it receives 796'BPRN.

e) The IMSC reads the contents of the command block into IMSC local memory, then releases the Multibus.

f) The IMSC processor executes the instructions in the command block. In this case, the block orders the IMSC to write a block from memory to disk.

g) The IMSC again requests the Multibus through the same procedure as before. It writes the first chunk of the data block in its buffer, releases the bus, and repeats the bus request sequence. When the bus is granted, it fetches the next chunk of data. It continues doing this until the entire data block, as defined in the command block, is in the IMSC local memory.

h) Once all the data is received, the IMSC requests the Multibus one last time and writes a 'command status' block.

It then asserts its 796'INT line, which generates an interrupt to the CPU processor. The CPU responds by reading the command status block.

5.2. IMSC Controlling Disk

The IMSC controls the disk drive(s) through two paths, the disk sequencer and the command bus. The commands passed over each of these paths are listed below:

Sequencer Commands

Write Data
Format
Read Data
Read ID
Idle Loop
Diagnostic

Command Bus Commands

Drive Select
Seek
Clear Faults
Move Data Strobe Early
Move Data Strobe Late

Both command paths are described in the following sections.

5.2.1. Disk Sequencer

The disk sequencer is shown on page 12 of the IMSC schematics. It converts commands from the IMSC processor to a stream of output bytes that implement the commands.

The disk sequencer is activated when the command latch (U131 and 132) latches data from B'AD lines 0, 1, and 2 onto pins A8, A9 and A10 of the sequencer PROMs. These three bits define one of the six disk sequencer commands.

To implement each of its six commands, the sequencer must step through up to 128 instructions. Pins A8 through A10 initialize a command sequence, and the PROMs output the instructions.

The PROMs are divided into two banks, each containing two 2K-by-4 PROMs. The first bank is the state PROM bank (U151 and U152), which recognizes the command on A8 through A10. It outputs an address on its output pins. These are latched by U150 onto pins A0 through A6 of the data PROM bank (U153 and U154 -- PROMs are each 1K by 4), which outputs an instruction on SQDATA. SQDATA then accesses the next byte in the state PROM bank, which outputs the next address. This sequence continues until the command is implemented.

The data byte output on SQDATA is divided and passed to several places on the IMSC board. These are not coherent bytes to be read and processed by any one device; they are latched off SQDATA and immediately go their separate ways. The following paragraphs describe some of the uses of this data.

SQDATA 0 through 7 are latched sequentially from U136 to U117 every time DKWDLD* pulses. The output of U117 provides the various disk control (DKCTL) signals

U134 and U155 are a word up-counter which use SQDATA to count the number of words in a disk transaction. The sequencer first provides it with a word count. When it reaches that count, it activates SQCNTOVF. The word count comes over SQDATA.

U137 latches SQDATA 0 through 4 onto SYN SQDATA every time SQCLK pulses.

5.2.2. Disk Command Bus

The disk command bus pins are shown in the IMSC schematics on page 15. These are BUS 0 through BUS 10, TAG 1, TAG 2, TAG 3 and SELECT TAG. The commands that are sent over these pins are listed earlier in this section.

5.2.2.1. Implementation

The command bus commands are implemented by writing data on the command bus and toggling the TAG lines. The command data is latched off the B'AD bus by U168 and U171. These outputs activate the command bus output chips (U188, U189, U190 and U191). The TAG lines are activated by U192.

The command sequence is as follows:

Select -- Write the disk drive number to the command bus and activate SELECT TAG.

Seek -- Write the cylinder address to the command bus and activate TAG 1. Write the head address to the command bus and activate TAG 2.

Clear Fault, Return to Zero, Move Data Strobe -- Write the appropriate bits to the command bus and activate TAG 3.

NOTE: The proper bit patterns for the command bus are provided in the disk drive manual.

5.3. ID Equate Latch

The desired disk sector is read from B'AD 8 through 15. These bits are latched by U114 and U115 and presented at the "A" inputs of U116, which is an 8-bit comparator. The disk reads sector IDs as they go by, and these appear on the "B" inputs. When the inputs match, the correct sector has been reached, and the comparator generates DKIDEQ* to inform U135 that the proper sector has been identified.

5.4. Serial-to-Parallel Conversion

During disk write operations, memory words from the MEMBUS are stored in U63 and U65 (one byte each) until the serial-to-parallel conversion circuit (see the IMSC schematics, page 10) is ready. The word is latched onto the SERBUS, which appears on the pins of U82 and U83. These convert it to a serial bit-stream (SERDES).

During disk read operations, the serial bit-stream from the disk (DKRDATA) is fed into U82 and U83, which convert it into parallel words. These are output on the SERBUS, and latched onto the MEMBUS by U62 and U66. They can also be multiplexed onto SERHR one byte at a time by U102 and U105. SERHR connects to the disk ID EQUATE circuit and the ECC processor.

5.5. Disk Interface

The connections from the IMSC to the disk(s) are shown in the IMSC schematics, pages 13 and 14.

Each disk has two cables connecting it to the IMSC. Connector J1 is 60 pins wide. It connects to all disks in series. Any signal which must go to all disk drives is on J1.

J2 and J3 are each divided so as to connect to two individual disks each; J3 connects to disk 1 and 2, and J2 connects to disks 3 and 4. Any signal which must go only to one particular disk is on J2 or J3.

5.5.1. Differential Drivers

The IMSC uses differential signals to communicate to the disks. The output chips divide each signal into a positive and a negative component which is transmitted on two lines. The receivers compare both components and translate the result back into a single signal. This arrangement protects against transmission errors and signal decay.

5.5.2. Inputs

The disk data receivers (U176 through U179) each receive the READ DATA, READ CLK, and PLO CLK inputs from one disk. These chips are enabled by the disk select (DRSELD 1 through 4) lines. The disk select signals are generated by U175, which receives the SELECTED signals from the disks. This arrangement ensures that only a selected disk can have its outputs read.

5.5.3. Outputs

The bit-stream to be written on the disk (WRDATA 1 through 4) and the write clocks (WRCLK 1 through 4) are generated by U193 and U194. These chips make no distinction about which disk they are transmitting to; their inputs are all common and their outputs are identical on all pins. U193 converts the bit-stream from U82 (the parallel-to-serial converter) into the differential signal for transmission to the disk drives and outputs it on all pins. U194 does the same with BITCLK. These signals appear at all disk drives; only the selected drive inputs it.

5.6. IMSC Controlling Tape Drive

The IMSC is connected to the intelligent tape drive by 16 pins in a 50-pin ribbon cable. Eight of these transmit data to and from the tape drive. The other eight are control lines, used by the IMSC to control the tape drive and by the tape drive to communicate with and acknowledge the IMSC.

5.6.1. Tape Data Bus

The tape data bus (pins HB0 through HB7) is buffered by U184, which is bidirectional. The direction of data through U184 is controlled by TPBON* which is derived from the tape drive DIR output. The tape drive asserts DIR to cause the IMSC to output data on the tape data bus. When DIR is low, data flows from the tape drive to the B'AD bus through U184.

The tape data bus also transmits command data from the IMSC Z8001, and status data from the tape drive to the IMSC. When the tape drive is in COMMAND MODE, the REQ* line informs the tape drive that valid command data is on the tape data bus. When the tape drive is in the STATUS INPUT mode, the controller asserts REQ to inform the tape drive that it has received status data placed on the tape data bus by the tape drive unit.

5.6.2. Control Lines

The following table shows the tape drive to IMSC control lines and provides a brief description of their functions:

IMSC to Tape Drive

Signal Identifier		Signal Name	Description
RST	--	RESET	Causes the tape drive to reset. This is reserved for system resets or powerup.
XFER	--	TRANSFER	Informs the tape drive that data has been placed on the tape data bus during WRITE mode and that data has been read from the tape data bus during READ mode.
REQ	--	REQUEST	Informs the tape drive that command data has been placed on the tape data bus during COMMAND mode and that status data has been read from the tape data bus during STATUS INPUT mode.
ONL	--	ON-LINE	Activated prior to transferring a READ or WRITE command and deactivated to terminate the command.

Tape Drive to IMSC

Signal Identifier		Signal Name	Description
DIR	--	DIRECTION	Controls the direction of data on the tape data bus. When inactive, causes the IMSC to output data on the tape data bus. When active, causes the tape drive to output data.
ACK	--	ACKNOWLEDGE	Informs the IMSC that data has been taken from the tape data bus during WRITE mode and that data has been placed on the tape data bus during READ mode.
RDY	--	READY	Informs the IMSC that data has been taken from the tape data bus during COMMAND mode

		and that data has been placed on the tape data bus during STATUS INPUT mode.
EXC --	EXCEPTION	Informs the IMSC that an exception exists in the tape drive unit, and that the IMSC must issue a STATUS command and perform a status input to determine the cause of the exception.

As shown in the above table, the control line used and the meaning of data on the tape data bus is a function of the tape drive mode.

5.7. Tape DMA

Tape-to-memory data transfers are handled by U107, a Zilog Z80 DMA chip described in the Zilog Z80 DMA Technical Manual. This chip is shown on page 8 of the IMSC schematics.

Because most of the pins on the Z80 DMA chip are bidirectional, control of this chip is somewhat complicated. The descriptions in this section are divided by functions according to what the DMA chip is required to do.

There are three basic functions that the Z80 DMA chip is required to perform: a) read instructions from the IMSC Z8001, b) write data from system memory to the tape and c) read data from the tape into system memory. These are described below.

5.7.1. DMA Receiving Instructions

Before the DMA chip can function, it must be programmed by the IMSC Z8001. The Z8001 must first place it in the receive instruction mode, then download the instructions.

To prepare the DMA chip to receive instructions, the Z8001 must activate CE*, IORQ* and WR* on the rising edge of the DMA clock (250CLK*). The data byte must be stabilized on the XCVRBUS by the rising edge of the next clock pulse. This data is latched off B'AD 0 through 7 by U87, which is enabled by ENXCVR*.

5.7.2. Tape-to-RAM Transfers

Before a DMA can take place, the Z8001 programs the DMA chip with a source port, a destination port and a block length. This is done as described above.

To implement the instructions it received from the Z8001, the DMA takes over control of its buses. To the devices on the buses, the Z8001 appears to be in control; this chip is designed to simulate it.

The DMA chip activates BUSREQ* to force the Z8001 to give up control (tristate) its outputs. The Z8001 returns BUSACK* to signal the DMA chip that it is in control. The DMA chip then requests the Multibus. When the Multibus is granted, BAI* goes low, and the DMA starts transferring data.

6. INTERRUPTS

The IMSC uses all three of the Z8001 interrupt capabilities: vectored interrupts, nonvectored interrupts and nonmaskable interrupts.

6.1. Vectored Interrupts

Vectored interrupts are used by the DMA chip, the CTC chips and the SIO. When one of these has a reason to interrupt the Z8001, it asserts its INT/ line, which connects to the Z8001 VI input. The Z8001 acknowledges the interrupt by asserting IORQ/ and M1/. It then looks for the interrupt vector, which is output by the interrupting device on the XCVR bus.

Each of the vectored interrupt-producing chips is programmed with the proper vectors, which provide one of the boundaries of the interrupt-handling routine. This data is latched from the XCVR bus to the B'AD bus, to the AD0 to AD7 pins of the Z8001.

The interupt is cleared when the Z8001 causes IORQ'PERIPH and M1/ to go low and P'RD to go inactive (high).

The DMA chip generates an interrupt to notify the Z8001 that it has completed a block transfer. CTC 1 interrupts the Z8001 in the event of a bus grant or a CPU Multibus interrupt reset. CTC 2 interrupts the Z8001 to signal a channel attention. The SIO interrupts the Z8001 when it receives a character or when it receives a status change from the terminal.

The vectored interrupt priority is arranged by a daisy chain consisting of the IEI/ and IEO/ lines of the DMA, CTCs and SIO.

6.2. Nonvectored Interrupts

The only nonvectored interrupt is used for diagnostics and debugging. The diagnostic software controls a switch which generates this interrupt to freeze all processes at their present state without disabling the CPU. This technique allows diagnostic personnel to inspect the state of the board at the time of the failure.

6.3. Nonmaskable Interrupts

Because there are valid reasons for disabling most interrupts, nonmaskable interrupts are only used when action must be taken regardless of other conditions. In the IMSC, this is only done in the event of a parity error or a power failure.

7. DMA

7.1. Overview

To qualify as a DMA, a circuit has to receive orders to move a block of data, and then move it without any more participation by the ordering device. In the IMSC, there are two true DMA circuits and one pseudo-DMA circuit. The disk sequencer provides DMA between the disk drives and the IMSC dynamic RAM. A Zilog Z80 DMA chip provides DMA between the tape drive and dynamic RAM, and the IMSC processor provides pseudo-DMA between the IMSC dynamic RAM and system memory array. This is not a true DMA because the processor gets involved.

7.2. Disk to IMSC Transfer

The disk sequencer provides the DMA function to transfer data between the disk and the IMSC dynamic RAM. This is a true DMA; the IMSC processor does not get involved after it issues the original commands.

Before the disk sequencer can begin a DMA transfer, it must provide various disk sequencer registers with the information that the sequencer will need to perform its function. To do this, it places the port address of the register to be loaded on the address bus. This enters U70, U144 and U160, which generate enables for the register to be loaded. The data for the register is then placed on the B'AD bus and appears on the input pins.

The IMSC processor must load registers with the following information before starting the sequencer:

Starting memory address -- Loaded into U98 through U101. When DKMUX is activated, this data is multiplexed down to 8 bits by U80 and U81, which output it on MADR0 to MADR7. This accesses a word in local RAM. That word is output on the B'AD bus for transmission to the disk during write operations and is read from the disk during read operations. The memory address in U98 through U101 is incremented by one and the process is repeated

Sector ID -- U114 and U115. When the disk reaches the proper track, it starts reading sectors into U102 and U105, which compare them with the sector loaded into U114 and U115. U116 generates DKIDEQ/ when they match.

Sector count -- U112 and U113. Because the minimum amount of data moved to or from the disk is one sector (512 bytes), this count controls the amount of data moved.

Command -- U131. This controls the type and direction of the action. The entire menu of disk sequencer commands is defined by the data in this latch. These commands are listed in Disk and Tape Controller, section 5.2.

When all the registers are loaded and the IMSC processor is ready to begin the DMA, it enables the command latch by activating uPSQ/LDCMD* (U132), which latches the command into the sequencer command latch. This ends the IMSC processor's involvement until the sequencer notifies it that the DMA is complete.

7.3. Tape DMA

The tape DMA is implemented with the Z80 DMA chip shown on page 8 of the IMSC schematics.

This chip works much like the disk sequencer in that the IMSC processor must load it with a starting address, a byte count and a command mode. This is different from the disk sequencer in that all this information is written to one place; it is stored in internal registers in the DMA chip, which is a port in the IMSC processor's I/O space.

This circuit goes through two distinct phases to accomplish its task. During the first phase, the IMSC processor programs the DMA chip. In the second phase, the DMA chip executes the program and transfers the data.

7.3.1. Programming the DMA Chip

When the IMSC processor wants to program the DMA chip, it activates DMAWR/ and writes a sequence of control bytes on the B'AD bus. These are latched onto the XCVR bus to appear on pins D0 to D7 of the DMA chip. The DMA chip must receive the following information:

Byte Count -- The number of bytes to be transferred.

