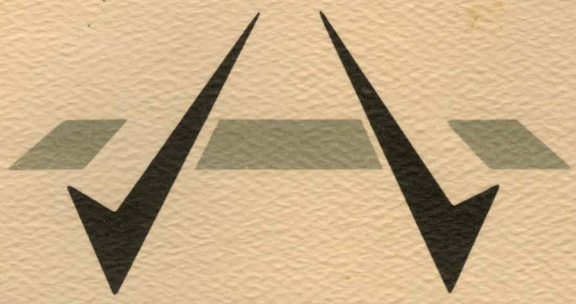


MiniFrame™ Hardware Manual

Convergent Technologies™

Data Systems Division

MiniFrame Hardware Manual

First Edition (February 1984) B-09-00411-01

MiniFrame™ Hardware Manual

CAUTION

This equipment generates, uses, and can radiate radio frequency energy. If this product is not installed and used in accordance with directions in the MiniFrame Installation Manual, it may cause interference to radio communications. It has been tested and found to comply with the limits for Class A computing devices, pursuant to Subpart J of Part 15 FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Nevertheless, operation of this equipment in a residential area is likely to cause radio interference; if this occurs the user, at his own expense, is required to make any necessary corrections to stop such interference.

Specifications Subject to Change

CTIX is derived from UNIX System V by Convergent Technologies under license from AT&T. UNIX is a trademark of Bell Laboratories. Convergent Technologies, Convergent, CTIX, Convergent Technologies Programmable Terminal, and Convergent Technologies Graphics Terminal are trademarks of Convergent Technologies, Inc.

First Edition (February 1984) B-Ø9-ØØ411-Ø1

Copyright © 1984 by Convergent Technologies, Inc.

PREFACE

This manual describes the Convergent Technologies MiniFrame Computer. It is written primarily for programmers, field service personnel, continuation engineers, and technical trainers. The manual comprises six sections and five appendices, as follows:

Section 1 provides an illustrated overview of the computer, including a list of the system features and options, a description of three possible system configurations, a physical description, and a brief functional description.

Section 2 describes and illustrates the processor board functional elements. It lays the groundwork for the remaining sections, but also is a nontechnical system description.

Section 3 describes the system main buses; provides an alphabetical listing, with brief descriptions, of the Main Processor board signals; and list the input/output connector pin assignments.

Section 4 presents a virtual memory overview, then describes the register set and DMA set up procedures.

Section 5 describes the functional elements of Section 2 at a circuit level. It references the schematics and includes state machine state sequencing diagrams. It provides necessary reference and background material for Section 6, thus freeing Section 6 to discuss system operations with minimal circuit description.

Section 6 describes system operations, including local transfers (nonperipheral), system input/output, and CPU interrupts. It includes timing diagrams.

Appendix A contains the programmable array logic (PAL) program listings and a description of how to decode the PAL equations. **NOTE:** Understanding how to decode the PAL equations is essential to understanding the computer's circuitry and system operation.

Appendix B contains the schematics for the Main Processor board and one Memory Expansion board. **NOTE:** The Main Processor board schematics are frequently referenced in Sections 3 through 6, and the reader must be familiar with the schematics to understand these sections.

Appendix C describes the power supply.

Appendix D describes Boot, the bootstrap ROM sequence.

Appendix E provides documentation on the disk drives.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

SECTION 1: OVERVIEW

The Convergent Technologies MiniFrame Computer is a powerful MC68Ø1Ø based computer that uses the CTIX operating system. With the maximum available storage, the computer has two megabytes of memory, a 5Ø-Mbyte hard disk, and a 64Ø-kbyte floppy disk. As system host, it can support a cluster of eight intelligent terminals via the RS-422 port; one line printer via the line printer interface; and a letter quality printer and modem via the RS-232-C A and B ports. An optional input/output expansion board is available to enhance the computer's input/output capability.

SYSTEM FEATURES

The MiniFrame Computer has the following features:

- Single board computer: the Main Processor (MP) board houses the CPU, 1/2 megabyte of memory, disk control, parallel line printer port, and communications ports.
- Multiuser demand-paged virtual memory CTIX operating system allowing four megabytes of virtual address space per process.
- Base memory expandable to two megabytes by adding one to three 1/2-Mbyte Memory Expansion (ME) boards.
- Mass storage on a 13-Mbyte, 26-Mbyte, or 5Ø-Mbyte hard disk drive and a 32Ø-kbyte or 64Ø-kbyte floppy disk drive.
- Two RS-232-C ports (A and B) and one RS-422 port.
- Centronix compatible parallel line printer interface.

SAMPLE SYSTEM CONFIGURATIONS

This subsection describes three possible MiniFrame Computer system configurations. The nondisk peripherals are optional and must be supplied by the user. The Programmable Terminal and Graphics Terminal are available from Convergent Technologies.

Small System Configuration

A small system configuration (for example, a standalone computer with a terminal) contains the computer plus peripherals, as follows:

- Main Processor board: CPU with 1/2 megabyte of memory.
- 10-Mbyte hard disk and 48-tracks-per-inch (TPI) 320-kbyte floppy disk.
- Line printer interface supporting a Centronix compatible printer.
- RS-422 port supporting one Programmable Terminal.

Figure 1-1 illustrates a small system configuration.

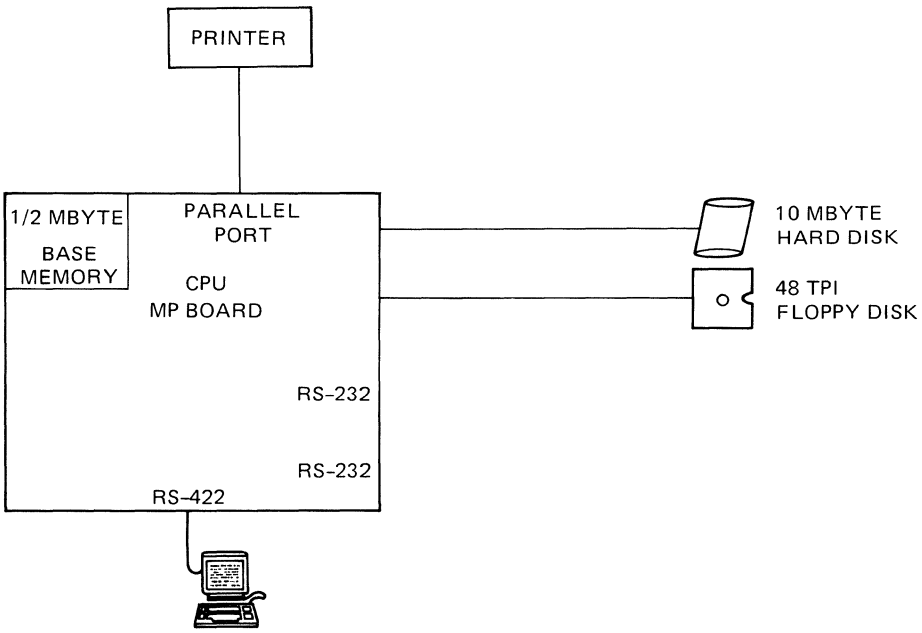


Figure 1-1. Small System Configuration

Medium System Configuration

A medium system configuration (for example, one supporting two to four users) contains the computer plus peripherals, as follows:

- Main Processor board: CPU with 1/2 Mbyte of memory.
- One Memory Expansion (ME) board supplying 1/2 of Mbyte additional memory.
- 20-Mbyte hard disk and 96 TPI 640-kbyte floppy disk.
- Line printer interface supporting a Centronix compatible parallel printer.
- RS-232-C port supporting a letter quality printer.
- RS-422 port supporting a combination of four Programmable Terminals and Graphics Terminals.

Figure 1-2 illustrates a medium system configuration.

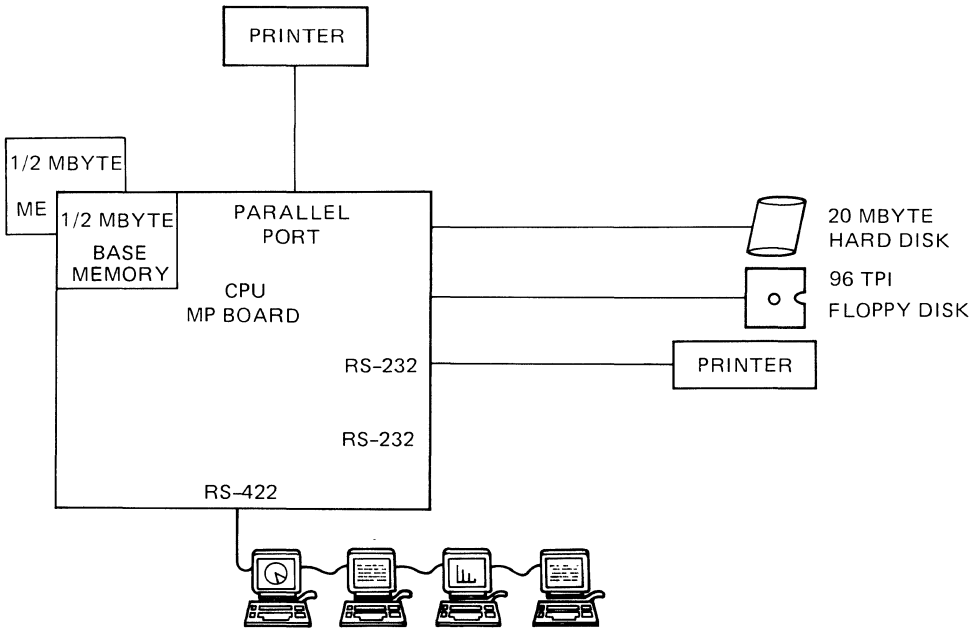


Figure 1-2. Medium System Configuration

Large System Configuration

A large system configuration (for example, one supporting four to eight users) contains the computer plus peripherals, as follows:

- Main Processor board: CPU with 1/2-Mbyte memory.
- One to three Memory Expansion (ME) board supplying between 1/2 and 1 1/2-Mbyte additional memory.
- 40-Mbyte hard disk and 96 TPI 640-kbyte floppy disk.
- Line printer interface supporting a high speed printer.
- RS-232-C A and B ports supporting a letter quality printer and modem.
- RS-422 port supporting a combination of eight Programmable Terminals and Graphics Terminals.

Figure 1-3 illustrates a large system configuration.