Port A Address -- The address in local memory where the transfer is to begin. This is a variable address that will be auto-incremented by the DMA chip.

Port B Address -- The fixed address of the tape drive.

Control Bytes -- Determine the direction of data transfer and other internal details.

Interrupt ID -- Initializes the DMA with a vector ID, which is returned to the processor on DMA interrupt.

7.3.2. Transferring the Data

When the DMA chip is ready to begin, it asserts BUSREQ, which causes the IMSC processor to give up control of the relevant buses and return DMA BUSACK/. Then, depending on whether the control bytes specified a tape write or read operation, the DMA and tape drive perform an XFER/ACK cycle, and the data is transferred to (or from) the tape.

Meanwhile, the DMA must control the local memory, either to get the data from there (tape write) or to store it there (tape read). It activates DMAMRQ, which U122 uses to generate three signals which eventually become uPMRAS, uPMCAS and B'RW. These control the B'AD bus, the MEMBUS, and the XCVR bus. The DMA chip is arranged internally to output the proper address and data to correspond to the CAS, RAS and B'RW signals in local memory (RAM).

The DMA chip is in the circuit in parallel with the processor, which gives up control of the buses and allows it to take over. To the latches and other devices involved, the Z8001 appears to retain control.

The DMA control cycles are interleaved with the IMSC processor cycles.

7.4. Local RAM to Memory Array -- Pseudo DMA

The transfers from the IMSC local RAM to the system main memory array are implemented using the block move capability of the Z8001. This is not a true DMA because the processor gets involved.

To accomplish a block move, the IMSC processor gives itself instructions. It loads the beginning memory address and byte count, and requests the Multibus. Upon receipt of the Multibus, it moves the entire block. The blocks are kept small enough so they can be moved in a single Multibus access. A typical block is 128 to 256 bytes long.

The block move is implemented by using the portion of the IMSC processor's memory space that is mapped into the Multibus memory space. The processor orders data to be written into (or out of) segment 9 of its memory space. This causes it to access the Multibus and transfer data to (or from) system main memory.

8. LOCAL BUS STRUCTURE

The IMSC requires an elaborate system of buses to maintain three concurrent DMAs. These buses, which are shown in Figure 1, are described below:

uPADBUS -- Takes AD 0 to AD 15 from the Z8001 to a series of latches (U88 through U91), which latch data to and from the B'AD bus and addresses to and from the LADR bus.

B'AD bus -- The primary carrier of data to and from the Z8001. Latched from the uPADBUS by U90 and U91. 16 bits wide.

LADR bus -- The primary carrier of addresses to and from the Z8001. Latched from the uPADBUS by U88 and U89. 16 bits wide.

MADR 0 to 7 -- Carries addresses from the LADR bus. The 16-bit addresses are multiplexed by U32 and U33 into 8-bit bytes. Carries addresses to local RAM and to the disk sequencer. 8 bits wide.

MEMBUS -- Carries 16 bits between the B'AD bus (latched by U45 and U48) and the data pins of local RAM. Also goes to U62 and U66, which latch data to and from the SERBUS.

SERBUS -- Main purpose of this bus is to carry 16 bits of data to and from U82 and U83, which are the serial-to-parallel conversion chips used to transmit and receive from the disk drives. Can latch data to and from the MEMBUS and the B'AD bus.

SERHR -- Takes disk sector IDs from the SERBUS to the disk ID equate register (U116) and takes all disk data to U84. Bytes are latched onto it by U102 and U105. 8 bits wide.

SQDATA, SQSTATE -- Used to pass information around in the disk sequencer. Selected bits are used to generate control logic which controls the disk drive.

XCVR -- Carries data to and from the CTCs, SIO and DMA chips. The data is latched on and off the B'AD bus by U87.

9. CTCs AND SIOs

9.1. Counter Timer Circuits

The IMSC has two Z80 CTC chips on board. Each of these has four channels (0 to 3). Each channel has a unique function. The IMSC processor downloads programs and time constants for each channel. Programming details are in the Zilog 1981 Data Book. The following paragraphs provide an overview of the function of each channel.

CTC 0, Channel 0 -- Generates multibus requests. It is programmed to operate in the timer mode and to generate an output roughly every 1 millisecond. The ZCTO0 signal turns into BUS'796'ARB'REQ.

CTC 0, Channel 1 -- Generates the baud rate for serial port 0. The diagnostic software tells the IMSC processor to program it for the proper baud rate, which may be between 50 and 38,400 baud.

CTC 0, Channel 2 -- Not used.

CTC 0, Channel 3 -- Interrupts the IMSC processor when the system CPU processor wants to reset the Multibus interrupt. It is programmed in the counter mode to generate an output on every input. The input is PORTADD'IR, and the output is VECT'INT/.

CTC 1, Channel 0 -- Generates the baud rate for serial port 1. The details are the same as CTC 0, channel 1.

CTC 1, Channel 1 -- Not used.

CTC 1, Channel 2 -- Not used.

CTC1, Channel 3 -- Programmed in the vectored interrupt mode. It sends a local vectored interrupt to the IMSC processor when the CPU processor sends a channel attention. The input signal is PORT'ADD'CA/. The IMSC processor demands a local interrupt vector, which CTC 1 puts out on the XCVR bus.

9.2. Serial Input/Output

The IMSC has two serial input/output channels on board. These are used for factory and field diagnostics, in conjunction with the diagnostic software.

The connecting and control circuitry for these ports is very basic and simple. The baud rates come from the CTC chips, the data comes from the XCVR bus, and the only connections to the terminals is TxD, RxD and ground. They are almost completely controlled by the diagnostic software.

10. ERROR HANDLING

10.1. Overview

The IMSC has three different local data error handling circuits. Disk errors are detected and corrected by an ECC chip which uses a 32-bit polynomial code (see the IMSC schematics, page 10). The intelligent tape drive provides its own error-handling capability transparent to the IMSC. The IMSC uses a byte-parity system (see the IMSC schematics, page 4).

10.2. Disk Errors

To ensure that what is written on the disk is exactly what is read, the IMSC uses an ECC chip that generates a 32 bit polynomial error code.

The ECC chip (U84) reads bytes from SERHR as they are written on the disk. At the end of each sector, it outputs 4 bytes on the SERBUS. These are stored on the disk at the end of the sector.

When the sector is read from the disk, the 4 bytes (32 bit polynomial code) are read with it. The entire sector plus the 32 bits are fed back through the ECC chip and a new polynomial code is generated.

The nature of the 32 bit polynomial code used is such that if an identical sector plus the code is coded again, the resulting code must equal zero. Any other result means that the data was not correct. This condition causes U84 to activate ERR. The IMSC software then uses the polynomial code to reconstruct the original data.

10.3. Tape Drive Errors

The intelligent tape drive provides its own error detection and correction without participation by the IMSC. When it is necessary to inform the IMSC, it sets its exception (EXC) line, which causes the IMSC to request the status. The IMSC software provides the proper response, based on the status read in.

When the intelligent tape drive signals an error that it cannot correct, the IMSC software informs the operating system.

10.4. Local Memory Errors

The IMSC dynamic RAM circuit uses a parity circuit to detect errors. The RAM bank is composed of sixteen 64K-by-1-bit RAMs. Two identical chips read in the parity bit generated by U44 and U50. When the data is read out of RAM, U44 and U50 again generate a parity bit each, and they are exclusive ORed in U10 with the previously stored bit. If they do not match, the exclusive OR generates an error output.

This circuit is explained in Disk and Tape Controller, section 3.

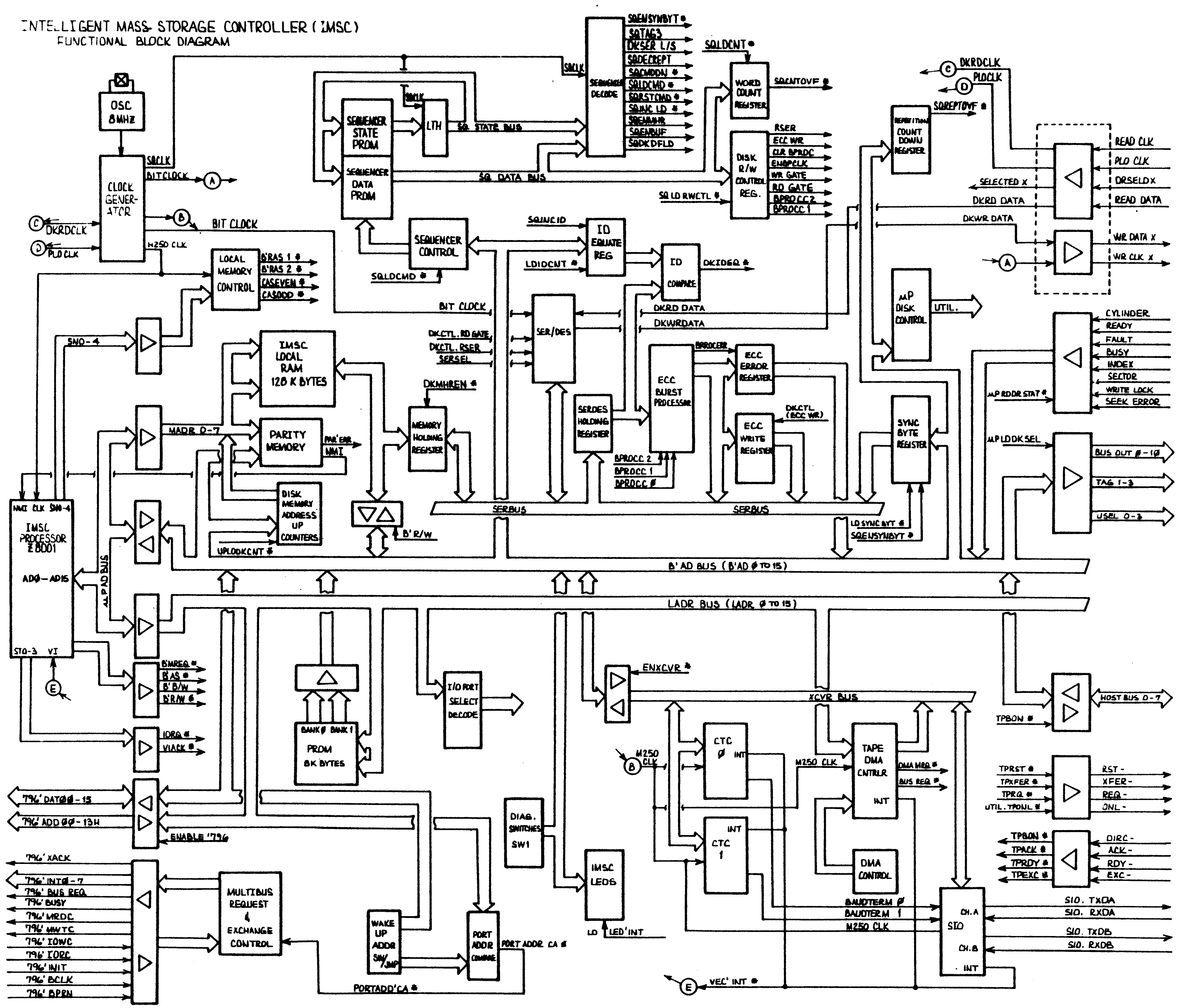
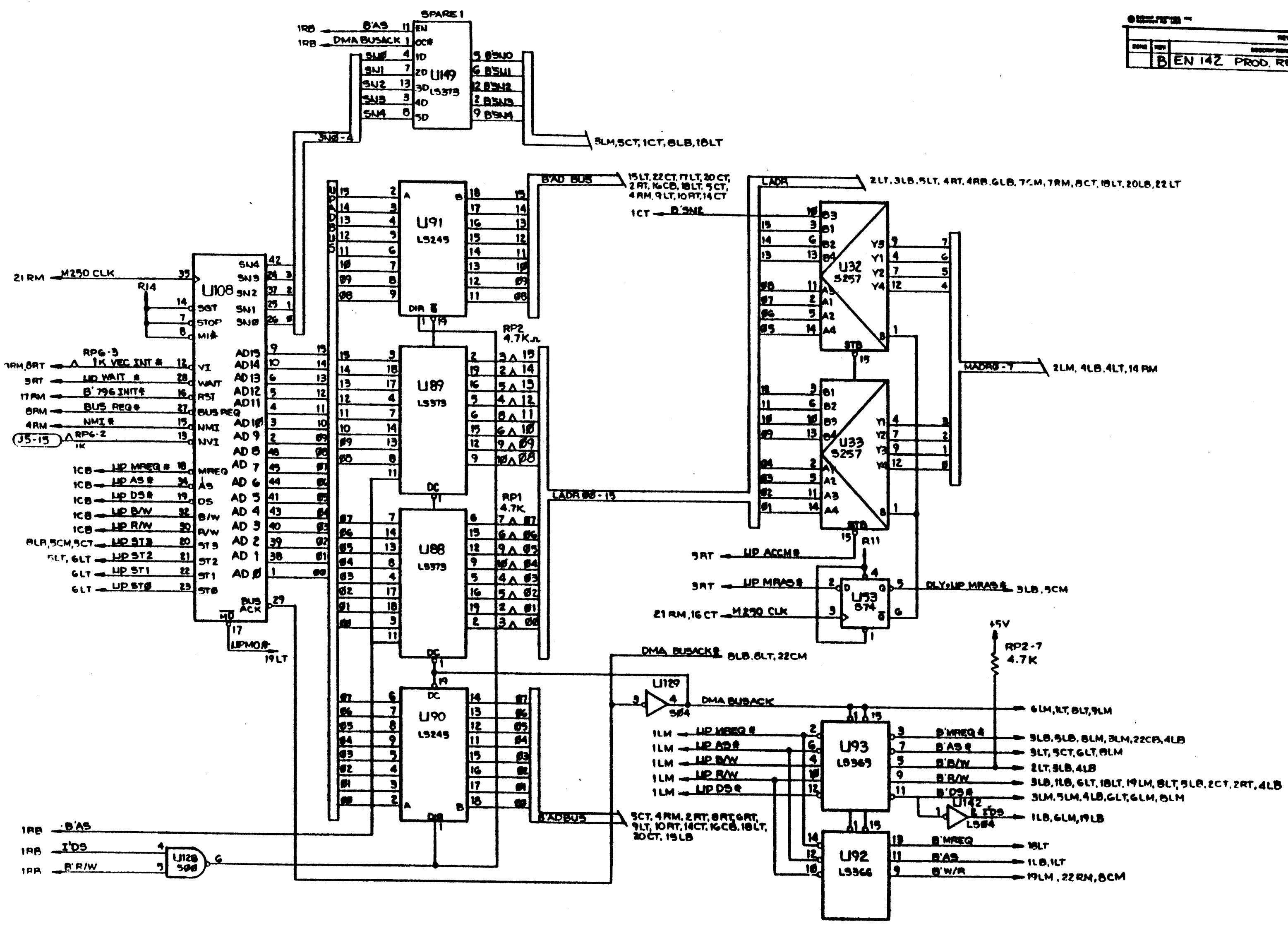


Figure 1. IMSC Board Block Diagram



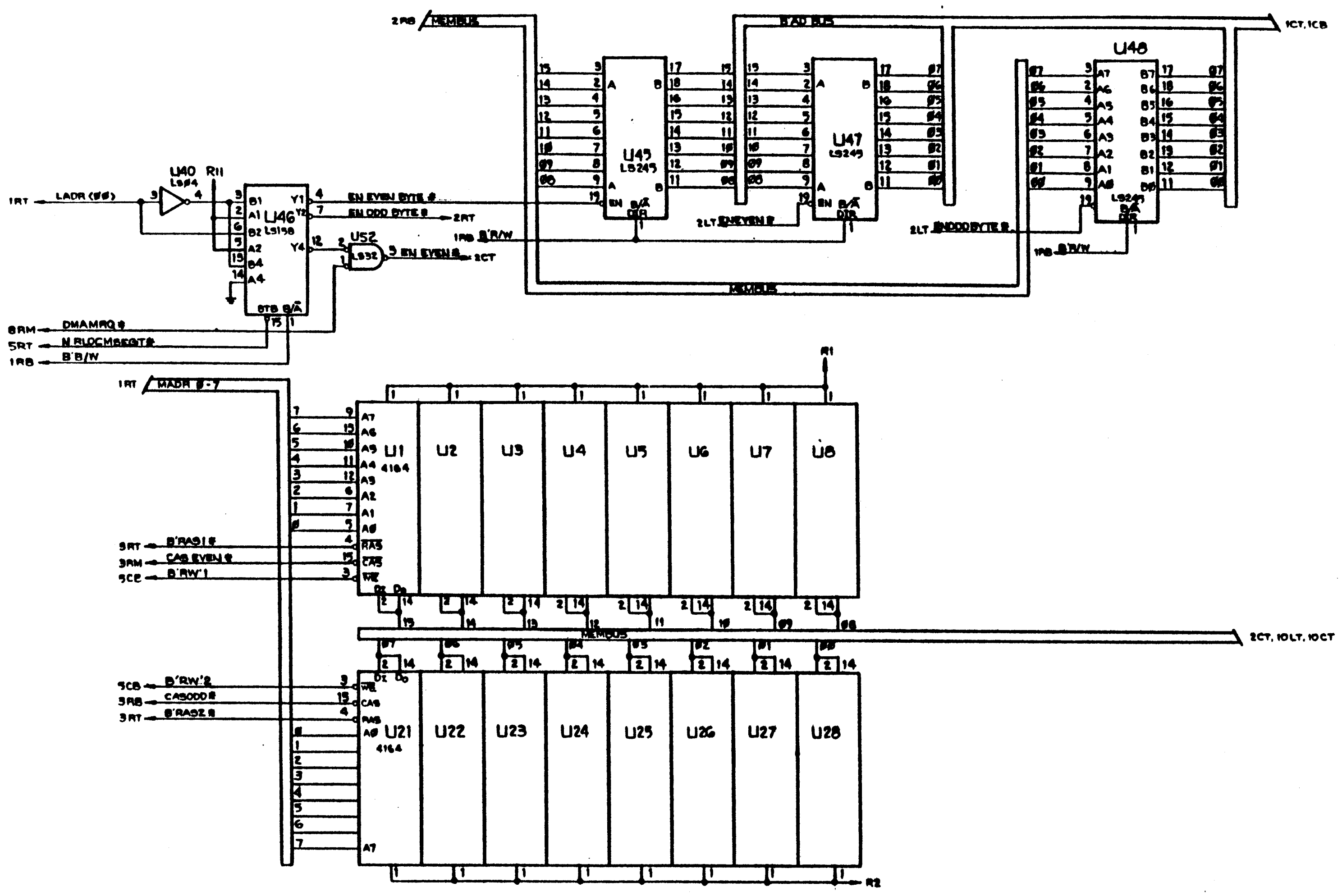
ZONE	REV	DESCRIPTION	DATE	APPROVED
	B	EN 142 PROD. RELEASE	6-2-82	MORRISON

PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER (Z8001 MICROPROCESSOR)

SCHEMATIC: IMSC

SIZE		REV
D	60-00083	B
SCALE: ~	SHT. 1 OF 23	



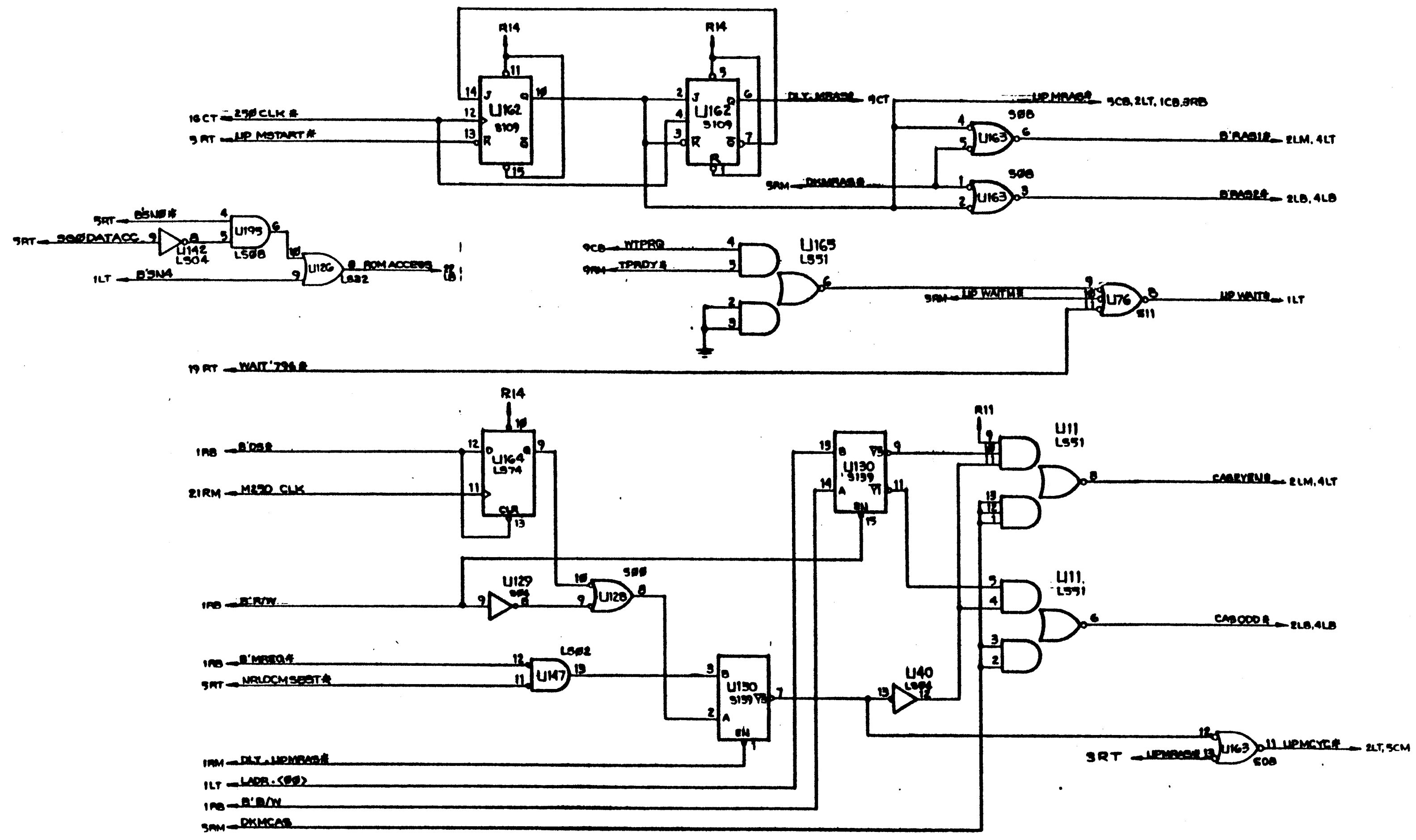
PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER

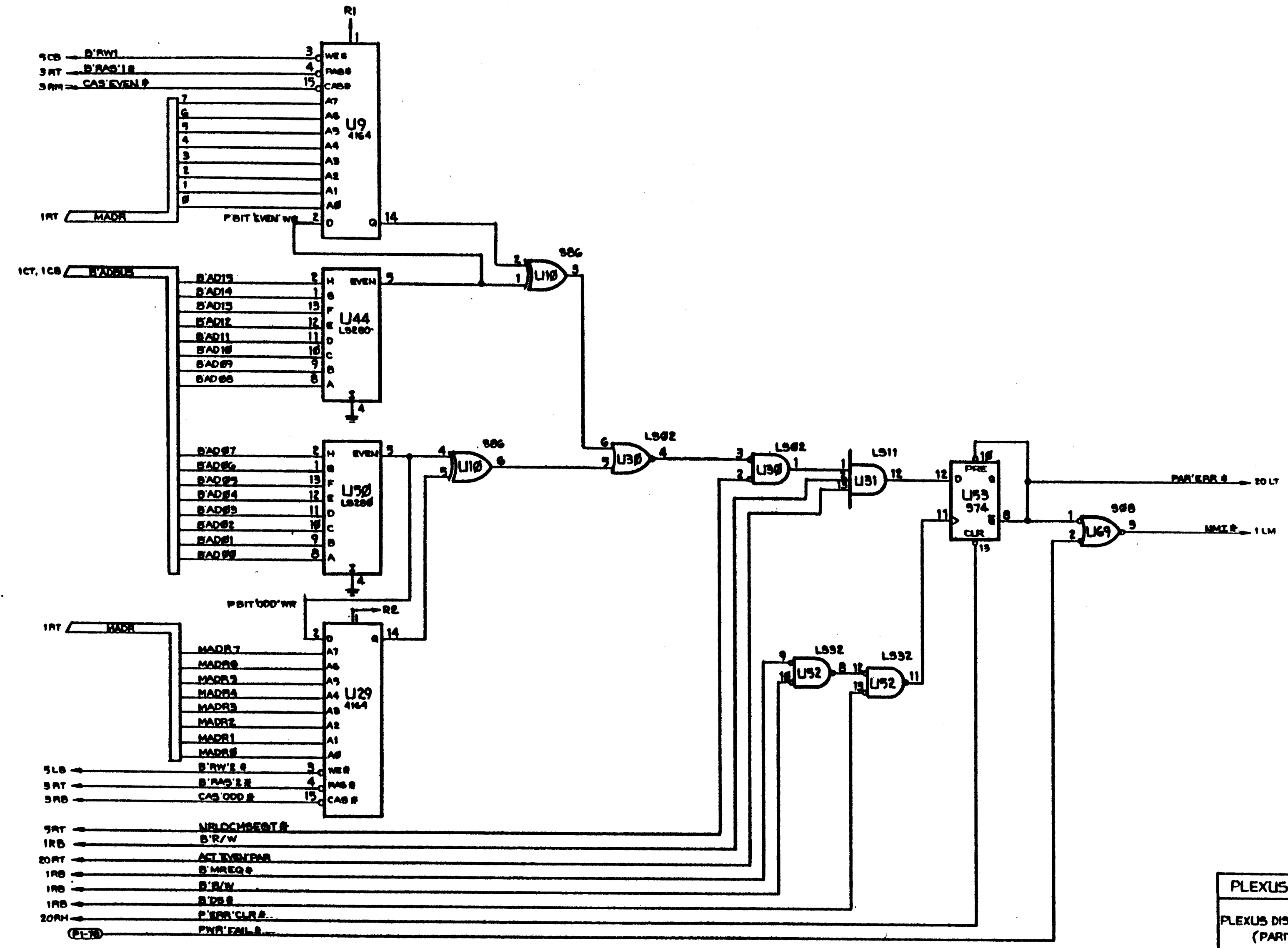
(CONTROLLER MEMORY)

SCHEMATIC: IMSC

SIZE			REV
D	60-00083		B
SIZE	SCALE: ~	SHT 8 OF 23	REV



PLEXUS COMPUTERS INC.			
PLEXUS DISK/TAPE CONTROLLER (MEMORY CONTROL LOGIC) SCHEMATIC: IMSC			
D	60-00083		B
SIZE	SCALE: ~	SHT: 3 of 23	REV

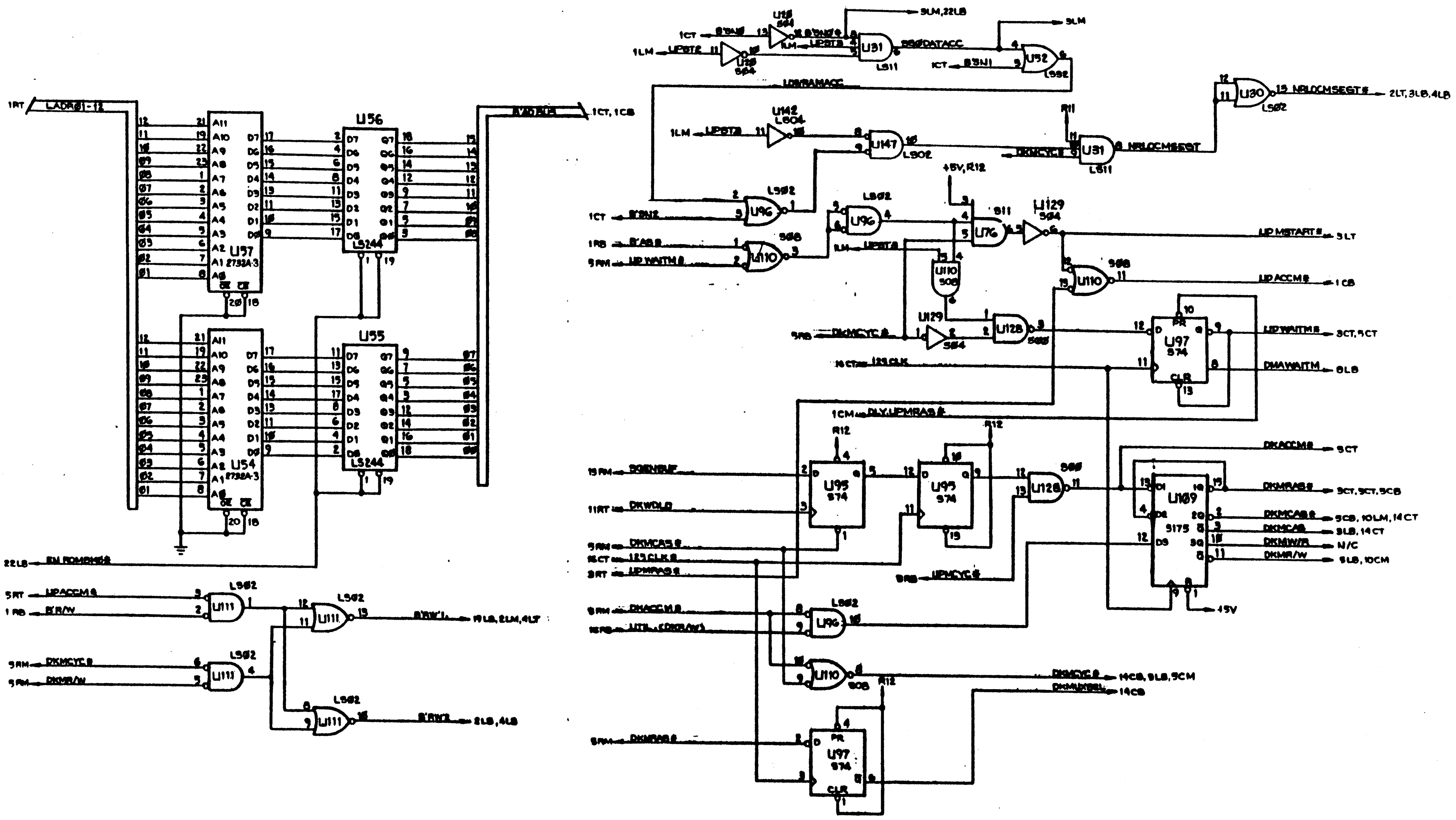


PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER
(PARITY MEMORY)