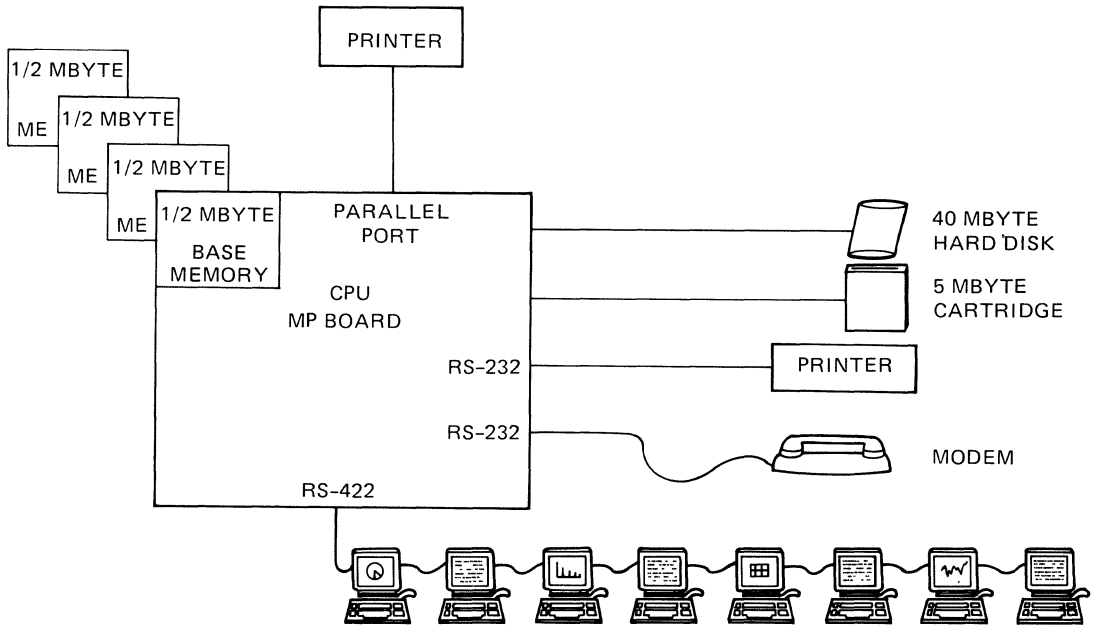


Figure 1-3. Large System Configuration

PHYSICAL DESCRIPTION

The MiniFrame Computer consists of a single enclosure with the following elements:

- One Main Processor board and zero to three Memory Expansion boards.
- Threaded-stud metal structure to support the Main Processor board and (if applicable) Memory Expansion boards.
- Two disk drives: one fixed media and one removable media.
- Power module.
- Cabling between the power module and the boards and drives. Expansion boards connected via zero insertion force (ZIF) connectors.

The Main Processor board is 15 by 18 inches. The Memory Expansion boards are 10 by 15 inches. All logic boards are four layers deep with complete uninterrupted ground planes.

As Figure 1-4 illustrates, the five status indicator lights, the on/off switch and power plug, the reset switch, and the port input/output connectors are all visible from the rear of the cabinet.

As Figure 1-5 illustrates, the Main Processor board mounts onto the swing-down side of the enclosure. The Memory Expansion boards attach to the Main Processor board, via the threaded studs and ZIF connectors. The hard disk mounts vertically at the lower front of the cabinet. The floppy disk drive mounts vertically at the top front of the cabinet. (That is, the opening for the floppy disk opening is aligned vertically at the front of the cabinet.)

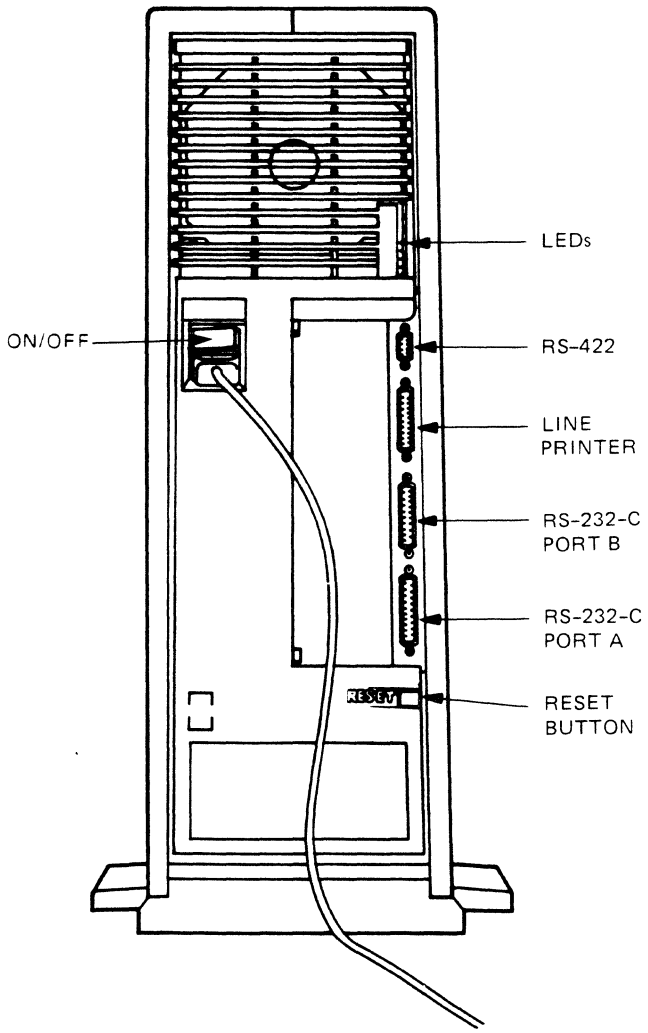


Figure 1-4. Rear of Computer Cabinet

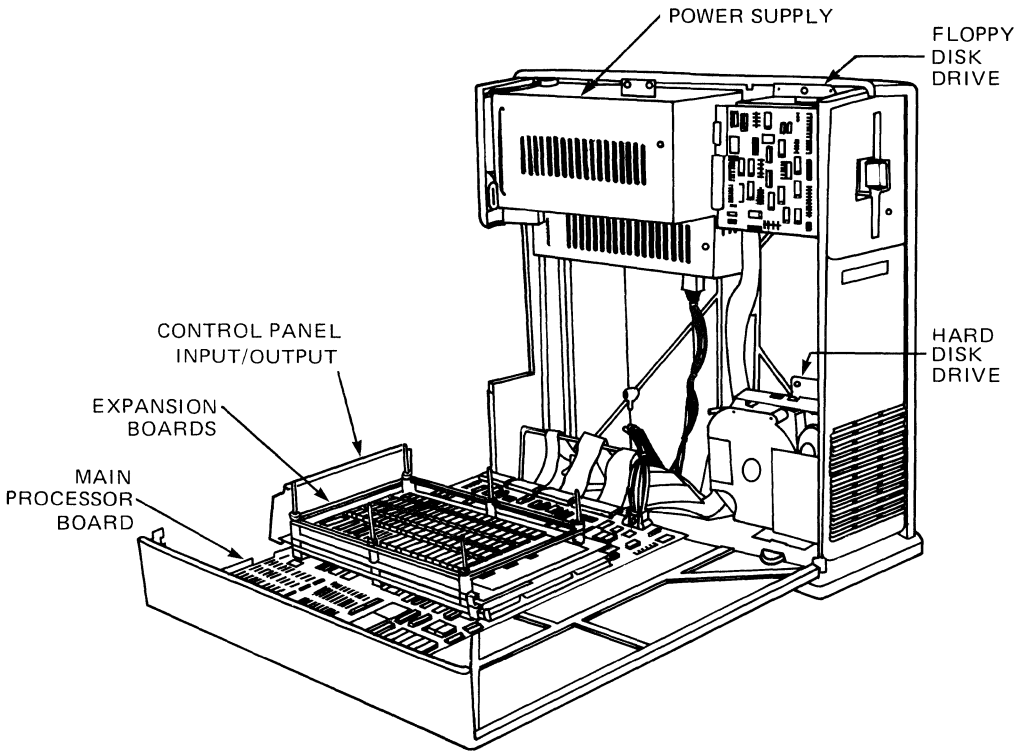


Figure 1-5. Inside of Computer Cabinet

FUNCTIONAL OVERVIEW

As Figure 1-6 illustrates, the computer includes six basic functional elements:

- System control
- Processor control
- Memory control
- Input/output
- Interrupt control
- Power system and distribution (not shown in Figure 1-6).

Appendix C describes the power system and distribution.

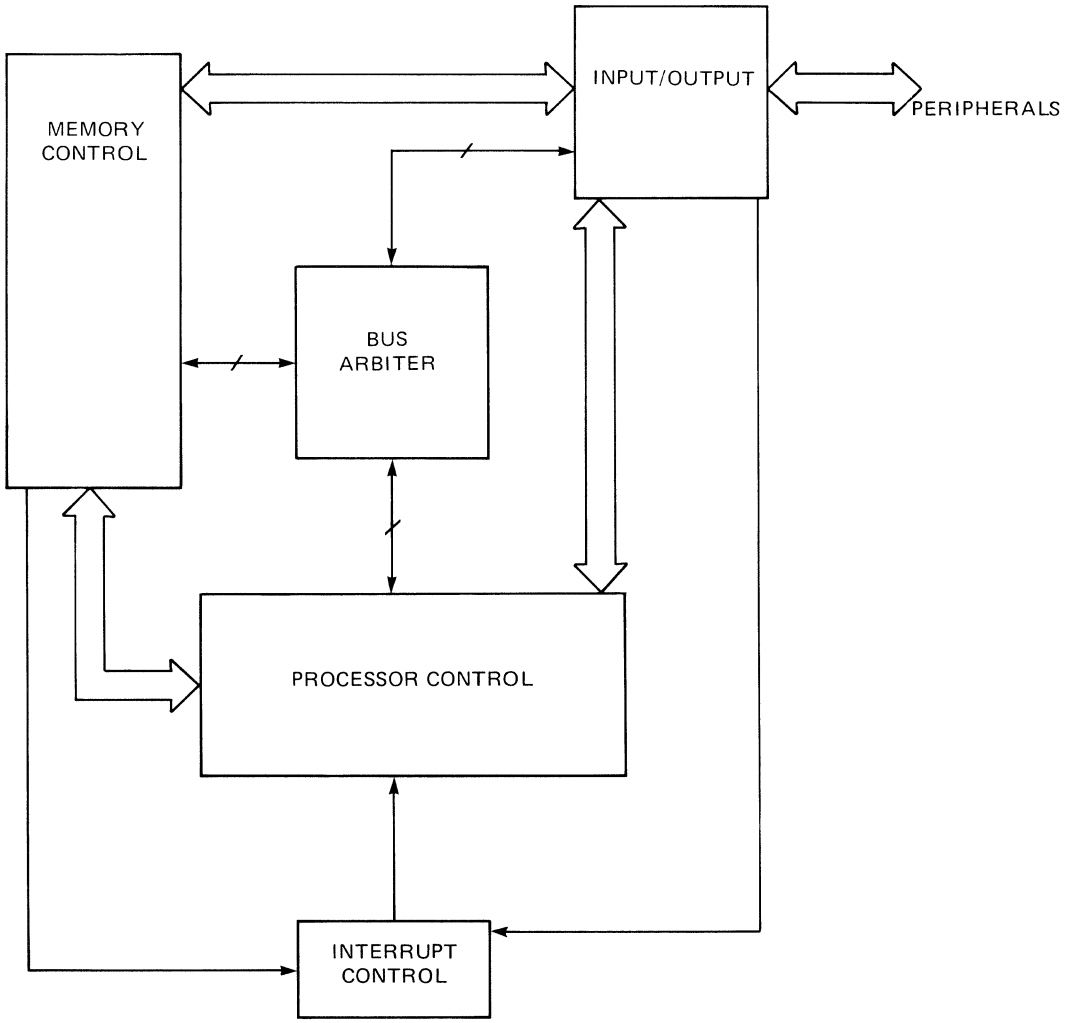


Figure 1-6. MiniFrame Simplified Functional Block Diagram

SECTION 2: FUNCTIONAL DESCRIPTION

This section describes the MiniFrame Computer functional elements. The computer, as illustrated in Figure 2-1, includes

- System control

  System clocks
  Bus control
  System reset
  Status indicator lights

- Processor control

  CPU
  CPU state machine
  Bootstrap ROM
  Processor buffers
  Processor address decode
  MiniFrame processor registers

- Memory control

  Map logic
  Memory access control
  Onboard and expansion RAM memory
  Parity logic
  Refresh logic
  Memory error control

- Input/Output

  Disk control
  Fast communications (RS-422) port
  Slow communications (RS-232-C) port
  Printer port
  External port (optional input/output expansion)

- Interrupt control

- Power (not shown in Figure 2-1).

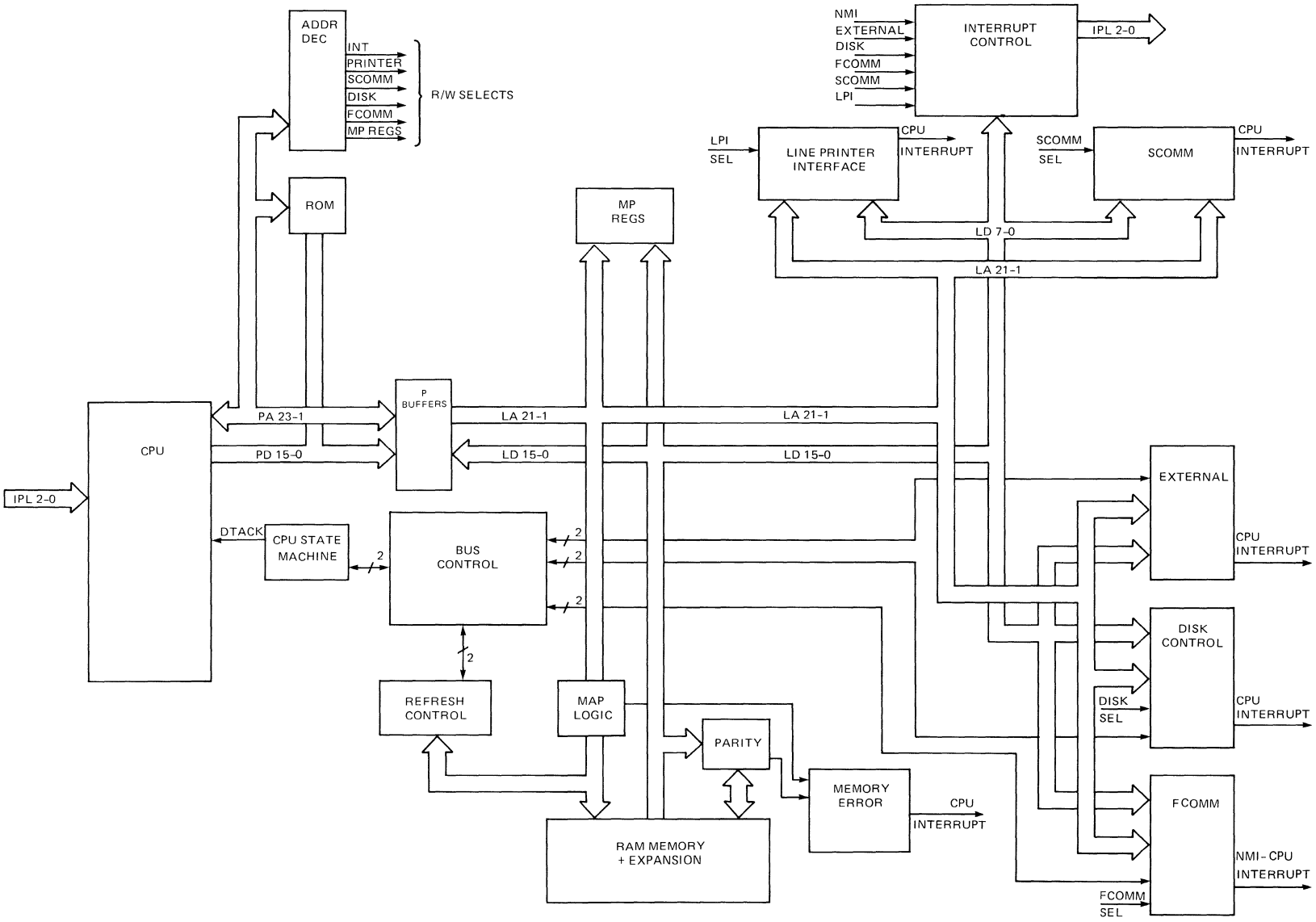


Figure 2-1. Detailed Functional Block Diagram

SYSTEM CONTROL

The system control directs and monitors overall system activity. It generates the system clocks, controls bus arbitration, performs system reset, and sets or resets under operating system (supervisor) control the system status indicators. **NOTE:** In supervisor (privileged) mode, the CPU controls system activity including DMA set up, processor register read/writes, map register accessing, and memory page allocation.

System Clocks

The computer has two 10-MHz system clocks, named PCLK and PCLK inverted, and two 20-MHz local clocks, named 20MHZ and 20MHZ inverted. The clocks trigger all synchronous system activites. The clock circuitry derives several additional clocks, for specific hardware, from the 20-MHz clock.

Bus Control

The bus control contains two elements:

- Bus arbiter
- Miscellaneous control PAL.

The bus arbiter directs local system activity (the local system is elements on the Main Processor and Memory Expansion boards) by receiving bus requests from the CPU and input/output elements and granting bus control according to the following priority scheme:

1. External port (optional)
2. Disk control
3. RS-422 fast communications port
4. Refresh control
5. CPU.

The external port has the highest priority, the CPU the lowest. The CPU requests bus control to perform memory accesses or register read/writes. The refresh control requests bus control to refresh the dynamic memory. All the other elements request bus control for DMA transfers.

The bus arbiter allows continuous local transfers to occur with no wait time between transfers; if a new request is pending at the end of a local transfer, the arbiter sends a new bus grant, and a new transfer begins. Input/output devices never perform two successive transfers, however, because of recovery time built into the request/grant protocol. The miscellaneous control PAL generates memory control and bus status signals.

System Reset

There are three ways to reset the computer: by software (the reset command), by hardware (the reset switch), or by a power on. A reset resets all the resettable hardware, including the CPU.

Bootstrap ROM

Upon a system power on or reset, the CPU automatically addresses ROM. The bootstrap program, Boot, directs the CPU to initialize the system and read in the operating software from the disk. Appendix D describes Boot.

PROCESSOR CONTROL

The processor control centers around the 32-bit Motorola MC68010 CPU and includes the CPU state machine, processor buffers, processor registers, processor address decode, and status indicator lights.

CPU

This manual provides minimal information on the MC68010 CPU; for complete information on the CPU, refer to the Motorola MC68010 specification; for a detailed description of the 68000 family instruction set, refer to the MC68000 16-bit Microprocessor User's Manual. The MC68010 specification provides the following information:

- Data organization in registers
- Addressing capabilities
- Instruction set
- Signal and bus operation
- Processing states
- Execution times.

NOTE

The information contained in the MC68010 specification is an essential reference to this manual. Consult Motorola for the availablility of this and other MC68010 or 68000 family documentation.

In the MiniFrame system, the MC68Ø1Ø performs seven basic cycles:

- CPU fast cycle and slow cycle read
- CPU fast cycle and slow cycle write
- CPU read-modify-write (test and set)
- CPU space (slow cycle interrupt acknowledge)
- Bus error
- Halt
- Reset.

CPU accesses to the processor registers, the map registers, or memory are fast cycle transfers and last 4ØØ ns. DMA transfers (between memory and either the fast communications port or the disk control) and memory refresh are also fast cycle transfers.

CPU accesses to the timers, the input/output device registers, or ROM are slow cycle transfers and last 1ØØØ ns. They are longer because of the slower response time of the addressed device.

The CPU is a multilevel interrupt device and thus requires a method of prioritizing and acknowledging interrupts. The CPU interrupt acknowledge cycle, also called the CPU space cycle, is a special type of slow cycle read which performs this function.

From the CPU's viewpoint, fast and slow cycle transfers are identical, except the CPU enters wait states during slow cycle transfers while it waits for an acknowledge signal. When doing the read-modify-write cycle, after the read portion, the CPU state machine releases control of the system bus to allow any pending DMA requests to proceed. The CPU does the write portion when the system is available for a CPU transfer.

The CPU executes a bus error cycle when a user attempts to access the Kernel, an input/output device register, or a write protected memory location. The CPU also executes the cycle when a user or the supervisor attempts to access a page not present. The bus error cycle is explained in the Motorola MC68Ø1Ø specification.

CPU State Machine

The CPU state machine is a state sequencer consisting of a single PAL. During CPU read/writes, the machine acts as a synchronizing interface between the CPU and the bus arbiter, memory control, and interrupt control.

Processor Buffers

Two sets of buffers, the processor address and processor data buffers, route the processor address and data buses to the logical address and logical data buses (the local system buses).

Processor Registers

The processor has the following five control and status registers:

- General Control register
- General Status register
- Clear Status register
- Bus Status register
- System Reset register.

Section 4, "Processor Registers," describes the registers.

Processor Address Decode

The processor address decode includes four PALs:

- Main address decoder
- CPU address decoder
- Two I/O address decoders

During CPU read/writes transfers, the main address decoder decodes the processor address. The decoder outputs a select signal to the element the CPU is addressing and, if necessary, enables the processor buffers. If the CPU accesses a processor or input/output device register, the main address decoder enables the CPU or I/O address decoders, and the decoder asserts the register read/write signal (or signals).

Status Indicator Lights

There are five status indicator lights, all visible from the back of the computer cabinet. Four lights are associated with bits in the General Control register. The supervisor turns these on and off. The other light is on when the computer is powered on.

MEMORY CONTROL

The memory control contains six major elements:

- Map logic
- Memory access control
- Base (onboard) and expansion RAM memory
- Parity logic
- Refresh logic
- Memory error control.

Map Logic

The system virtual address is 24 bits long. Virtual memory, which is four megabytes, comprises one quarter of the total addressable system space. All register or memory accesses must be even byte aligned, and the hardware sees only bits 23-1; bit Ø is not used and is always Ø.

During a memory access, after the system control determines that memory has been addressed (virtual address bits 23-22 = ØØ), the memory control receives the lower 21 bits of the virtual memory address, via the logical address bus. The upper ten bits (bits 21-12) address the map logic to select a page. The maps output a 9-bit logical mapped address (LMA2Ø-12), which, with the lower 11 bits of the virtual memory address (bits 11-1), forms the complete physical memory address.

Each user can address the full four megabytes of virtual memory. When a user starts a process in the computer, the supervisor sets up the map registers for that process, assigning the process to unused places in physical memory.

During each memory access, the map logic updates a table, called the page map table, to indicate the result of the access. The table records the status for each page in virtual memory, as follows:

- Not present (no memory available at that address)
- Present but not accessed (memory available, but it has not been accessed yet)
- Accessed but not written to (memory available which has been read, but not written)
- Written (memory available which has been written to and must be sent back to the disk before being overwritten).

Memory Access Control

The memory access control, triggered by the bus control, outputs control signals to the rest of the memory control. The memory access control includes a delay timer (also used in CPU/register accesses) and address decoders and multiplexers.

Main Memory

Main memory is sectioned off into pages 4Ø96 bytes long. The page address is the 9-bit latched map address, from the map

logic. The upper two bits of the latched map address select the board to be accessed; the lower seven bits select the page on the board. The Main Processor board (and each Memory Expansion board) contains 128 pages, or 512 Kbytes. The lower ten bits of the logical (virtual memory) address select the word on the page.

Parity Circuit

During memory writes, the parity circuit generates parity on the write data. The memory access control writes the parity into the parity RAM when it writes the data into the memory RAM. During memory reads, the parity circuit checks the parity of the data and generates a parity error signal if it is incorrect. If the supervisor has not masked out the error, the error generates a nonmaskable CPU interrupt.