SCHEMATIC: IM8C

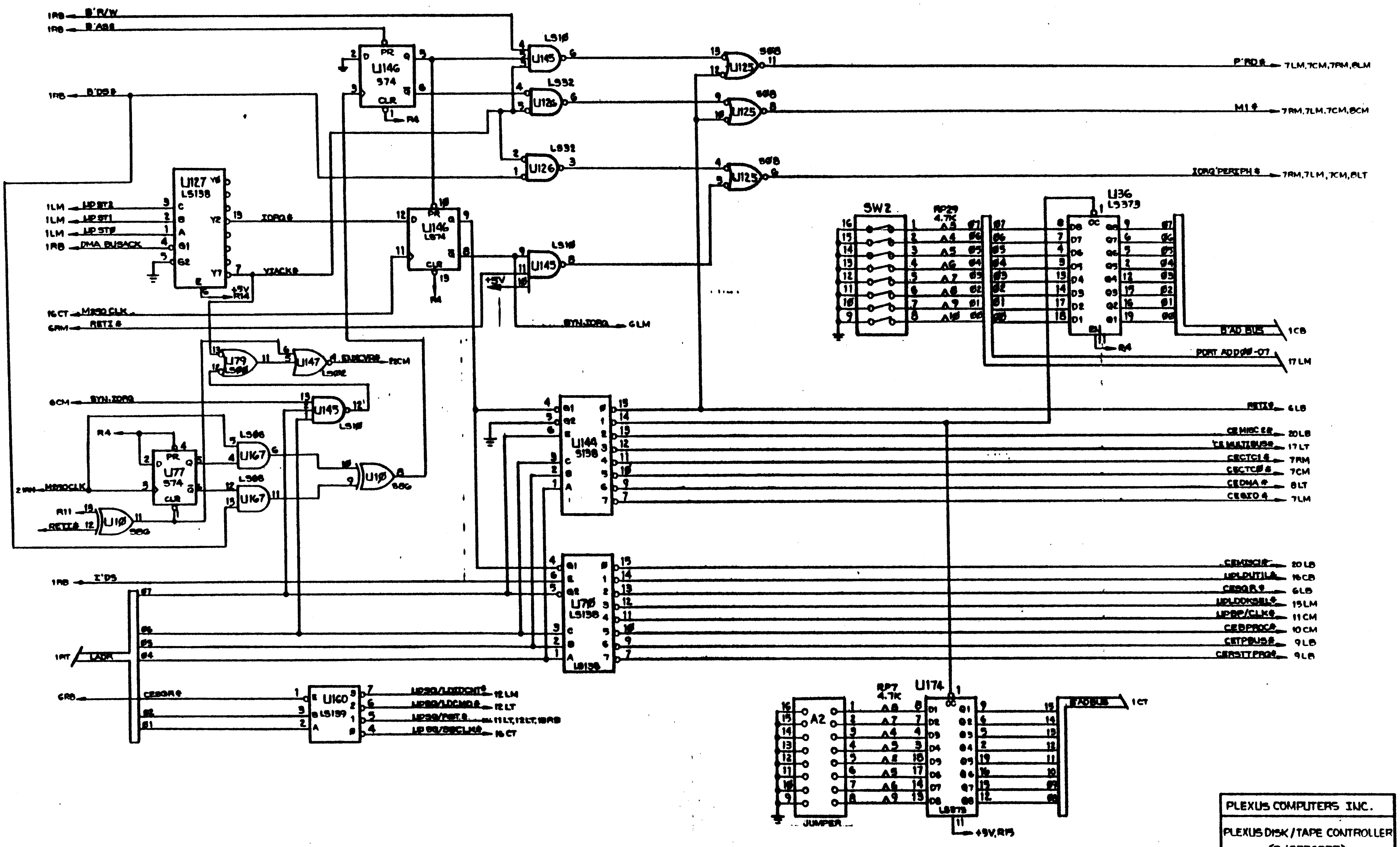
D	60-00083	B
SIZE	SCALE: ~ SHT 4 OF 23	REV



PLEXUS COMPUTERS INC.

PLEXUS DISK / TAPE CONTROLLER (LOCAL ROM & MEMORY ARBITER) SCHEMATIC: IMSC

SIZE		SCALE: ~	SHT 5 OF 23	REV
D	60-00083			B

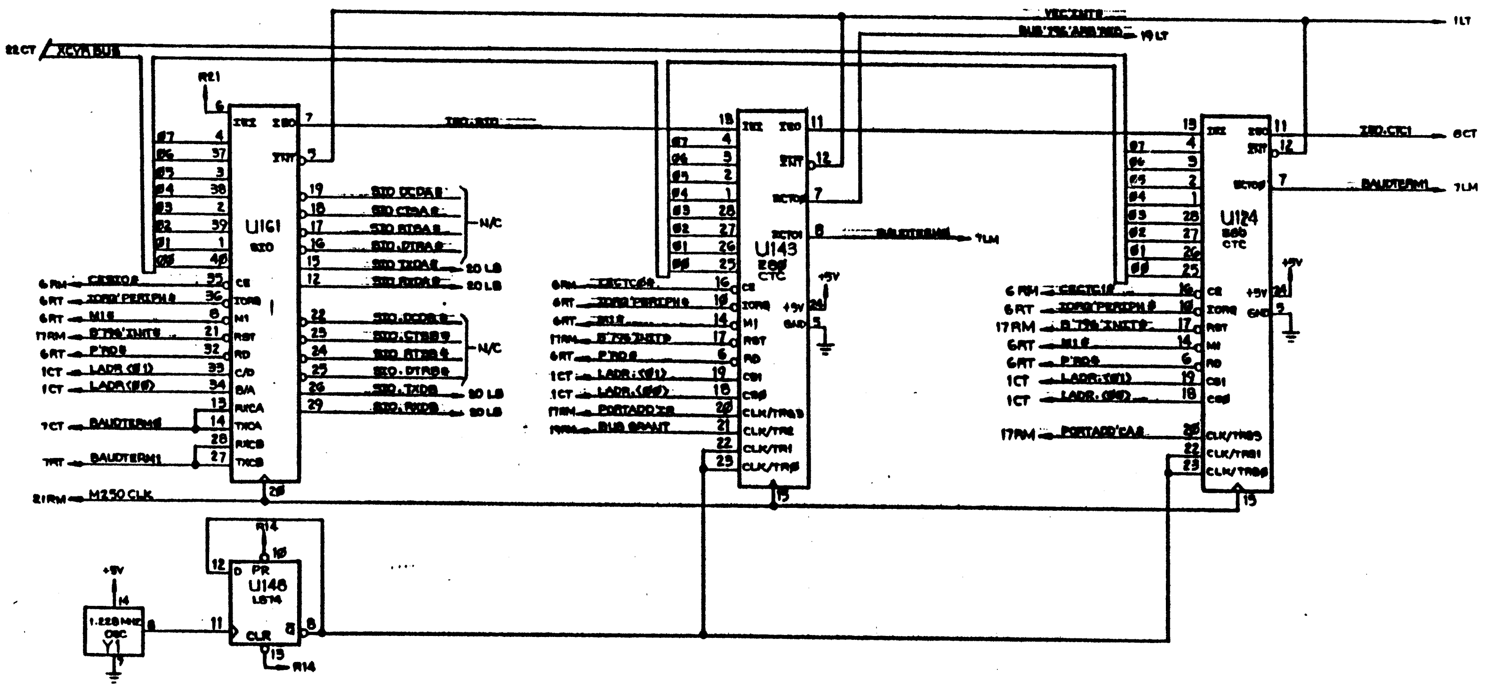


PLEXUS COMPUTERS INC.

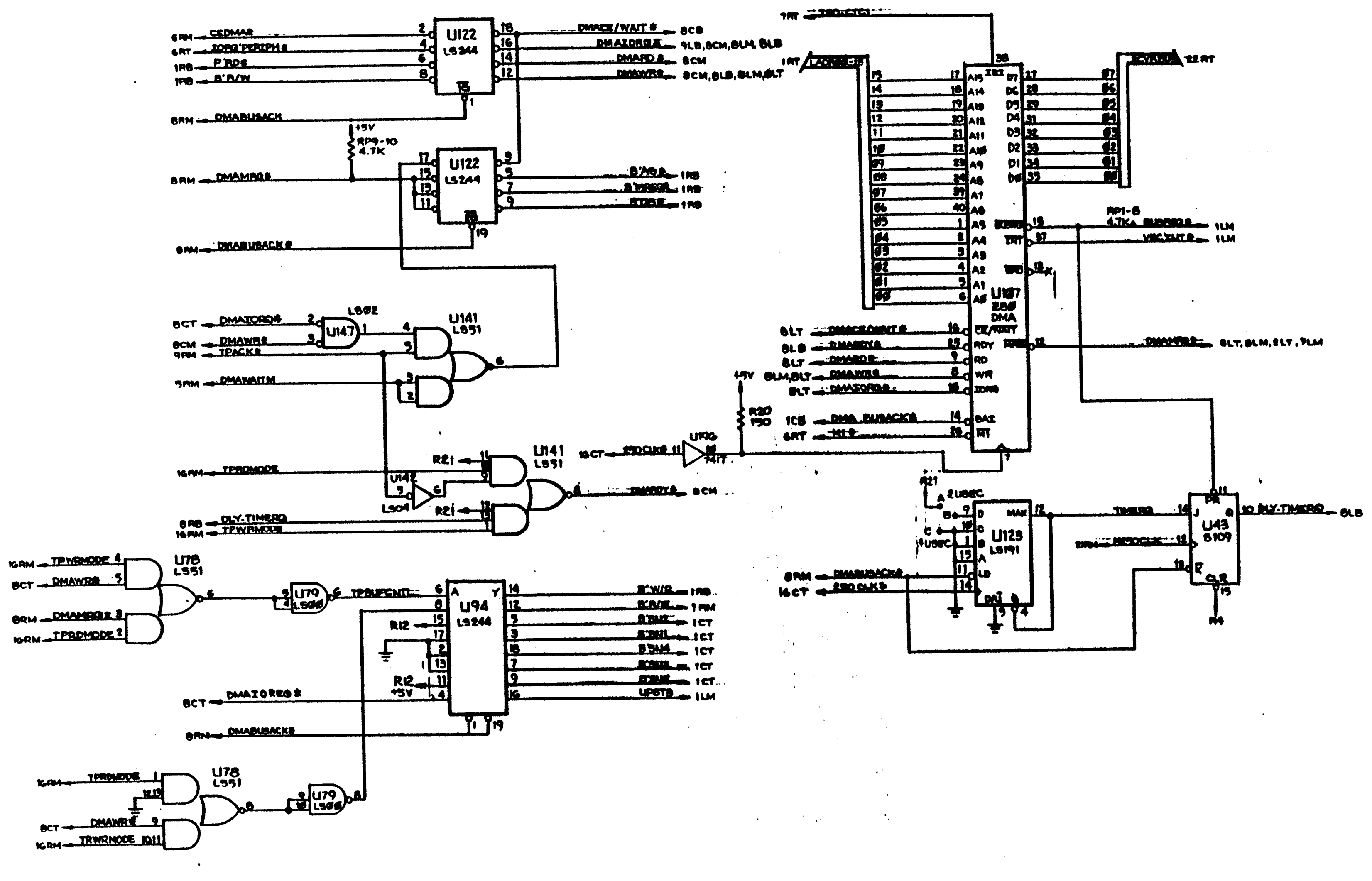
PLEXUS DISK/TAPE CONTROLLER (I/ODECODE)

SCHEMATIC: IMSC

SIZE	SCALE: ~	60-00083	SHT 6 OF 29	REV B



PLEXUS COMPUTERS INC.		
PLEXUS DISK/TAPE CONTROLLER (SIO ∮ CTC) SCHEMATIC: 1MSC		
	60-00083	B
SIZE	SCALE ~ SHT 7 OF 23	REV

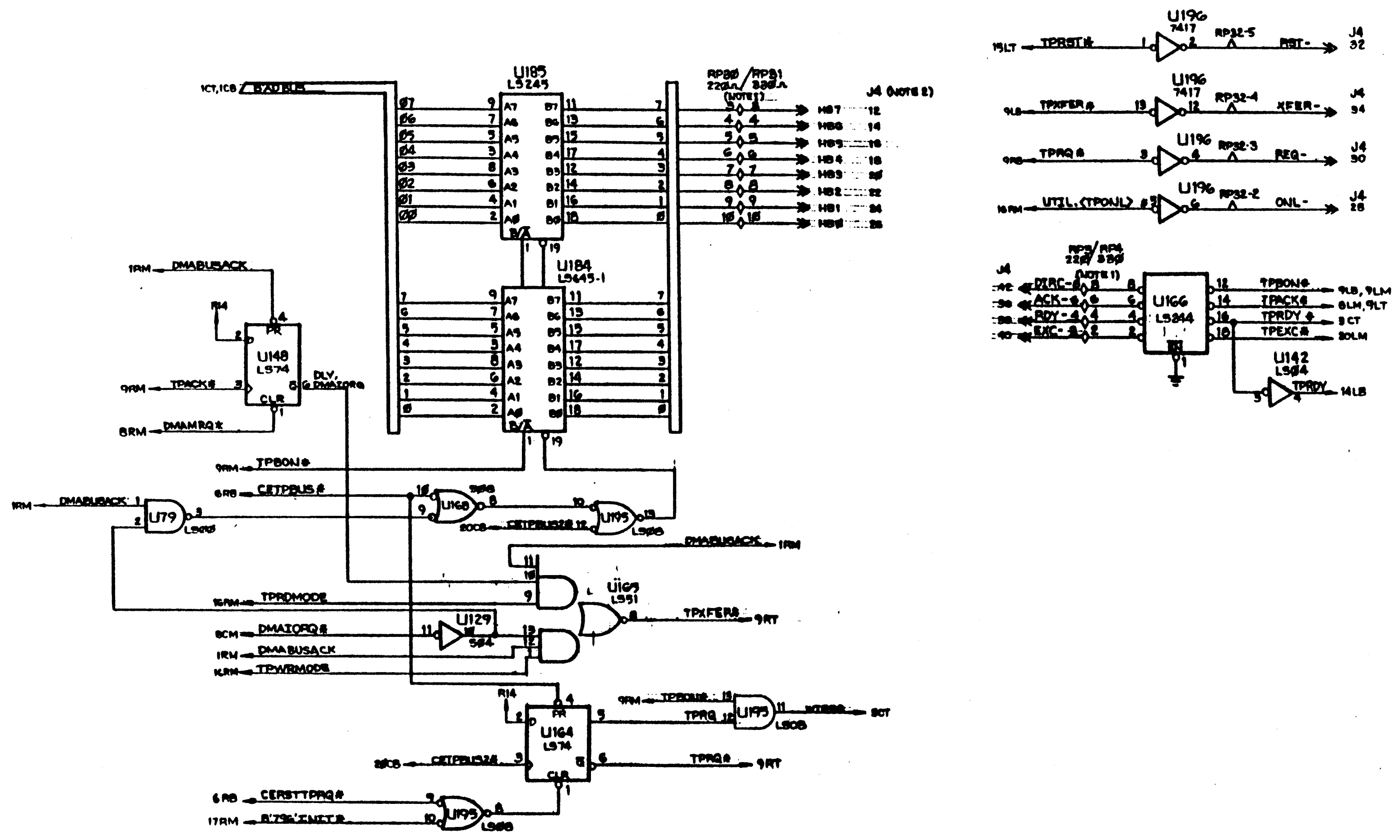


PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER (TAPE DMA)