Refresh Control

The refresh control contains the refresh request and refresh memory address generators. The control refreshes a row of memory every 13 microseconds; thus refreshing all of memory (including expansion) every 1.65 milliseconds (for RAMs that refresh 128 rows) or every 3.3 milliseconds (for RAMs that refresh 256 rows).

Memory Error Control

The memory error control responds to invalid memory accesses (user attempts to write to a write protected or operating system location) or errors during an access (parity or nonexistent memory address) by sending a bus error signal to the CPU or generating a nonmaskable interrupt request to the interrupt control. (Nonmaskable interrupts occur without bus errors, and bus errors occur without nonmaskable interrupts.)

DISK CONTROL

The disk control contains three elements:

- Disk DMA controller
- Disk bus interface unit
- Hard disk and floppy disk controllers.

Figure 2-2 illustrates the disk control in connection with the CPU, the bus arbiter, and memory control.

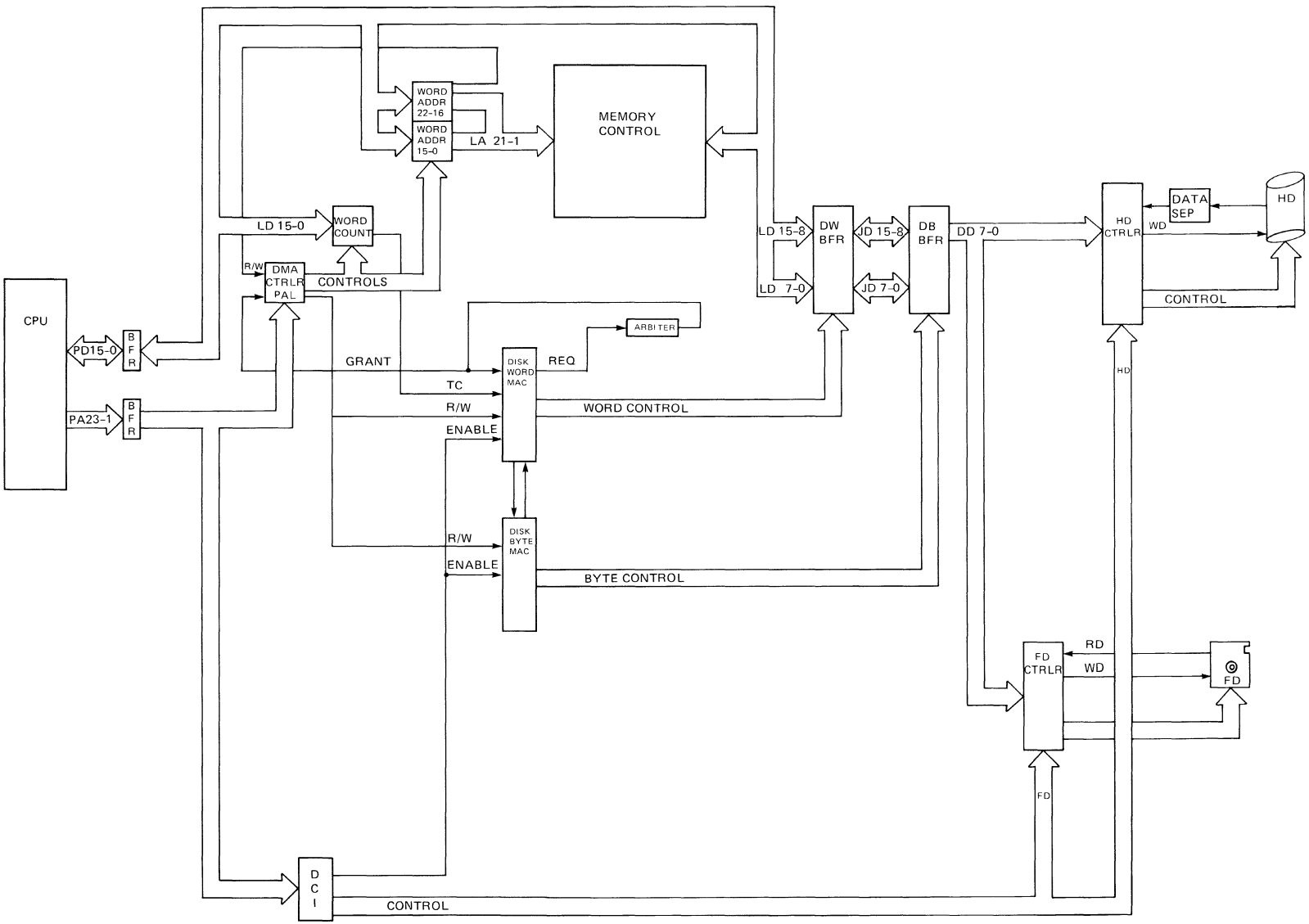


Figure 2-2. Disk Control Functional Block Diagram

Disk DMA Controller

The disk DMA controller stores and updates the memory address and word count during each disk DMA transfer and tells the disk bus interface unit when the DMA operation is finished. The controller consists of the disk DMA control PAL, the Disk DMA Word Count register, the Disk DMA Word Address registers, and disk terminal count flip/flop.

The disk DMA control PAL controls CPU accesses to the word count and word address registers, and it also updates the registers after each disk/memory transfer. After the disk control transfers the last word to or from memory, the word count register generates a terminal count signal that sets the terminal count flip/flop. The flip/flop alerts the interface unit to the end of operation status.

Disk Bus Interface Unit

The disk bus interface unit is the interface between the disk controllers and the CPU and memory control. It has four primary functions: (1) to provide an input/output data path between the controller registers and the CPU, (2) to service DMA requests by the controllers during DMA operations, (3) to provide a DMA data path between the controllers' data registers and memory, and (4) to mask out pending interrupts from a disabled controller.

The disk bus interface unit includes:

- Disk Control register
- Disk interrupt PAL
- Disk word/byte control.

Disk Control Register

The Disk Control register enables or disables DMA transfers, resets the disk controllers, turns the floppy disk motor on or off, selects single or double density for the floppy, and selects either the hard disk or the floppy disk for the DMA operation.

Disk Interrupt PAL

The disk interrupt PAL routes data requests from the disk controllers to the disk bus interface unit. It also routes the hard disk and floppy disk controller interrupt requests to the interrupt control, while at the same time blocking an interrupt from a deselected controller.

Disk Word/Byte Control

The disk word/byte control contains the disk word state machine and word buffer and the disk byte state machine and byte buffer. The disk state machines, comprised of sequencer PALs and decoder PALs, defines control states for the disk control.

Different combinations of inputs determine which state a machine is in (the word machine can be and generally is in a different state than the byte machine), and during each state the machines assert or negate none, one, or more outputs.

Inputs to the machines come from the Disk Control register, the disk DMA controller, the main address decoder, and the bus arbiter. Outputs go to the disk word and disk byte buffers, the Disk Control register, the bus arbiter, the disk controllers, and the interrupt control. The word sequencer and byte sequencer each outputs a control field which defines the current state of the word and byte machines, respectively.

The disk word state machine controls the interface between the disk word buffer and the memory control (during DMA), between the disk word buffer and the CPU (during CPU/register accesses), and between the disk word buffer and the disk byte buffer (during both DMA and register accesses).

The disk byte state machine controls the interface between the disk byte buffer and the floppy disk controller and hard disk controller. The disk byte machine also responds to overflow and underrun errors by generating a CPU interrupt and stopping the transfer.

During a disk read DMA operation, the selected disk controller begins reading in the serial data stream from the disk drive and alerts the disk byte state machine when it has a byte ready to transfer. The disk byte machine routes the byte into the upper byte buffer. The same procedure is repeated for the lower byte buffer, then the disk byte machine alerts the disk word machine that a word is available.

The disk word machine takes the word and arbitrates for bus control while the disk controller and disk byte machine reload the byte buffers. When the disk word machine receives a bus grant, it sends the word to memory. The disk word machine must unload the word buffer at or before the time the disk byte machine refills the byte buffer to avoid an overflow error.

During a disk write DMA operation, the disk word machine arbitrates for bus control and, when it receives control, reads a word from memory into the word buffer. It notifies the disk byte machine and clocks the word into the byte buffer.

The disk word machine does another memory read to reload the word buffer, while the disk byte machine writes the word to the disk controller, one byte at a time. An underun error occurs if the byte buffer is empty and the disk controller requests a byte.

If a disk DMA operation results in an overflow or underrun error, the disk word state machine goes to the idle state, and the disk byte machine goes to the interrupt state. In the interrupt state, the disk byte machine generates a nonmaskable interrupt request, which the interrupt control routes to the CPU. When the CPU acknowledges the interrupt, the disk byte machine returns to the idle state.

During CPU/register accesses, the address decode and a portion of the processor address are input to the disk state machines and cause them to execute a one word transfer to or from the disk controller. The transfer is the same as DMA, except the hard disk controller becomes a bus slave instead of a bus master. The floppy disk controller is always a slave device. The state diagrams in Section 5 illustrate the state machines' sequencing during DMA and CPU/register accesses.

Disk Controllers

The disk controllers control the interface to the disk drives, including drive and command select, addressing, serial input/output, timing, and other drive related functions. The controllers also interface to the disk bus interface unit for byte transfers. Finally, the controllers generate CPU interrupts at the end of the DMA operations. There are two types of controllers: the hard disk controller, which is a bus master device; and the floppy disk controller, which is a bus slave device.

Hard Disk Controller

After the disk bus interface unit enables the hard disk controller for a disk read/write operation, the controller sends data requests to the unit shortly before it either sends a byte of data (DMA disk read) or latches the disk data bus contents (DMA disk read). Without waiting for an acknowledge, after roughly 400 ns the controller either assumes the unit has latched the read byte or itself latches the write byte.

In both cases, the hard disk controller removes the data request without prompting. In effect, the disk bus interface unit is the bus slave, and the hard disk controller is the bus master. If the disk bus interface unit is not ready for the transfer when the hard disk controller requests it, an overflow or underrun error occurs.

During register accesses, the disk bus interface unit sends control and address lines to the controller, causing the controller to read or write the addressed register. The byte and word buffers route the register data between the CPU and the disk controller.

Floppy Disk Controller

After the disk bus interface unit enables the floppy disk controller for a disk read/write operation, the controller is inactive (in the local system) until it receives a chip select from the interface unit. Then, when the controller is ready to read/write data, it returns a data request.

The disk bus interface unit returns a read enable or write enable, causing the floppy disk controller to output the read byte (DMA disk read) or open its input latches (DMA disk write). The disk bus interface unit removes the enable, then removes the chip select. The negated enable causes the floppy disk controller either to disable its disk data bus drivers or latch the write data.

In the just described sequence, the floppy disk controller always waits for a prompt from the disk bus interface unit, then when it receives the prompt, performs the next step. The disk bus interface has a predetermined ammount of time to send the prompt, but the floppy disk controller is unaware if the unit does or does not meet this constraint. The disk bus interface unit is thus the bus master and the floppy disk controller the bus slave.

During register accesses, the disk bus interface unit sends control and address lines to the floppy disk controller, causing the controller to read or write the addressed register. The byte and word buffers route the register data between the CPU and the disk controller.

FAST COMMUNICATIONS -- RS-422 PORT

The fast communications port operates as a secondary station transceiver, to allow other devices interfacing to the RS-422 bus (for example, intelligent terminals or another computer) to act as bus masters. When the fast communications port is receiving information, it acts as a bus slave; when it is sending information, it acts as a bus master.

Whenever any device on the port performs a transmission, the device commences the transfer by sending a secondary station address, which identifies the intended receiver. The receiver, which must already be receiver enabled, decodes the address, determines that the information is intended for it, and receives the information.

The RS-422 fast communications port contains:

- Fast communications DMA controller
- Fast communications bus interface unit
- RS-422 port controller.

Fast Communications DMA Controller

The fast communications DMA controller, identical to the disk DMA controller, stores and updates the memory address and word count during each fast communications DMA transfer and tells the fast communications bus interface unit when the DMA operation is finished.

The controller consists of the fast communications DMA control PAL, the fast communications DMA word count and word address registers, and the fast communications terminal count flip/flop. The DMA control PAL controls CPU accesses to the word count and word address registers and updates the registers after each disk/memory transfer.

When the bus interface unit has transferred the last word to or from memory, the word count register generates a terminal count signal, which sets the terminal count flip/flop. The flip/flop alerts the interface unit to the end of operation status.

Fast Communications Bus Interface Unit

The fast communications bus interface unit is the control and data interface between the CPU and memory control and the RS-422 port controller. It has three primary functions: (1) to provide an input/output data path between the port controller and the CPU, (2) to provide a DMA data path between the port controller and memory, and (3) to service DMA requests from the port controller during DMA operations.

The fast communications bus interface unit contains

- Fast Communications Control register
- Word/Byte control

Fast Communications Control Register

The Fast Communications Control register consists of a single PAL, that emulates several single-bit, write only registers, or

latches. Each latch has two addresses associated with it, a set address and a reset address. When set, a latch asserts the control signal associated with it. When reset, the latch negates the signal.

The CPU sets or resets the latches by writing to the set or reset address for the latch. (The CPU does not read the register.) During the CPU write, only the address bus is valid; the data bus is irrelevent. The address decode enables the PAL, and the PAL decodes a portion of the address field, then asserts or negates the corresponding control signal.

The register enables or disables fast communications DMA transfers, selects the transmit/receive clock, enables or disables the transmitter and receiver, and resets the RS-422 port controller.

Fast Communications Word/Byte Control

The fast communications word/byte control is very similar to and performs identical functions as the disk word/byte control, with the main exception that the fast communications word/byte control combines the word state machine and byte state into one machine, called the word/byte state machine. Also the fast communications word/byte control has only one data buffer, called the word/byte buffer.

Like the disk machines, the fast communications word/byte state machine is comprised of a sequencer PAL and decoder PALs. Also similarly, the word/byte machine defines control states for the fast communications port, during which states the machine receives combinations of inputs and asserts or negates one or more outputs. The outputs control bus arbitration, the routing of data between memory or the CPU and the RS-422 port controller, and the chip selects and read/write enables to the controller.

The machine receives inputs from the fast communications control register, the fast communications DMA controller, the device select PAL, and the bus arbiter. Outputs go to the fast communications word/byte buffers, the fast communications control register, the RS-422 port controller, the bus arbiter, and the interrupt control.

During a fast communications DMA operation, the word/byte machine controls the interface between the memory control and the RS-422 port controller, and, during CPU register accesses, between the CPU and the RS-422 port controller registers. Both transfers occur via the word/byte buffer. The operating system also allows for odd byte boundary memory transfers by having the CPU generate only one data strobe.

During CPU/register accesses, the disk/word state machine is activated by register select signals from the I/O address decoders, a portion of the processor address (via the logical address bus), and the processor read/write signal. The machine places the lower word/byte buffer in transceiver mode and disables the upper buffer for the register access, then routes the lower byte of register data to or from the CPU, via the fast communications data, logical data, and processor data buses. The lower byte bypasses the word byte control and goes between CPU and controller, via the processor data and logical data buses.

RS-422 Port Controller

The RS-422 port controller is an 8274 chip and interfaces to the fast communications bus interface unit as a slave device, the same way that the floppy disk controller interfaces to the disk bus interface unit. After alerting the bus interface unit that it is ready to do a transfer, the controller is passive and waits for the unit to assert and negate the read/write control signals.

SLOW COMMUNICATIONS PORT

The slow communications port controls the RS-232-C serial data ports (A and B). The main control, including serializing incoming and outgoing data, comes from the 8274 multiprotocol serial controller. (A 72Ø1 chip can substitute for the 8274.) Other elements include the programmable baud rate generator, external interface, and miscellaneous registers.

Unlike the fast communications port, which does DMA, the slow communications port operates in the local system only as a CPU interrupt device. When the port is ready to transfer data to or from memory, or when it wishes to alert the CPU to some error, command, or status condition, it interrupts the CPU. The CPU vectors to an interrupt routine, where it reads registers in the port controller to determine the reason for the interrupt.

On the basis of the information it reads, the CPU responds to the error/command/status condition or acts as the intermediary for a data transfer between memory and the port controller (by performing CPU/memory and CPU/port controller data register accesses).

The CPU may receive information telling it to set up a slow communications input/output operation to or from a peripheral; the CPU does several slow cycle writes to load the slow communications registers, then triggers the port by a final write to commence the operation.

Slow communications address decode determines when the CPU is accessing a slow communications port register. (This circuitry also determines if the CPU is writing the interrupt controller, in the interrupt control.)

There are three sets of slow communications registers:

- Programmable baud rate generator
- Slow Communications Clock/External Data register
- 8274 (or 72Ø1) internal registers

The CPU loads the baud rate generator with a divisor to obtain the BAUD rate transmit/receive clocks. The CPU can also choose to use external clocks by setting up the slow communications miscellaneous PAL to do so. This PAL also contains registers that output secondary transmit data. Refer to the 8274 (or 72Ø1) vendor's specification for a description of the port controller registers.

LINE PRINTER INTERFACE

The line printer interface routes data to the printer. Like the slow communications port, the line printer interface interrupts the CPU when it is ready to transfer data. The CPU then does a memory read and printer write. The port has an output latch. The CPU also reads status from the line printer interface, via an input buffer that connects incoming information to the logical data bus.

INTERRUPT CONTROL

The MC68Ø1Ø is a multilevel interrupt device with seven priority levels of interrupts. An interrupt control field, input to the CPU, alerts the CPU to the interrupt condition and also defines the priority level of the interrupt. Interrupts of a higher level preempt interrupts of a lower level, even if the CPU is servicing a lower level interrupt at the time of the higher level interrupt. This prevents random interrupt nesting.

The CPU contains a programmable priority level, or interrupt mask, of its own and, with one exception, ignores interrupts of a lower or equal level. The exception is when a device sends a level seven interrupt, in which case the CPU responds even if the CPU itself is set at level seven. The level seven interrupt is thus a non maskable interrupt. The supervisor changes the CPU's priority level frequently, by executing a single instruction.

The CPU responds to interrupts by executing an interrupt acknowledge cycle, where it outputs the priority level it is ackowledging on the address bus and indicates the interrupt

acknowledge, via the function code. The interrupt control has three principle elements:

- Interrupt controller A (ICA on schematics)
- Interrupt level generator
- Interrupt acknowledge PAL

Interrupt Controller

The interrupt controller, which controls normal vector interrupts, receives interrupt requests from various system elements and generates a single interrupt request to the interrupt control encoder. Before sending the interrupt request, the interrupt controller compares the active inputs with its own interrupt mask, which the supervisor loads. This allows the supervisor to block unwanted interrupts.

Interrupt Level Generator

The interrupt level generator receives CPU interrupt requests from the interrupt controller, the memory error PAL, and other system devices. The generator, which consists of a single PAL, determines the highest priority device requesting an interrupt and sends a 3-bit interrupt control field to the CPU, with the value of the field indicating the interrupt priority level.

If the CPU does not mask out the interrupt, the CPU responds via an interrupt acknowledge cycle, where the CPU indicates the selected interrupt level via the address field. The interrupt acknowledge PAL decodes the CPU response.

Interrupt Acknowledge PAL

The interrupt acknowledge PAL decodes the interrupt acknowledge cycle address field to determine which device the CPU is ackowledging; it then alerts the device by sending it an interrupt acknowledge signal. There are two types of CPU interrupts: autovector and normal vector.

If it is a normal vector interrupt, the interrupting device supplies the CPU with a vector address and (normal vector) acknowledge. The CPU uses the address to determine the location of the interrupt routine for that device.

If it is an autovector interrupt, after the device requests an interrupt and receives an interrupt acknowledge, it returns only an (autovector) acknowledge. The CPU supplies its own vector address, based on the interrupt level of the device.

The interrupt controller is a normal vector device. When it receives an interrupt ackowledge from the interrupt acknowledge PAL, the interrupt controller prioritizes the currently active interrupt requests and outputs the interrupt vector (previously loaded by the CPU) associated with its highest priority active input. The CPU vectors to the interrupt service routine.

The fast communications interrupts and optional expansion board interrupts are also normal vector. All other interrupts are autovector. When the CPU ackowledges an autovector interrupt, the interrupt control encoder returns the autovector acknowledge, and the CPU supplies its own interrupt vector.

During the interrupt routine the CPU should erase the condition causing the interrupt and, effectively, remove the interrupt. Upon completing the interrupt, the CPU resumes normal processing, unless a different unmasked interrupt is pending, in which case the CPU responds to that interrupt. **NOTE:** As the interrupt controller receives more than one interrupt request input, it may continue to assert its CPU interrupt even after the CPU has already serviced it.

SECTION 3: INTERFACE

This section describes the computer's buses, signals, and input/output connnector pin assignments.

BUSES

Table 3-1 describes the computer's main buses.

Table 3-1. Main Buses

Name	Acronym	Function
Disk Data	DD7-Ø	DD7-Ø is the data bus between the Disk byte buffer and the disk controllers.
Fast	FCD15-Ø	FCD7-Ø routes data between the port controller and the CPU or memory. FCD15-8 routes control and status to and from the CPU.
Interrupt Priority Level	IPL2-Ø	The interrupt level PAL outputs IPL2-Ø to request an interrupt. IPL2-Ø is the binary equivalent of the CPU interrupt priority level.
Data	JD15-Ø	JD15-Ø is the data bus between the disk word and disk byte buffers.
Latched Logical Address	LLA2-1	LLA2-1 are latched from the lower two bits of the logical address (LA2-1) when the map logic generates the page address (LMA2Ø-12). LLA2-1 goes to the memory control.

Continued

Table 3-1. Continued

Name	Acronym	Function
Latched Map Address	LMA2Ø-12	LMA2Ø-12 is the memory page address that the map logic outputs to the memory control during memory accesses.
Line Printer Data	LPD7-Ø	LPD7-Ø is the data, control, and status bus between the line printer interface and the logical data bus.
Logical Address	LA21-1	LA21-1 is the address bus for the local system. All memory and I/O transfers (except CPU/timer and CPU/ROM accesses) use this bus.
Logical Data	LD15-Ø	LD15-Ø is the data bus for the local system. All memory and I/O transfers (except CPU/timer and CPU/ROM accesses) use this bus.
Processor Address	PA23-1	PA23-1, the address bus from the CPU, goes to the ROM, system timer, and the logical bus. PA23-1 selects one of the eight megawords of the virtual address space.
Processor Data	PD15-Ø	PD15-Ø, the CPU data bus, goes to the ROM, system timer, and logical data bus.
Slow Communications Data	SCD 7-Ø	SCD7-Ø connects the slow communications port and the interrupt controller to the CPU via the logical data bus.
RAM Address	W, X, Y, and Z7-Ø.	These 8-bit address buses each address one of the four RAM banks on the Main Processor board. (The Memory Expansion boards have similar buses.) Each bus is simultaneously loaded with the row address (LA1Ø-3), then the column address (LMA18-11).