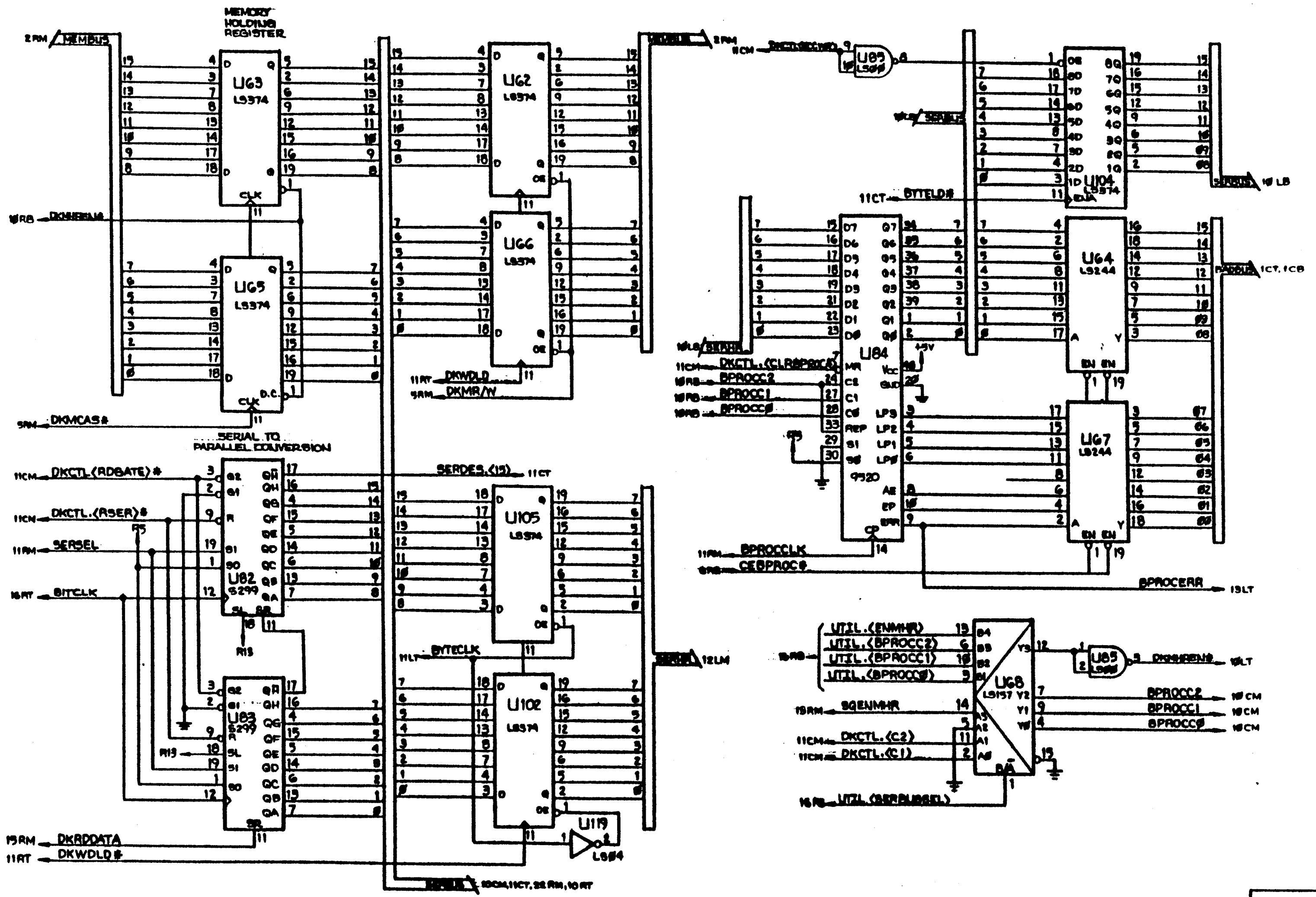
SCHEMATIC: IMSC

SIZE		SCALE	SHT	REV
D	60-00083	~	8 OF	B



1. ◊ STANDS FOR 220Ω/330Ω RESISTOR (330 TO GND, 220 TO +5) TERMINATOR SIPS.
2. J6: ALL ODD PINS GROUNDED

PLEXUS COMPUTERS INC.		
PLEXUS DISK/TAPE CONTROLLER TAPE INTERFACE SCHEMATIC: IMSC		
D	60-00083	B
SIZE	SCALE: ~ SHT: 9 of 13	REV



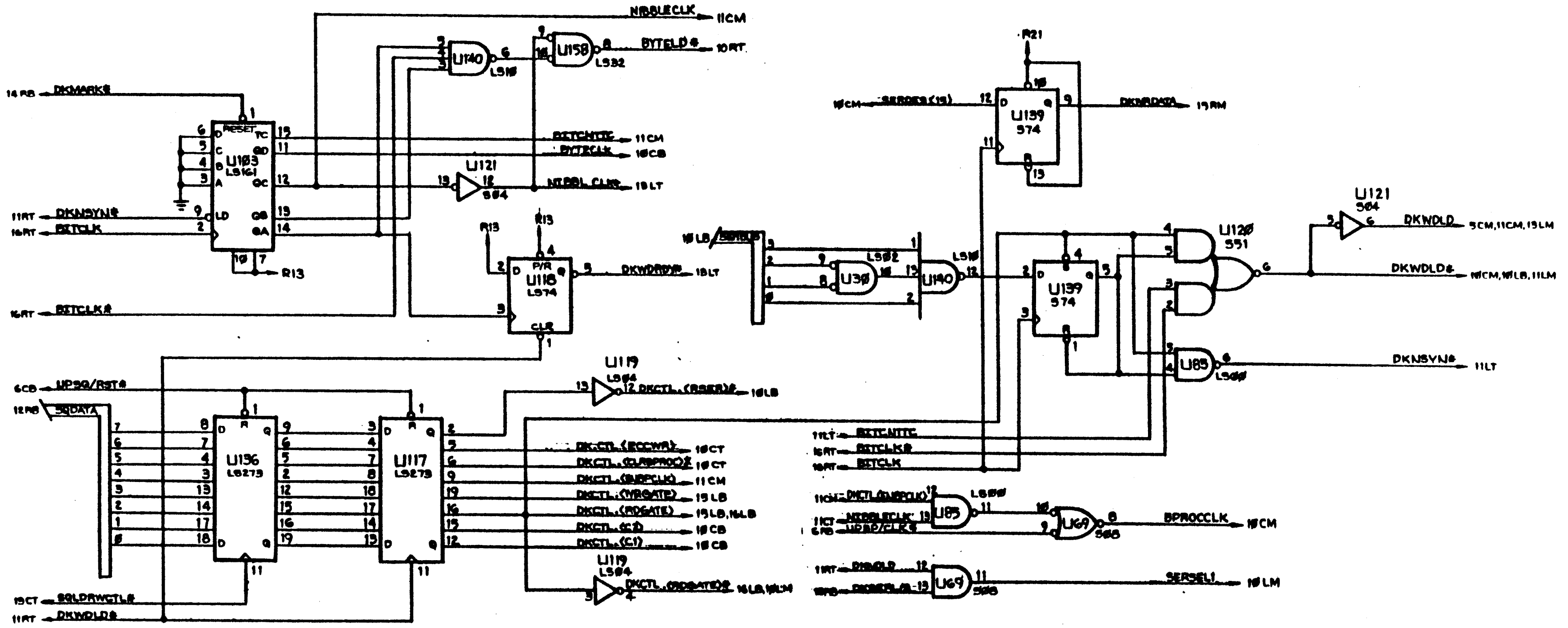
PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER (SERDES & ECC PROC.)

SCHEMATIC: IMSC

SIZE		REV
D	60-00083	B

SIZE SCALE: ~ SHT 10 OF 25 REV

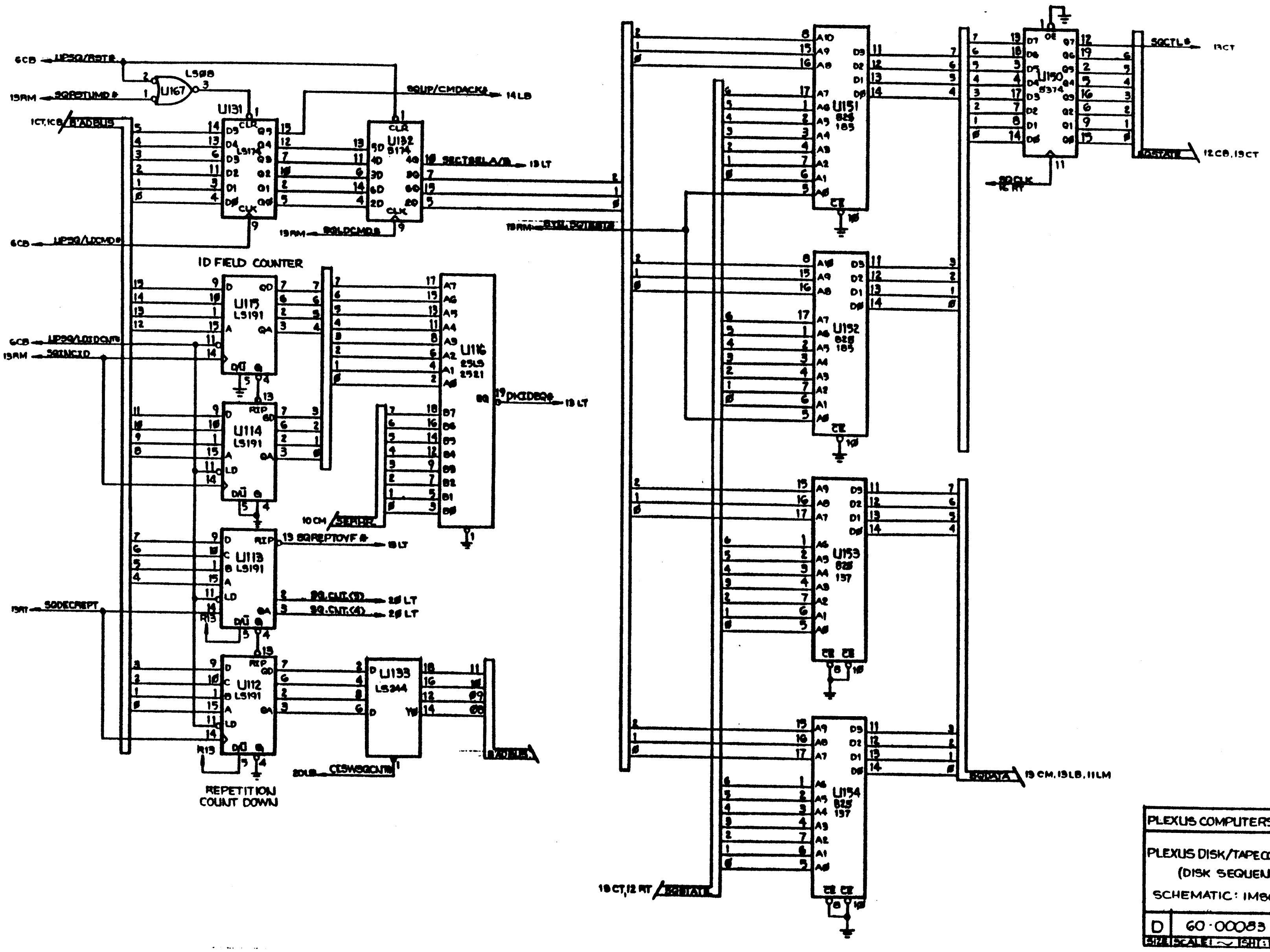


PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER (DISK R/W CTL LOGIC)

SCHEMATIC : IMSC

SIZE		SCALE: ~ SHT: 11 23	REV
D	60-00083		B

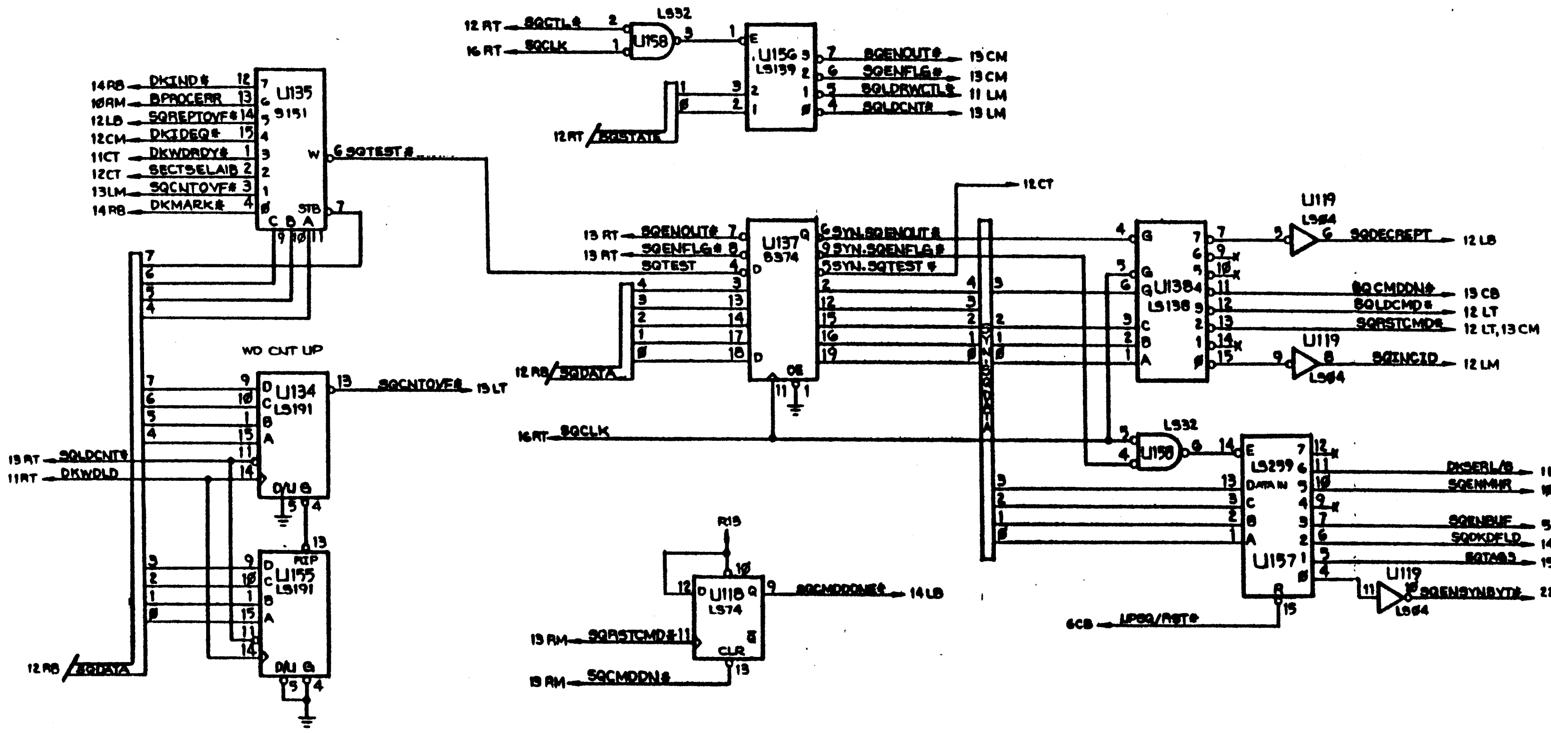


PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER
(DISK SEQUENCER)