Continued

Table 3-1. Continued

Name	Acronym	Function
RAM Data	RD15-Ø	RD15-Ø is the data bus between RAM memory and the logical data bus. RD15-Ø also goes to the RAM parity ciruitry to allow the circuitry to generate parity or check for a parity error.
RAM Parity	RD17-16	RD17-16 contains the two byte parity bits for RAM data.

SIGNALS

Table 3-2 describes the computer's timing, control, error, and status signals.

Table 3-2 Signals

Acronym	Name and Function
AAS-	Accepted address strobe. The processor control asserts AAS- at 2ØØ ns during any normal input/output cycle (PA23 high) and also during an illegal ROM access (ROMEN- asserted and PA23 low). The control does not assert AAS- during a bus error cycle, because PA23 stays low. When the bus arbiter returns the slow cycle acknowledge, LIO-, the CPU completes the slow cycle access. If the arbiter does not return LIO-, AAS- stays asserted, and the CPU enters waits states until the arbiter asserts LIO-.
AS-	Address strobe. The CPU asserts AS- when the processor address is valid. AS- goes to the CPU state machine, the fast cycle latch, and other timing elements.
AT1-7	At time 1-7. During CPU input/output (slow) cycle transfers the at time sequencer (sheet 4) asserts AT1+ at 45Ø ns, then asserts each succeeding AT at 1ØØ ns increments to provide timing to other computer elements.

Continued

Table 3-2. Continued

Acronym	Name and Function
BACK-	Byte acknowledge. This is part of the BREQ/BACK handshake between the disk byte state machine and the disk word state machine. The disk byte state machine asserts BREQ- when it is ready to read or write a word from the disk word state machine. After receiving BREQ-, on the next clock, the disk word state machine asserts BACK- and the buffer read/write transfer control signals. The transfer occurs, and BREQ- and BACK- negate at the next clock.
BASEMEM-	Base memory. The memory address decode asserts BASEMEM- when LMA2Ø-19 addresses the Main Processor board memory.
BERR-	Bus error. The error control PAL (PAL 25) outputs this to the CPU when a memory access generates one of the following errors: • CPU/Memory page fault. • User attempt to write a device input/output or processor register. • The memory error monitor (PAL 3Ø) asserts MMUERR-. (See MMUERR-.)
BIURESET-	Bus interface unit reset. The CPU sets this Disk Control register latch to reset the disk bus interface unit.
BØROE-	Board Ø RAM output enable. BØROE- enables the Main Processor board memory read/write data buffers (sheet 21) during a memory access. There is a similar signal on each expansion board.
BREQ-	Byte request. The disk byte state machine asserts BREQ- when it is ready to transfer a word to or from the disk word state machine. (See BACK-, above.)
CASACK-	Column address strobe acknowledge. When the memory control on a Memory Expansion board has been addressed for a memory access, it asserts CASACK- after it receives the column address strobe signals UCAS- and LCAS-. If the expansion board doesn't assert CASACK-, the error control PAL (PAL 25) aserts MNP-.

Continued

Table 3-2. Continued

Acronym	Name and Function
CLK+	Clock. (CPU input.) Main system clock. 10-MHz square wave derived from PCLK.
CLRRAS-	Clear RAS2-. CLRRAS- resets the RAS2- flip/flop and on a refresh, blocks the map logic page table update. The memory control PAL (PAL 27) asserts CLRRAS- in the middle (180 ns) of a refresh cycle or at the end (390 ns) of a DMA or CPU/memory cycle.
DBINT-	Disk byte state machine interrupt. The state machine sends DBINT- to the interrupt control when an overflow (disk read) or underrun (disk write) error occurs.
DISKTC+	Disk terminal count. The disk terminal count flip/flop asserts this signal when the disk control has finished the last word transfer between memory and the disk bus interface unit.
DSABLDMA-	Disable DMA (disk). The disk byte state machine asserts DSABLDMA- during a disk DMA operation if an overflow or underrun occurs or if the disk word state machine asserts BRESET-. DSABLDMA- causes DSKDMAEN- to negate on the next clock.
DSEL-	Disk control select. The address decode asserts DSEL- when the CPU addresses the Disk Control register (PAL 6), the disk word state machines, or either of the disk controllers.
DSKDMAEN-	Disk DMA enable. The CPU asserts or negates this Disk Control register latched output to enable (assert) or disable (negate) the disk word state machine and the selected disk controller.
DSKDMAWR-	Disk DMA write. The CPU asserts or negates this via a write to the Disk Control register. The CPU asserts DSKDMAWR- for disk write DMA (memory to disk) and negates it for disk read DMA.
DTACK-	Data transfer acknowledge. The CPU state machine asserts this to tell the CPU that the recipient of a read/write cycle or normal vector interrupt acknowledge cycle has finished the cycle. The CPU latches DTACK-, at the rising edge of PCLK. If it is a read (or interrupt acknowledge), the CPU latches the data 100 ns later.

Continued

Table 3-2. Continued

Acronym	Name and Function
EDTK-	Error data transfer acknowledge. EDTK- causes the CPU state machine to send DTACK- to the CPU. The error control PAL asserts EDTK- when a parity, memory not present, or page fault error occurs.
FC 2-Ø	Function code. (CPU output.) The CPU outputs this at the start of each CPU cycle to indicate the nature of the cycle as follows: user data (ØØ1), user program (Ø1Ø), supervisor data (1Ø1), supervisor program (11Ø), or CPU space (111). FC2 is assigned the signal name SUPV+, FC1 is assigned the signal name PROGRAM+ and is only used during a CPU space cycle.
FCHI-	Function code high. Memory error monitor (PAL 3Ø) asserts this signal when the CPU does an interrupt acknowledge cycle (FC 2-Ø = 111).
HALT-	Halt. (CPU input.) Wired with RESET- for a system reset.
HARIKARI-	The CPU reset instruction asserts HARIKARI- (system suicide), via the processor address decode. HARIKARI- generates a system reset, by asserting RESET-.
HDDMA-	Hard disk DMA. The CPU asserts HDDMA- to select the hard disk controller for disk DMA and negates it to select the floppy disk controller.
INTACLK-	Interrupt clock. When the address decode asserts INTADDR-, INTACLK- causes the interrupt acknowledge PAL (PAL 24) to assert one or two interrupt ackowledge pulses. Two pulses occur for the 8274, 8253, and 8259 chips.
INTACNTRE-	Interrupt acknowledge counter enable. The address decode asserts INTACNTRE- when the CPU commences a slow cycle. If the cycle is an interrupt acknowlede cycle, INTACNTRE- enables the interrupt acknowledge counter, which outputs INTACLK- to the interrupt acknowledge PAL.
INTADDR-	Interrupt address. The address decode asserts INTADDR- when the CPU commences an interrupt ackowledge cycle.

Continued

Table 3-2. Continued

Acronym	Name and Function
IOSEL-	Input/output device register select. The processor address decode asserts IOSEL- when the CPU has received bus control for an input/output device register access. A user cannot access these registers. (SUPV+ must be asserted.)
LIO-	Latched input/output cycle. The bus arbiter asserts LIO- to grant bus control to the CPU for an input/output (slow) cycle access.
LMEM-	Latched memory cycle. The bus arbiter asserts LIO- to grant bus control to the CPU for a memory, processor register, or map register fast cycle access.
LWT-	Latched write. Output by the CPU or input/output devices during a CPU or DMA access to indicate the type of transfer. LWT- high indicates a read, low indicates a write.
MMUSEL-	Memory management unit select. The processor address decode asserts MMUSEL- when the supervisor addresses the map registers (SUPV- is asserted and NPCYCLE+ is negated).
MPREGSEL-	Main processor register select. The processor address decode asserts MPREGSEL- when the supervisor accesses a main processor register.
NMI-	Nonmaskable interrupt. The error control PAL asserts NMI- when a nonmaskable interrupt condition occurs.
NPCYCLE+	Nonprocessor cycle. The miscellaneous logic PAL (PAL 36) asserts NPCYCLE+ during a DMA or refresh cycle. NPCYCLE+ disables the processor address decode.
PLLDS-	Parity lower data strobe. The miscellaneous logic PAL (PAL 36) asserts PLLDS- and PLUDS- (see below) when the current memory access is to base memory. On a read, PLLDS- enables the upper byte, and PLUDS- enables the lower byte. Parity on each byte is generated and checked separately. This signal has no function during a write.
PLUDS-	Parity upper data strobe. See PLLDS-.

Continued

Table 3-2. Continued

Acronym	Name and Function
PROCREG-	Processor register. The processor address decode asserts this when the supervisor addresses the processor or map registers.
PROGRAM+	Program. This is the middle order bit (FC1) of the CPU function code. When PROGRAM+ is high, the CPU access is to a program (code) location or is an interrupt acknowledge cycle (FC 2-Ø = 111).
PS+	Parity sense. The CPU sets or resets this General Control register bit to select the mode of parity. The CPU normally sets PS+ (high), which selects odd parity.
RAS+	Row address strobe. ORred RAS1- and RAS2-. RAS+ activates the memory access multitap delay for a CPU fast cycle access or DMA.
RAS1-	Row address strobe 1. The processor control outputs RAS1- to bus arbiter and memory control at 13Ø ns in a CPU fast cycle transfer. (PA23 is low.) RAS1- confirms CPU bus control (LMEM- asserted) and aserts RAS+.
RAS2-	Row address strobe 2. This is the DMA equivalent of RAS1-. After the bus arbiter selects an input/output device (or the refresh circuitry) for DMA, at 13Ø ns, the address decode asserts RAS2-. RAS2- asserts RAS+.
RESET-	Reset. (CPU input/output.) Wired to HALT and the reset button. A power on, pressing the reset button, or the reset instruction (see HARIKARI-) each resets the system.
ROE-	Row output enable. Asserted by RAS+, via the memory control PAL (PAL 25), at 17Ø ns into a DMA or CPU fast cycle access. ROE- is ANDed with BASEMEM- (and similar signals on the Memory Expansion boards) to assert BØROE- (or similar signal on expansion boards.)
ROMSEL-	ROM select. The processor address decode asserts ROMSEL- when the CPU reads ROM, as follows: (1) PA23-22 equals 1Ø during a supervisor slow cycle (SUPV- and LIO asserted) or (2) ROMEN- is asserted, and PA23-22 equals ØØ during a supervisor slow cycle.

Continued

Table 3-2. Continued

Acronym	Name and Function
SYNCDDREQ-	Synchronized disk data request. When the disk DMA controller PAL (PAL 13) receives a data request from the selected disk controller (FDDRQ- from the floppy or HDBCS- and either HDRE- or HDWE- from the hard disk), the PAL asserts DSKDRQ-. A latch synchronizes DSKDRQ- to PCLK- and outputs SYNCDDREQ- to the disk byte state machine. SYNCDDREQ- tells the machine that the controller is ready to read or write data.
SUPV+	Supervisor cycle. SUPV+ is the highest order bit (FC2) of the CPU function code. When SUPV+ is high, the CPU is executing a priviledged instruction that is not available to the user. An example of a suvervisor cycle is the interrupt acknowledge cycle.
UDS- & LDS-	Upper and lower data strobes. In conjunction with address strobe (AS-), the CPU asserts or negates UDS- and LDS- to indicate whether the upper byte (bits 15-8), lower byte (bits 7-Ø), or both bytes (bits 15-Ø) are valid on the data bus during a CPU read/write.
UIOER-	User input/output error. The processor address decode asserts this when the user attempts to access a processor, map, or input/output device register. UIOER- goes to the error control PAL (PAL 25).
UPT-	Update page table. The map logic outputs UPT- when the CPU accesses memory or an input/output device does DMA. UPT- updates the page map table to the new status of the page accessed.
VPA-	Valid peripheral address. The line printer control PAL (in an auxilary function) asserts this to tell the CPU that the interrupt acknowledge cycle currently being executed is an autovector interrupt cycle. This signal replaces DTACK-, which is the acknowledge during a normal vector interrupt acknowledge (and CPU read/write) cycle.
WRITE-	Read/write. (CPU output.) Defines the data bus transfer as a read (high) or a write (low).

CONNECTOR PIN ASSIGNMENTS

Table 3-3 lists the RS-422 connector pin assigments. Tables 3-4 and 3-5 list the RS-232-C ports A and B connector pin assignments. Table 3-5 lists the line printer interface connector pin assigments. In all the tables, pins not listed are not used. The bidirectional RS-422 port uses the J4 connector as an input connector when the port is a receiver and as an output connector when the port is a transmitter.

Table 3-3. RS-422 Connector J4 Pin Assignments

Pin	Input Signal (Receiver)	Output Signal (Transmitter)
6	/FCRCLK	/FCTXCLK
7	FCRCLK	FCTCLK
8	FCRXSI	FCTXSO
9	/FCRXSI	/FCTSXO

Table 3-4. RS-232-C A Port Connector J1 Pin Assigments

Pin	Signal	Input/Output	Pin	Signal	Input/Output
2	TXDA	Output	14	STDA	Output
3	RXDA	Input	15	TXCA	Input
4	RTSA	Output	16	SRDA	Input
5	CTSA	Input	17	RXCA	Input
6	DSRA	Input	20	DTRA	Output
7	GND		22	RIA	Input
8	CDA	Input	24	CLK OUT A	Output

Table 3-5. RS-232-C B Port Connector J2 Pin Assigments

Pin	Signal	Input/Output	Pin	Signal	Input/Output
2	TXDB	Output	14	STDB	Output
3	RXDB	Input	15	TXCB	Input
4	RTSB	Output	16	SRDB	Input
5	CTSB	Input	17	RXCB	Input
6	DSRB	Input	20	DTRB	Output
7	GND		22	RIB	Input
8	CDB	Input	24	CLK OUT B	Output

Table 3-6. Line Printer Interface Connector J3 Pin Assignments

Pin	Signal	Input/Output	Pin	Signal	Input/Output
1	LPDØ	Input/output	1Ø	GND	
2	LPD1	Input/output	11	GND	
3	LPD2	Input/output	12	LPPRESENT-	Input
4	LPD3	Input/output	14	LPSTROBE-	Output
5	LPD4	Input/output	15	GND	
6	LPD5	Input/output	17	LPBUSY	Input
7	LPD6	Input/output	21	LPNOPAPER	Input
8	LPD7	Input/output	22	LPSELECT	Input
9	GND				

SECTION 4: PROGRAMMING

The MiniFrame Computer is a virtual memory machine. Virtual memory refers to the address space seen by the processor. Regardless of the amount of primary or secondary storage in the system, the virtual address space is always at its maximum: four megabytes. Adding expansion memory improves system performance, yet retains the same address space and, therefore, software compatibility. Additionally, all registers (including processor, mapping, and input/output) are located in virtual memory.

MEMORY MANAGEMENT OVERVIEW

By means of the computer's demand-paged memory mapping scheme, each user has the use of the full four megabytes of virtual address space (excluding the lower 1/2 megabyte, for the operating system). The virtual memory is sectioned off into 1Ø24 pages, each 4Ø96 bytes long, and the pages reside in either primary (RAM) or secondary (disk) storage. From any user's viewpoint, these pages are simply available storage, and a major task of the operating system (supervisor) is to juggle pages between primary and secondary storage.

The supervisor uses the map registers to locate pages in physical memory. There are 1Ø24 map registers, one for each virtual memory page address. Since in real time the computer can only execute one process at a time, at any given time the map registers contain page addresses pertaining to only one user process. However, processes can share code or data; each time the supervisor reloads the maps, it can load them such that they point to both user-exclusive and user-shared memory.

The process working set is the set of pages that currently reside in physical memory. Whenever the CPU accesses a memory location, the memory management hardware determines whether or not the page addressed is in the process working set and, if it isn't, generates a page fault CPU interrupt. This invokes the supervisor. Page faults are common, as even with maximum available physical memory, the virtual address space is still twice the size of physical memory.

Upon receiving a page fault, the supervisor sets up a disk DMA operation to obtain the missing page from the disk, then, if

another process is available (free to run), the supervisor context switches and returns the CPU to user mode. The CPU runs the other process while the disk control transfers the page to memory. When the disk control finishes the DMA operation, it interrupts the CPU. The supervisor notes the updated process working set and, either now or later, returns the CPU to the original user process.

The amount of physical memory determines the process working set upper size limit. As memory size increases, the upper size limit increases, which improves system performance by reducing disk transfers. Table 4-1 is a system virtual memory map.

Table 4-1. System Virtual Memory Map

Starting Address	Contents	Ending Address
$000000	Mapped RAM area	$3FFFFF
$400000	Page mapping RAMs	$4007FF
$400800	Unused	$40FFFF
$410000	Processor registers	$4FFFFF
$500000	Unused	$7FFFFF
$800000	Boot ROMs	$801FFF
$802000	Unused	$BFFFFF
$C00000	Input/output device address space	$FFFFFF

NOTE

Memory below $80000 (1/2 Mbyte) in the virtual address space is reserved for the Kernel. A user attempt to access this virtual area causes the map logic to generate an invalid memory access error; however, the supervisor may complete the access in software by generating the null address (for example, $80000 becomes $C0000).

During memory or register accesses, the upper two bits of the virtual address select one of the following four groups:

- A23-22 = 00 addresses RAM.
- A23-22 = 01 addresses the processor or map registers.
- A23-22 = 10 addresses ROM.
- A23-22 = 11 addresses the input/output device registers.

Address bit 23 high selects a fast cycle (400 ns) access, low selects a slow cycle (1000 ns) access.

MAP REGISTERS ($400000 to $4007FF)

After the upper two virtual address bits have specified memory for an access, virtual address bits 21-12, via the logical address bus (LA21-1), address the map RAMs. The logical address points to one of 1024 entries in the page mapping RAM. The addressed map outputs a 12-bit field.

Nine of the map RAM outputs (P8-0) go to the memory access control to specify the physical memory page being accessed. These become latched map address 20-12 (LMA20-12). The other three outputs are page table status bits. Table 4-2 describes the map registers. Here's a map register summary.

D15	D14	D13	D12	D11	D10	D9	D8
WE+	PS1	PS0	-	-	-	-	P8

D7	D6	D5	D4	D3	D2	D1	D0
P7	P6	P5	P4	P3	P2	P1	P0

Table 4-2. Map Registers

Bit	Name	Function
15	Write (WE+) Enable	The write enable bit indicates the write status of a page. When WE+ = 0, a user attempt to write to the page generates an error. When WE+ = 1, writes are allowed. During a read or in supervisor mode, the memory error control ignores WE+.
14-13	Page Status (PS1-0)	The page status bits decode as 00 Page not present. 01 Present but not accessed. 10 Accessed but not written. 11 Written to (dirty). **NOTE:** The map logic does not check Kernel accesses for read/write protection, as the protection mechanisms are inactive during such accesses

Continued

Table 4-2. Continued

Bit	Name	Function
12-Ø9	Not used.	These bits are unused on a write and undefined on a read.
Ø8-ØØ	Page (P) Address	These bits specify the physical page to be accessed. A user attempt to access the lowest 1/2 MByte (Kernel) in virtual memory causes an error.

PROCESSOR REGISTERS

This subsection describes the five processor registers.

General Control Register ($45ØØØØ)

The RESET- signal, generated by either a power up, pressing the reset button, or a CPU write to the System Reset register, sets the General Control register to Øs. Table 4-3 describes the General control register. Here's a register summary.

D15	D14	D13	D12	D11	D1Ø	D9	D8
EE+	PIE+	PS+	ROMEN-	DS4	DS3	DS1	DSØ

Table 4-3. General Control Register ($45ØØØØ)

Bit	Name	Function
15	Error (EE+) Enable	Error enable enables or disables nonmaskable interrupts (NMI-) and bus errors (BERR-). EE+ = Ø disables NMI- and BERR-. EE+ = 1 enables NMI- and BERR-. (See General Status and Bus Status registers in this section and "Memory Error Control" in Section 5.)
14	Parity (PIE+) Interrupt Enable	Parity interrupt enable disables or enables interrupts from parity errors. PIE+ = Ø disables interrupts from parity errors. PIE+ = 1 enables interrupts from parity errors.

Continued

Table 4-3. Continued

Bit	Name	Function
13	Parity (PS+) Sense	The parity sense bit determines the mode of the parity ciruit. PS+ = Ø places the circuit in even parity mode; even parity accompanies data to memory. PS+ = 1 places the circuit in odd parity mode; odd parity accompanies data to memory. This bit is normally set to 1, but software can alter it during diagnostics to verify the correct operation of the parity circuit.
12	ROM Enable (ROMEN-)	ROM enable allows the CPU to address ROM with a memory address. A reset or power asserts ROM enable by clearing the General Control register. Even though the CPU outputs addresses $ØØØØØØ through $ØØØØØ6 after a reset, ROM enable causes the main address decode to interpret these as ROM addresses. ($8ØØØØØ through $8ØØØØ6). The bootstrap routine then negates ROM enable, allowing the CPU to access RAM.
11	Status (DS4-3, 1-Ø)	These bits control the status indicator lights visible through the rear of the computer cabinet. DS2, the middle LED, is on when power is on. DSØ goes to the lowest LED (nearest the computer cabinet base), while DS4 goes to the top of the array.

General Status Register ($41ØØØØ)

When a nonmaskable interrupt or bus error occurs, the General Status register and Bus Status register are latched to indicate the reason for the interrupt or error. Table 4-4 describes the General Status register. Here's a register summary.

D15	D14	D13	D12	D11	D1Ø	D9	D8
MNP-	LWT-	NPC-	PGF-	--	PIE+	PE+	UIE-

Table 4-4. General Status Register

Bit	Name	Function
15	Memory (MNP-) Not Present	The memory not present bit indicates the status of the previous memory cycle. MNP- = Ø indicates that a device attempted to access local or expansion memory, and there was no memory response. MNP- = 1 indicates there was a memory response. This bit generates a nonmaskable interrupt even if errors are disabled (EE- is reset).
14	Logical Write (LWT-)	The logical write bit indicates the type of the previous memory cycle. LWT- = Ø indicates a memory write. LWT- = 1 indicates a memory read.
13	Nonprocessor Cycle (NPC+)	The nonprocessor cycle bit indicates the type of the previous cycle. NPC+ = Ø indicates it was a processor cycle. NPC+ = 1 indicates it was a nonprocessor (DMA or refresh) cycle.
12	Page (PGF-) Fault	The page fault bit indicates if the previous memory cycle generated a page fault. PGF- = Ø indicates that a page fault occurred. PGF- = 1 indicates that one did not.
11	Not Used	This bit is not used.
1Ø	Parity (PIE+) Interrupt Enable	The parity interrupt enable bit, from the General Control register, enables or disables interrupts from parity errors. The CPU sets PIE+ high to enable interrupts from parity errors and resets it to disable parity interrupts.
Ø9	Parity (PE+) Error	The parity error bit indicates if the previous memory cycle generated a parity error. PE+ = Ø indicates no parity error occurred. PE+ = 1 indicates there was a parity error.
Ø8	User (UIE-)	The user input/output error bit indicates if the user attempted to access input/output device address space on the previous cycle. UIE+ = Ø indicates that the user did. UIE = 1 indicates that the user did not.