SCHEMATIC: IM8C

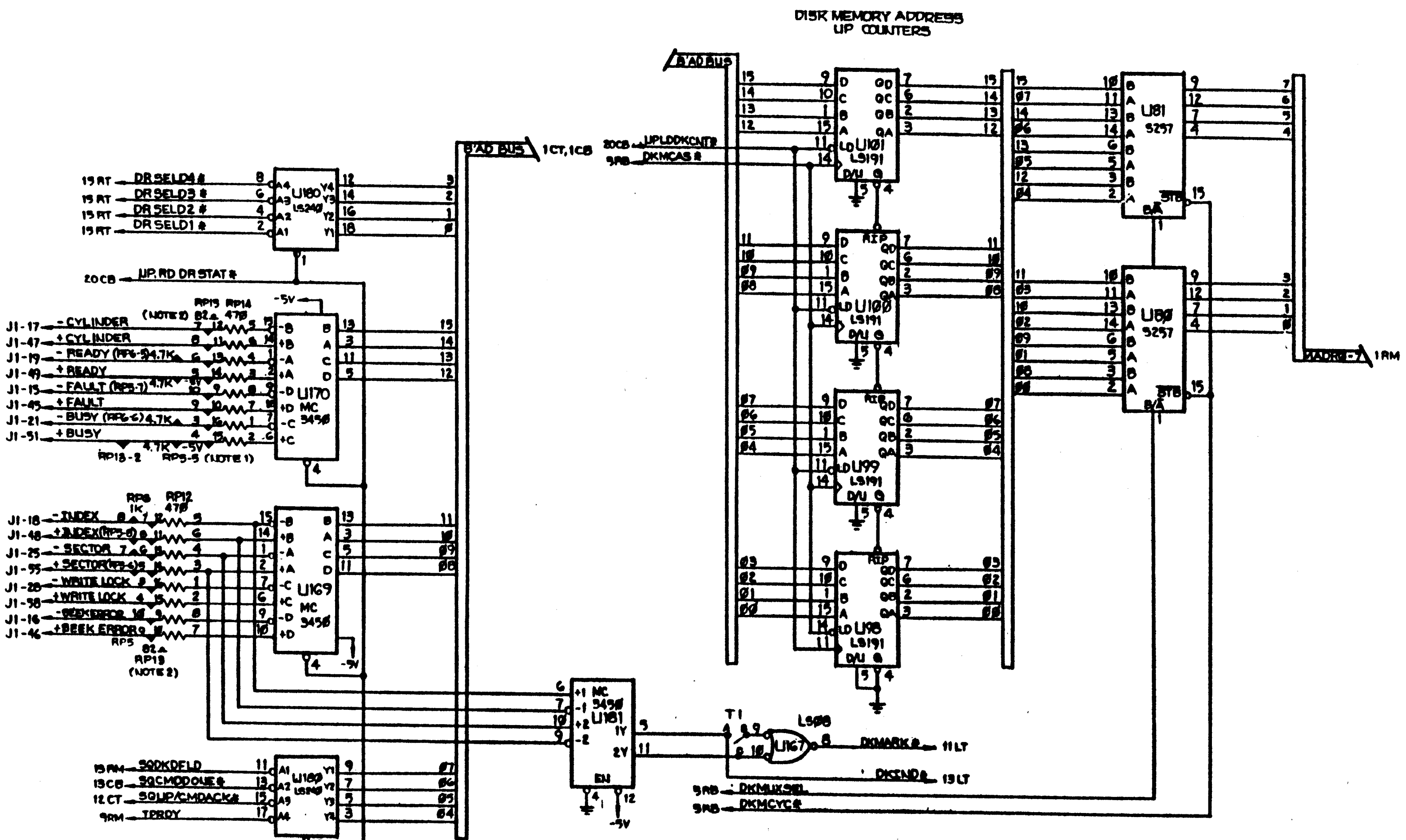
SIZE			REV
D	60-00083		B
	SCALE: ~	SHT: 12 OF 23	



PLEXUS COMPUTERS INC.

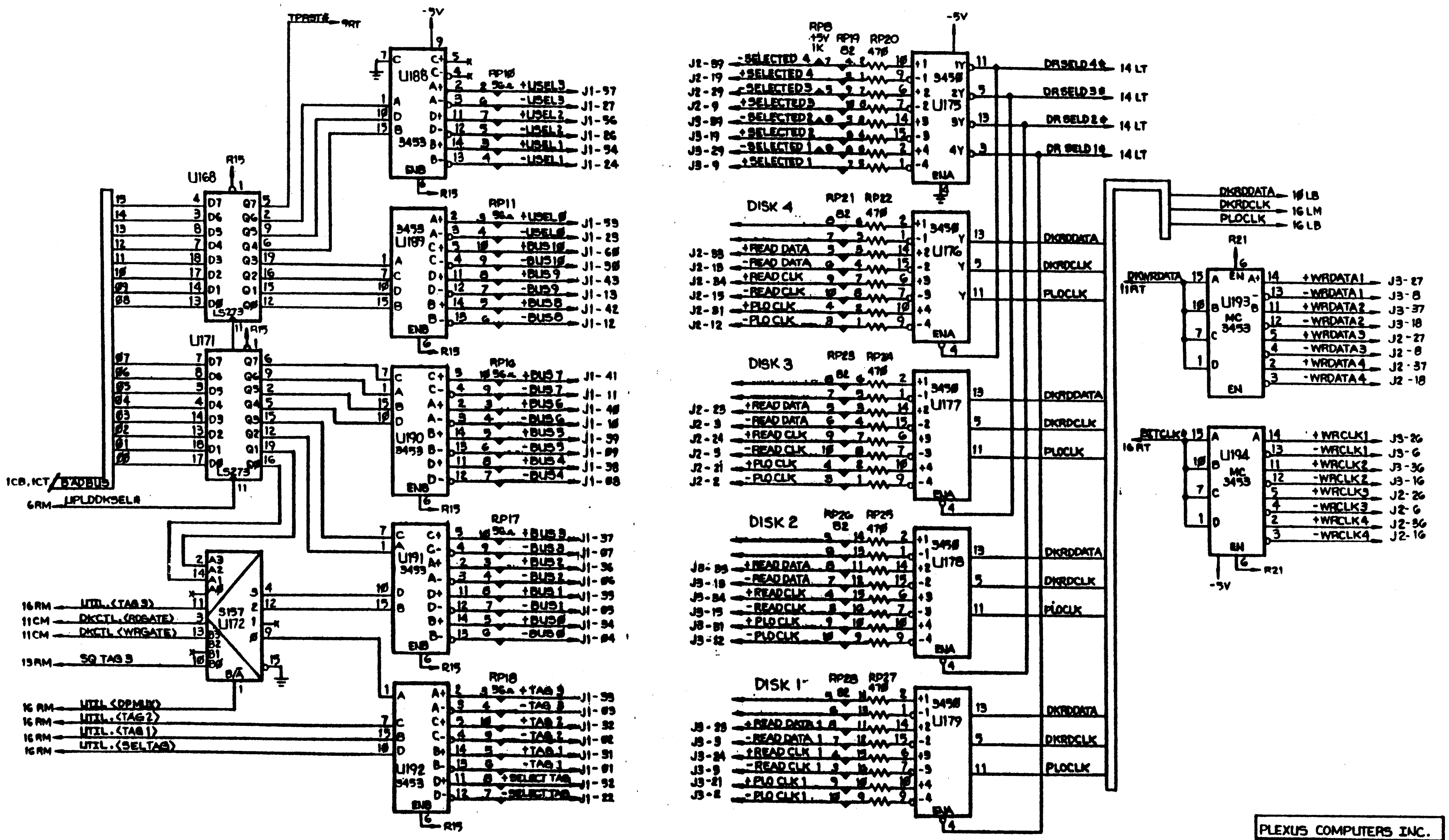
PLEXUS DISK/TAPE CONTROLLER
(SEQUENCER DECODE)
SCHEMATIC: IM6C

SIZE		SCALE	SHT. 13 OF 23	REV
D	60-00085	~		B



NOTE 1: RP5 IS 4.7K Ω TO -5 VOLTS.
NOTE 2: RP13, RP15 IS 82 Ω TO GND.

PLEXUS COMPUTERS INC.		
PLEXUS DISK/TAPE CONTROLLER (DISK INTERFACE & ADDR. CNTR) SCHEMATIC: IMSC		
D	60-00083	B
SIZE	SCALE ~ SHT. 14 OF 23	REV



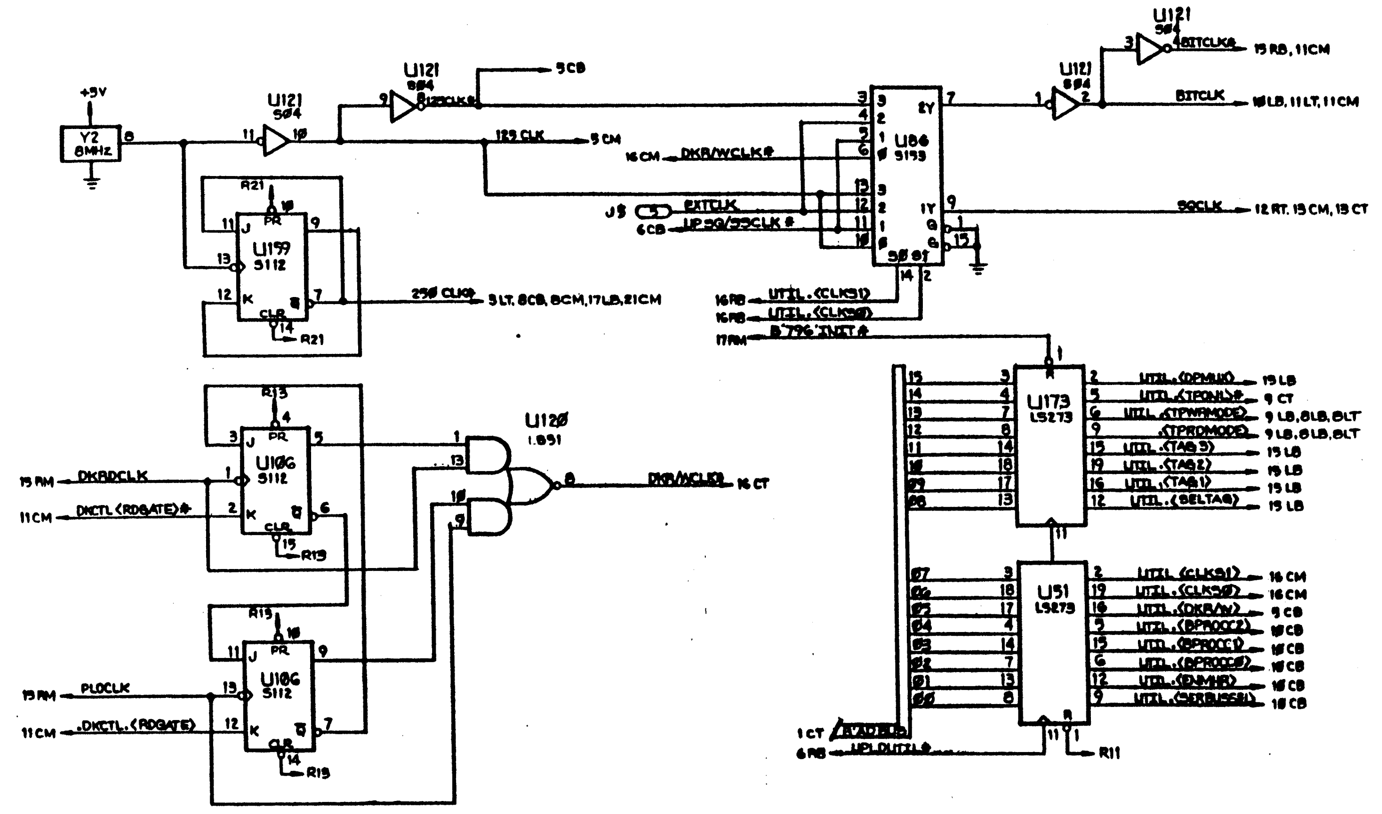
NOTE: ALL 82 Ω, 56 Ω RESISTORS TO GROUND.

PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER (DISK INTERFACE)

SCHEMATIC: IMSC

SIZE	SCALE	SHT.	REV
D	60-00083	15 OF 23	B

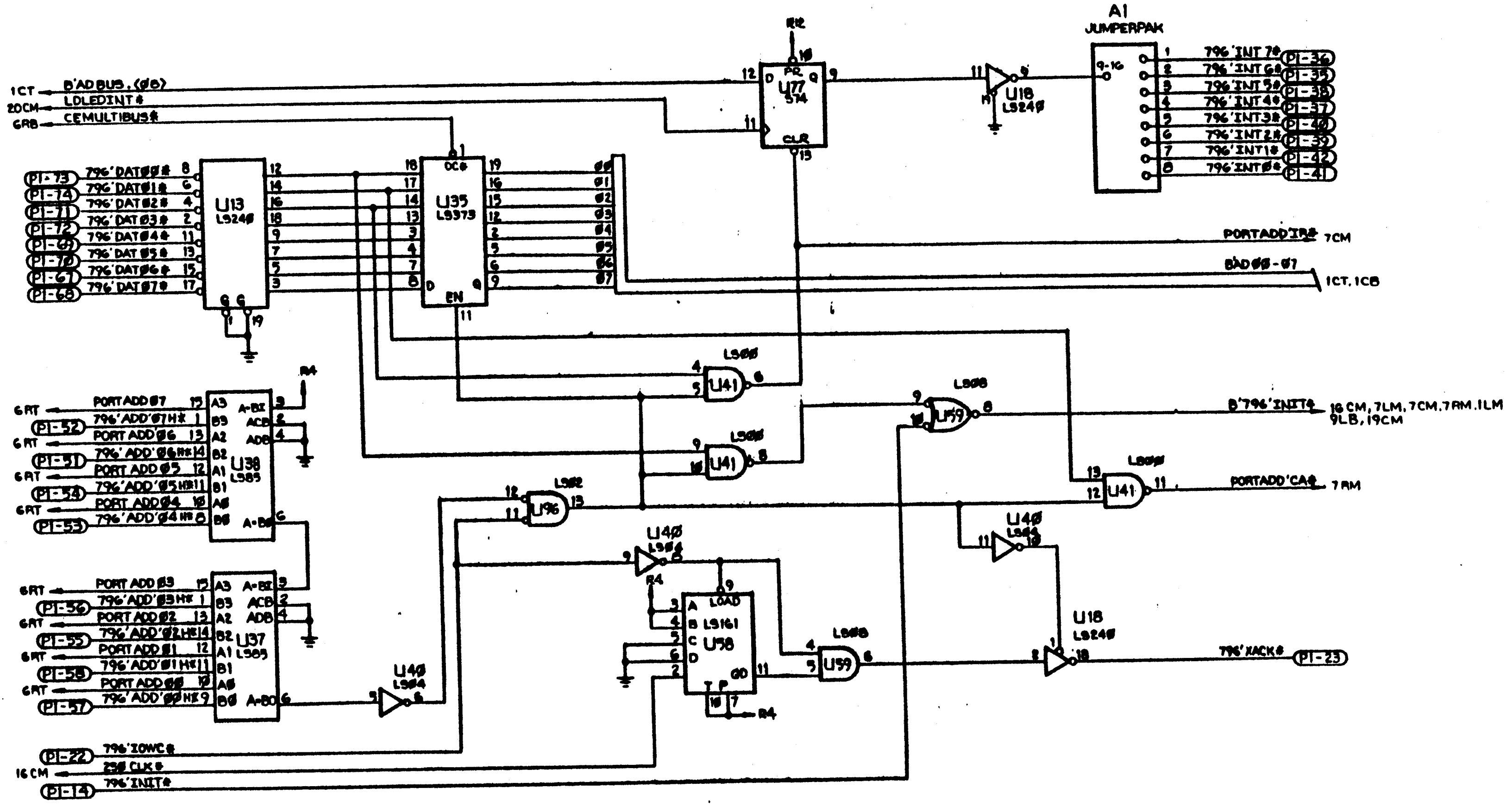


PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER (CLOCK GENERATION)

SCHEMATIC: IMSC

SIZE		REV
D	60-00083	B
	SCALE: ~ SHEET [illegible] OF [illegible]	

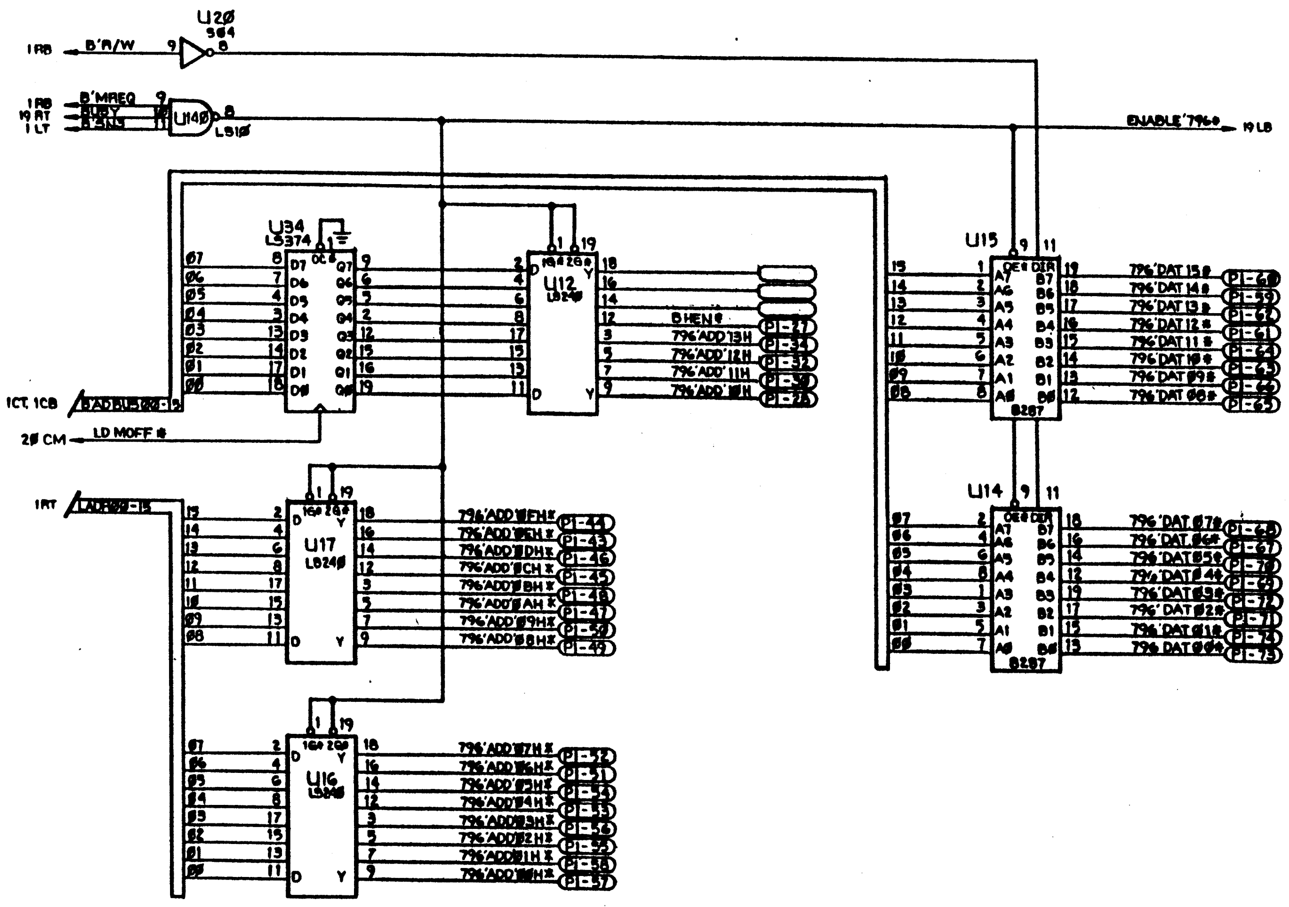


PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER (MULTIBUS INTF 1)

SCHEMATIC: IMSC

SIZE	SCALE	SHT	REV
D	60-00083		B



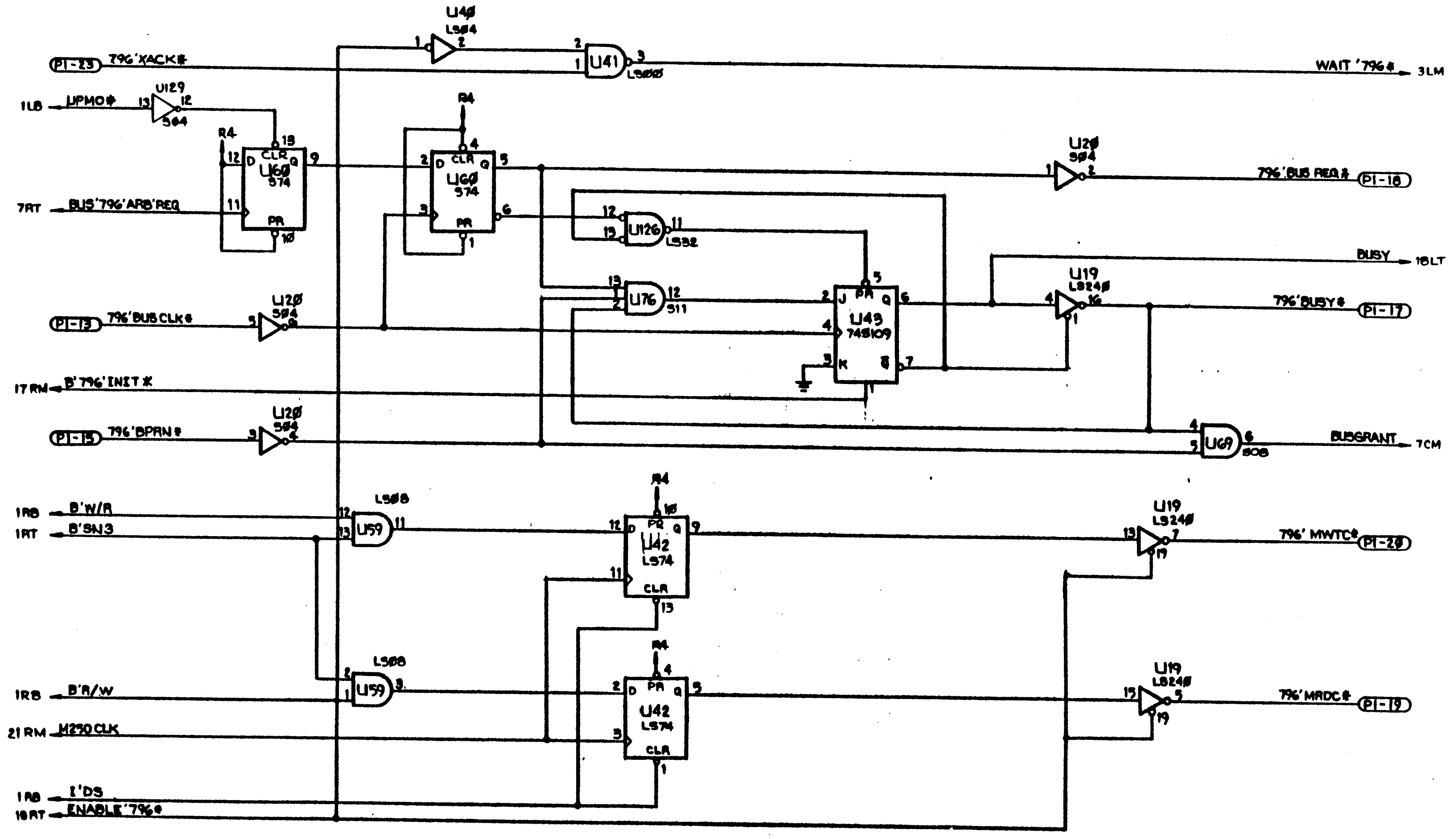
PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER

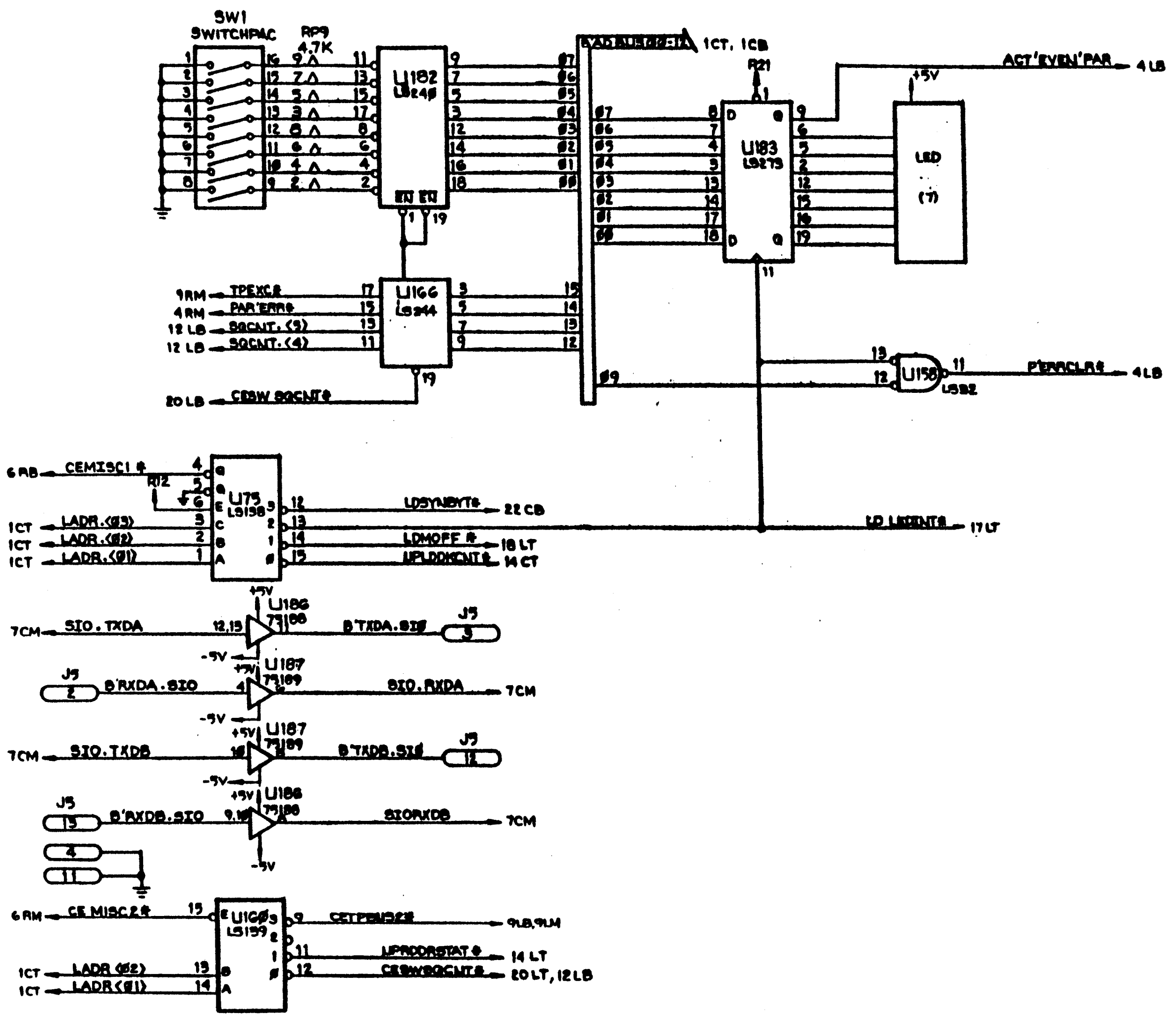
(MULTIBUS INTF II)

SCHEMATIC: IMSC

SIZE			REV
D	60-00083		B
	SCALE: ~	SHT. 15 OF 23	



PLEXUS COMPUTER INC.	
PLEXUS DISK/TAPE CONTROLLER (MULTIBUS INTF III)	
SCHEMATIC: IMSC	
60-00083	B

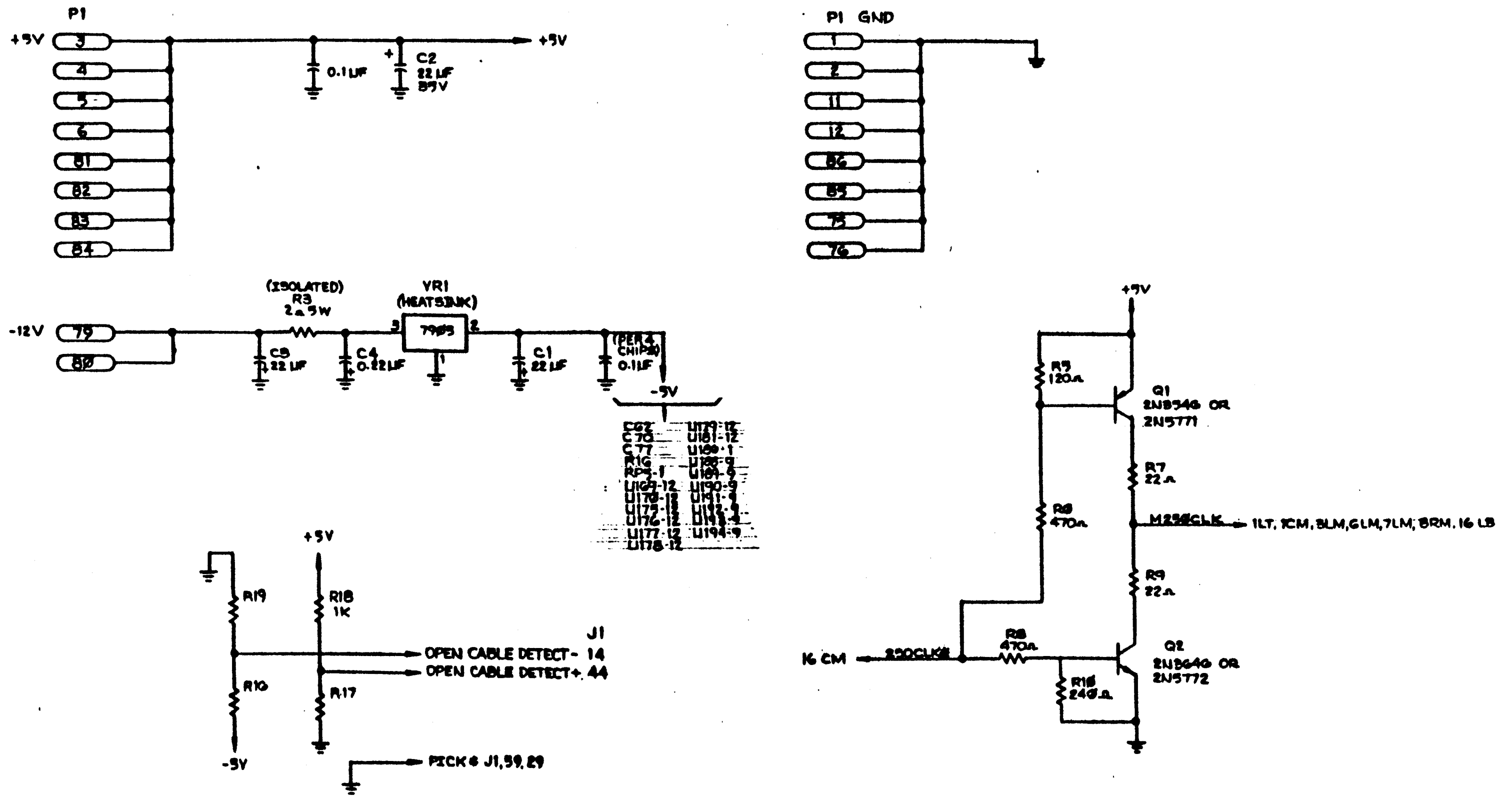


PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER (MISC I/O PORT)

SCHEMATIC: IMSC

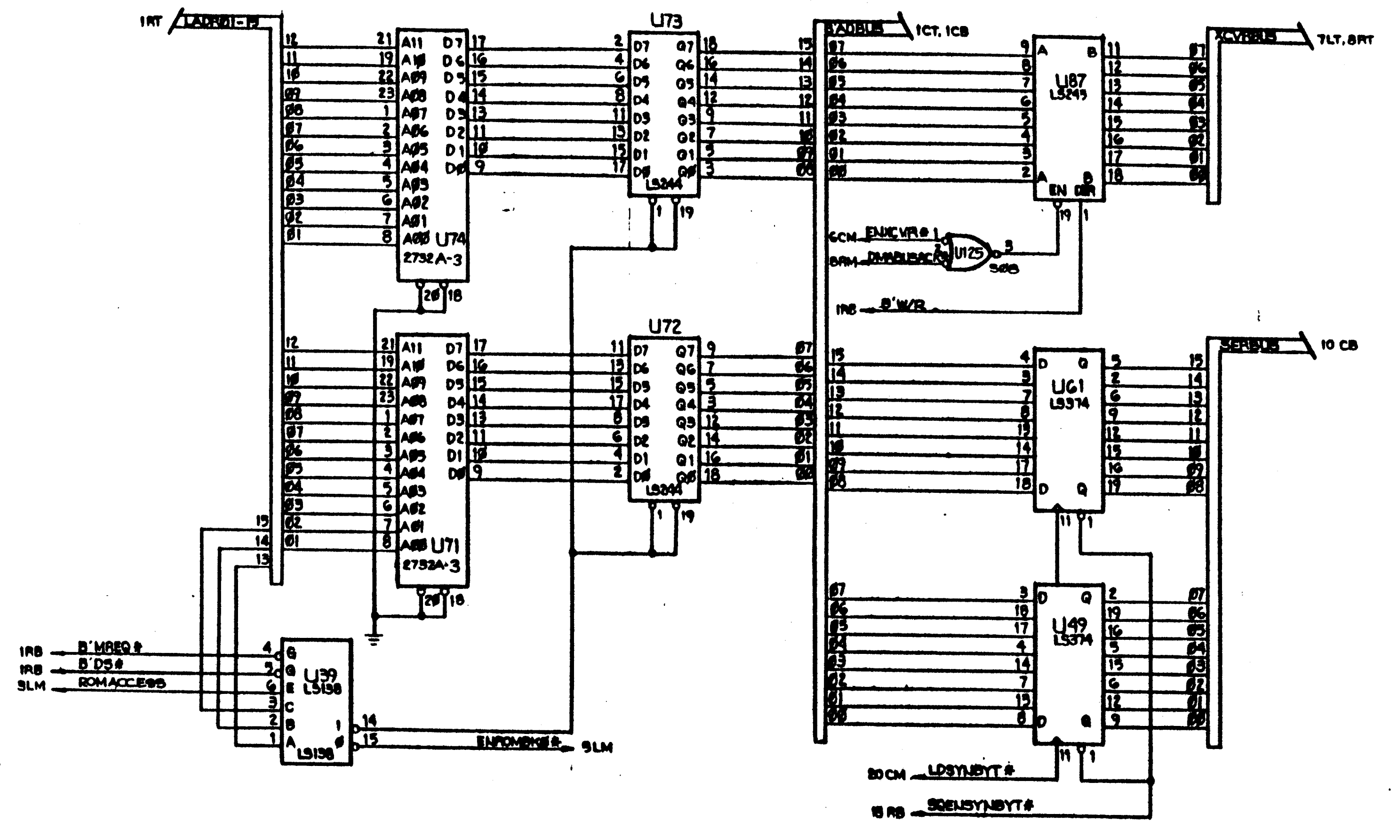
SIZE		REV
D	60-00083	B
	SCALE: ~ SHT 2 OF 29	



NOTES:

1. PULLUP NETS SHOULD BE AROUND 2Ø
2. MAINTAIN SHORT CLOCKLINES FOR 125CLK ∮ 125CLK PLO CLK, DKRDCLK ∮ BITCLK, BITCLK#
3. 825185, 825157, 2732 LOCATIONS SHOULD BE SOCKETED.
4. 4164 RAM LOCATIONS SHOULD BE SOCKETED AND TIGHTLY PACKED TO REDUCE TRACE LENGTH
5. BYPASS EVERY OTHER 4164 RAM LOCATIONS.
6. DO NOT RUN CLOCK LINES LENGTHWISE ACROSS THE CHIPS.

PLEXUS COMPUTERS INC.		
PLEXUS DISK/TAPE CONTROLLER (POWER CIRCUIT)		
SCHEMATIC: IMSC		
D	60-00083	B
SIZE	SCALE: — SHT 2 OF 2	REV



PLEXUS COMPUTERS INC.

PLEXUS DISK/TAPE CONTROLLER
(PROM BANK 1 & XCURBUS)

SCHEMATIC: IMSC

SIZE			REV
D	60-00083		B
	SCALE: ~	SHT 22 OF 23	

REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED
U1	4164	2		U37	LS85	17		U73	LS244	22	
U2	4164	2		U38	LS85	17		U74	2732A-3	22	
U3	4164	2		U39	LS138	22		U75	LS138	2Ø	
U4	4164	2		U4Ø	LSØ4	2,3,17,19	6	U76	S11	3,5,19	3
U5	4164	2		U41	LSØØ	17,19	4	U77	S74	6,17	2
U6	4164	2		U42	LS74	19	2	U78	LS51	8	2
U7	4164	2		U43	S109	8,19	2	U79	LSØØ	6,8,9	4
U8	4164	2		U44	LS28Ø	4		U8Ø	S257	14	
U9	4164	4		U45	LS245	2		U81	S257	14	
U1Ø	LS86	4,6	4	U46	LS158	2		U82	S299	1Ø	
U11	LS51	5		U47	LS245	2		U83	S299	1Ø	
U12	LS24Ø	18		U48	LS245	2		U84	932Ø	1Ø	
U13	LS24Ø	17		U49	LS374	22		U85	LSØØ	1Ø,11	4
U14	8287	18		U5Ø	LS28Ø	4		U86	S153	16	
U15	8287	18		U51	LS273	16		U87	LS245	22	
U16	LS24Ø	18		U52	LS32	2,4,5	4	U88	LS373	1	
U17	LS24Ø	18		U53	S74	1,4	2	U89	LS373	1	
U18	LS24Ø	17	2	U54	2732A-3	5		U9Ø	LS245	1	
U19	LS24Ø	19	2	U55	LS244	5		U91	LS245	1	
U2Ø	SØ4	5,18,19	6	U56	LS244	5		U92	LS366	1	
U21	4164	2		U57	2732A-3	5		U93	LS365	1	
U22	4164	2		U58	LS161	17		U94	LS244	8	
U23	4164	2		U59	LS08	17,19	4	U95	S74	5	2
U24	4164	2		U6Ø	S74	19	2	U96	LSØ2	5,17	4
U25	4164	2		U61	LS374	22		U97	S74	5	2
U26	4164	2		U62	LS374	1Ø		U98	LS191	14	
U27	4164	2		U63	LS374	1Ø		U99	LS191	14	
U28	4164	2		U64	LS244	1Ø		U1ØØ	LS191	14	
U29	4164	4		U65	LS374	1Ø		U1Ø1	LS191	14	
U3Ø	LSØ2	4,5,11	4	U66	LS374	1Ø		U1Ø2	LS374	1Ø	
U31	LS11	4,5	3	U67	LS244	1Ø		U1Ø3	LS161	11	
U32	S257	1		U68	LS157	1Ø		U1Ø4	LS374	1Ø	
U33	S257	1		U69	SØ8	4,11,19	4	U1Ø5	LS374	1Ø	
U34	LS374	18		U7Ø	LS138	6	1	U1Ø6	S112	16	2
U35	LS373	17		U71	2732A-3	22		U1Ø7	Z8ØADMA	8	
U36	LS373	6		U72	LS244	22		U1Ø8	Z8ØØ1	1	

REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED
U1Ø9	S175	5		U145	LS1Ø	6	3	U181	MC3450	14	2
U11Ø	SØ8	5	4	U146	S74	6	2	U182	LS24Ø	2Ø	
U111	LSØ2	5	4	U147	LSØ2	3,5,6,8	4	U183	LS273	2Ø	
U112	LS191	12		U148	LS74	7,9	2	U184	LS645Ø	9	
U113	LS191	12		U149	LS373	1		U185			
U114	LS191	12		U15Ø	S374	12		U186	75188	2Ø	2
U115	LS191	12		U151	82S185	12		U187	75189	2Ø	2
U116	25LS2521	12		U152	82S185	12		U188	MC3453	15	
U117	LS273	11		U153	82S137	12		U189	MC3453	15	
U118	LS74	11,13	2	U154	82S137	12		U19Ø	MC3453	15	
U119	LSØ4	1Ø,11,13	6	U155	LS191	13		U191	MC3453	15	
U12Ø	S51	11,16	2	U156	LS139	13	1	U192	MC3453	15	
U121	SØ4	11,16	6	U157	LS299	13		U193	MC3453	15	
U122	LS244	8		U158	LS32	11,13,2Ø	4	U194	MC3453	15	
U123	LS191	8		U159	S112	16	1	U195	LSØ8	3,9	4
U124	Z8ØACTC	7		U16Ø	LS139	6,2Ø	2	U196	7417	6,9	5
U125	SØ8	6,22	4	U161	Z8ØASIO	7					
U126	LS32	3,6,19	4	U162	S1Ø9	3		R1	47ØΩ	2,4	
U127	LS138	6		U163	LSØ8	3,9	4	R2	47ØΩ	2,4	
U128	SØØ	1,3,5	4	U164	LS74	3,9	2	R3	2Ω	21	
U129	SØ4	1,3,5,9,11	6	U165	LS51	3,9	2	R4	47ØΩ	6,17,19	
U13Ø	S139	3		U166	LS244	9,2Ø	2	R5	12ØΩ	21	
U131	LS174	12		U167	LS86	6,12	4	R6	47ØΩ	21	
U132	S174	12		U168	LS273	15		R7	22Ω	21	
U133	LS244	12		U169	MC345Ø	14		R8	47ØΩ	21	
U134	LS191	13		U17Ø	MC345Ø	14		R9	22Ω	21	
U135	S151	13		U171	LS273	15		R1Ø	24ØΩ	21	
U136	LS273	11		U172	S157	15		R11	47ØΩ	1,2,3,6,16	
U137	S374	13		U173	LS273	16		R12	47ØΩ	3,8,17	
U138	LS138	13		U174	LS373	6		R13	47ØΩ	1,7,13,16	
U139	S74	11	2	U175	MC345Ø	15		R14	47ØΩ	1,3,6,19	
U14Ø	LS1Ø	11,18	3	U176	MC345Ø	15		R15	47ØΩ	15	
U141	LS51	8	2	U177	MC345Ø	15		R16		21	
U142	LSØ4	1,3,8,9	5	U178	MC345Ø	15		R17		21	
U143	Z8ØACTC	7		U179	MC345Ø	15		R18		21	
U144	S138	6		U18Ø	LS24Ø	14		R19		21	

REF. DES.	COMP. I.D.	PAGE NO.	GATES NOT USED	REF. DES.	COMP. I.D	PAGE NO.	GATES NOT USED
R20	15ØΩ	8					
R21	47ØΩ	7,8,11,13,14,2Ø					
RP1	4.7K	1,8					
RP2	4.7K	1					
RP3	22ØΩ	9					
RP4	33ØΩ	9					
RP5	4.7K	14					
RP6	1K	1,14					
RP7	4.7K	6					
RP8	1K	15					
RP9	4.7K	8,2Ø					
RP1Ø	56Ω	15					
RP11	56Ω	15					
RP12	47ØΩ	14					
RP13	82Ω	14					
RP14	47ØΩ	14					
RP15	82Ω	14					
RP16	56Ω	15					
RP17	56Ω	15					
RP18	56Ω	15					
RP19	82Ω	15					
RP2Ø	47ØΩ	15					
RP21	82Ω	15					
RP22	47ØΩ	15					
RP23	82Ω	15					
RP24	47ØΩ	15					
RP25	47ØΩ	15					
RP26	82Ω	15					
RP27	47ØΩ	15					
RP28	82Ω	15					
RP29	4.7K	6					
RP3Ø	22ØΩ	9					
RP31	33ØΩ	9					

PARTS LIST

NEXT ASSY	USED ON
APPLICATION	DO NOT SCALE DRAWING

APPROVALS | DATE

PLEXUS COMPUTERS INC

SCHEMATIC · IMSC

D | 60-00083 | B

SCALE — | SHEET 23 OF 23