Clear Status Register ($420000)

The CPU writes to the Clear Status register to unlatch the General Status register. When the CPU writes the Clear Status register latches automatically unlatch. The Clear Status register receives no data; thus the contents of the data bus are irrelevent.

Bus Status Register ($430000 & $430001, $440000 & $440001)

The 4-byte Bus Status register records the state of the bus when a memory transfer error occurs by storing the levels of various bus grant lines, the 22-bit physical byte error address, and the levels of the data strobes. The register is latched at the same time as the General Status register, and is unlatched with the Clear Status register write. Tables 4-5 and 4-6 describe the upper and lower halves of the Bus Status register. Here's the register summary.

Upper

D15	D14	D13	D12	D11	D10	D9	D8
MMUERR-	DGNT-	FCGNT-	XGNT-	-	-	-	LDS-

D7	D6	D5	D4	D3	D2	D1	D0
-	-	A21	A20	A19	A18	A17	A16

Lower

D15	D14	D13	D12	D11	D10	D9	D8
A15	A14	A13	A12	A11	A10	A9	A8

D7	D6	D5	D4	D3	D2	D1	D0
A7	A6	A5	A4	A3	A2	A1	UDS-

Table 4-5. Upper Bus Status Register ($430000 to 430001)

Bit	Name	Function
15	Memory Management Unit Error (MMUERR-)	The memory management unit error bit indicates if the previous memory access generated an invalid memory access error. MMUERR- = 1 indicates no invalid memory access error occurred. MMUERR- = Ø indicates one of the following occurred: • The page was not present. • A user attempted to access the lowest 1/2 MByte of memory. • User attempted to write a page not write enabled.
14	Disk (DGNT-) Grant	The disk grant bit indicates if the disk control had bus control during the previous bus cycle. DGNT- = Ø indicates it did. DGNT- = 1 indicates it did not.
13	Fast (FCGNT-) Communi-cations Grant	The fast communications grant bit indicates if fast communications had bus control during the previous bus cycle. FCGNT- = Ø indicates it did. FCGNT- = 1 indicates it did not.
12	External Grant (EGNT-)	The external grant bit indicates if the external port had bus control during the previous cycle. EGNT- = Ø indicates it did. EGNT- = 1 indicates it did not.
11-9	Not used.	These bits are not used.
8	Lower (LDS-) Data Strobe	If the previous cycle was a processor cycle, the lower data strobe bit indicates if the CPU asserted the upper data strobe line. LDS- = Ø indicates the lower byte was accessed. LDS- = 1 indicates it was not.
7-6	Not used.	These bits are not used.
5-Ø	Address (A 21-16)	The address bits record the address that was on bits 21-16 of the logical address bus on the previous cycle.

Table 4-6. Lower Bus Status Register ($440000 to 440001)

Bit	Name	Function
15-1	Address (A 15-1)	The address bits record the address that was on bits 15-1 of the logical address bus during the previous cycle.
0	Upper (UDS-) Data Strobe	If the previous cycle was a processor cycle, the upper data strobe bit indicates if the CPU asserted the upper data strobe line. UDS- = 0 indicates the upper byte was accessed. UDS- = 1 indicates it was not.

System Reset Register ($460000)

When the CPU writes the System Reset register, a system-wide reset occurs for approximately 500 milliseconds. This resets all resettable devices, including the CPU, to their initial power on state. Like the Clear Status register, the System Status register automatically sets when the CPU writes it, and the data bus contents are irrelevent.

BOOTSTRAP ROM ($800000 to $801FFF)

The bootstrap ROMs contain three parts, as listed in Table 4-7.

Table 4-7 Bootstrap ROMs

Virtual Address	ROM Contents
$800000 to $800003	Initial system stack pointer
$800004 to $800007	Initial program counter
$800008 to $801FFF	Unassigned and available for code.

After a reset or during a power on, hardware clears the General Control register. When the register is cleared, the ROM enable bit (ROMEN-) is asserted. This allows the CPU to access ROM by outputting the reset vector addresses, $000000 and $000004. The first eight bytes of ROM vector the system to the normal ROM address space, whereupon the software negates the ROM enable, allowing RAM accesses to occur.

INPUT/OUTPUT DEVICE REGISTERS

The following are input/output device register read/write cycle programming restrictions:

- The 8253 timer chips cannot handle two successive input/output cycles, as this violates their read or write recovery time. For these chips, the programmer must guarantee that 700 ns elapse from the end of one input/output cycle to the start of the next input/output cycle. This, combined with the timing within the cycle, provides the 1 microsecond total recovery time needed by the 8253 chips.

- During disk DMA, the CPU cannot access either disk controller chip. Attempting to do so may cause the chip involved in the operation to generate a disk write overflow or disk read underrun error.

- All DMA transfers are 16-bit word transfers, aligned on the even byte; therefor they must be done only from 16-bit word boundaries within memory. Attempting to transfer on an odd byte address produces erroneous results.

The input/output device registers are the following:

- System timer
- Fast communications registers
- Slow communications registers
- Line printer interface port registers
- Disk control registers
- Interrupt control registers.

System Timer Registers ($C00000 to $C00006)

The System Timer registers, located on an 8253 chip, comprise two timers and one counter. Table 4-8 is a timer summary.

Table 4-8. Timer Registers

Address	Timer Element
$C00000	Timer 0
$C00002	Timer 1
$C00004	Counter 2
$C00006	Control word

A 76,800-Hz clock is input to timers 0 and 1, while the out pin of timer 1 is input to counter 2. Counter 2 decrements by two for every tick received from timer 1.

Counter 2 can be used to determine if the software has missed any time ticks by storing in memory a count of the number of ticks since counter 2 was initialized. The supervisor periodically compares this count with the count of ticks attempted by hardware to detect a difference.

The outputs of timers 0 and 1 can also generate interrupts. The supervisor programs Timer 1 to generate a periodic interrupt at a 60-Hz rate.

Fast Communications Port Registers

The fast communications port registers include the 2652 port controller registers, the fast communications bus interface registers, and the fast communications DMA controller registers.

Port Controller Registers ($C10000 to $C1000C)

Table 4-9 lists the CPU read/write addresses of the 2652 port controller registers. Any data transfers to or from the 2652 registers must have the data in the lower eight bits of the word. Refer to the 2652 data sheet for a description of these registers.

Table 4-9. 2652 Registers

Address	Register	Type
$C10000	Receive data/status register	Read only
$C10004	Transmit data/status register	Read/write
$C10008	Parameter control sync/address register	Read/write
$C1000C	Program control register	Read/write

Bus Interface Unit Registers ($C10020 to $C10034)

All of the fast communications bus interface unit registers are data independent latches; when the CPU writes one of the latches, a signal asserts or negates, depending on whether the CPU writes the assert address or negate address for the signal. Table 4-10 lists the assert and negate addresses and the bus interface unit signal that is asserted or negated.

Table 4-10. Bus Interface Unit Registers

Address	Register	Function
$C10020	2652 receiver enable	Asserts FCRXE-
$C10022	2652 receiver disable	Negates FCRXE-
$C10024	DMA enable	Asserts FCDMAEN-
$C10026	DMA disable	Negates FCDMAEN-
$C10028	307-kHz clock select	Asserts FCCLKSEL-
$C1002A	2-MHz clock select	Negates FCCLKSEL-
$C1002C	2652 transmitter enable	Asserts FCTXE-
$C1002E	2652 transmitter disable	Negates FCTXE-
$C10030	RS-422 transmitter output enable	Asserts FCCLKOE-
$C10032	RS-422 receiver input enable	Negates FCCCLKOE-
$C10034	2652 reset	Asserts FCRESET-

The state of these control signals after a reset (either power-on or by writing to address $C10034) is as follows:

- The 2652 receiver is disabled (FCRXE- is high).
- The 2652 transmitter is disabled (FCTXE- is high).
- The transmit clock is disabled (not present on cable).
- The clock rate is set to 307 kHz (FCCLKSEL- is low).
- DMA transfers are disabled (FCDMAEN- is high).

The CPU samples the RS-422 carrier detect signal by reading the 8274A Extended register. See the 8274A data sheet for description of the Extended register.

Fast Communications DMA Controller Registers ($C20000 to $C20008)

The DMA controller registers generate the memory address, maintain the word count, and determine the direction of transfers during FCOMM DMA transfers. Table 4-11 lists the CPU read/write addresses for the FCOMM DMA Registers.

Table 4-11. Fast Communications DMA Controller Registers

Address	Register	Function
$C20000	DMA word count register	Read/write. This CPU loads this register with the word count at the beginning of a DMA operation. The CPU reads the register for diagnostics or error recovery.

Continued

Table 4-11. Continued

Address	Register	Function
(cont)	DMA word count	The register allows up to 128K bytes (64K words) to be transferred.
$C20002	DMA lower word address register	Write only. The CPU loads the DMA lower word address register with the lower 16 bits of the starting memory address before commencing a DMA operation. All DMA transfers must start on an even byte address. The lower word address register increments after every transfer and, when it overflows, sends an increment signal to the upper word address register.
$C20004	Not used	This address currently has no function.
$C20006	DMA upper word address/select write register	Write only. To select a fast communications DMA write operation (memory to port), the CPU writes the high six bits of the starting memory address (virtual memory address bits 21-16) into the lower six bits of this register. The CPU writes the data via the lower five bits of the logical data bus, LD5-0), and the register write also asserts FCDMAWRT-.
$C20008	DMA upper word address/select read register	Write only. To select fast communications DMA read operation (port to memory), the CPU writes the high six bits of the starting memory address (virtual memory address bits 21-16) into the lower six bits of this register. The CPU writes the data via the lower five bits of the logical data bus (LD5-0), and the register write also negates FCDMAWRT-.

Setting up Fast Communications DMA

The proper sequence for setting up a DMA transfer is as follows:

1. Load the DMA counter with the 2's complement form of the number of words to be transferred.

2. Load the DMA lower word address register with the least significant 16 bits of the DMA word address. **NOTE:** the address is incremented from low memory to high memory.

3. Load the DMA high word address with the most significant five bits of the DMA word address. Write it to address $C20006 for a DMA write and address $C20008 for a DMA read.

4. Issue the proper command sequence to the 2652.

5. Enable a fast communications DMA transfer by writing to the fast communications BIU address $C10024.

The word count and word address registers are incremented after every DMA transfer. When the carry out from the counter is set (the counter clocks from $FFFF to $0000), the fast communications bus interface unit is notified and ends further memory transfers.

Slow Communications Port Registers

The slow communications port registers contain the RS-232-C port registers and the slow communications timer registers.

RS-232-C Port Registers

Table 4-12 lists the RS-232-C port registers. Refer to the 8274 data sheet for a description of the 8274 registers.

Table 4-12. RS-232-C Port Registers

Address	Register	Type
$C30000	Channel A data (8274 register)	Read/write
$C30002	Channel B data (8274 register)	Read/write
$C30004	Channel A command (8274 register)	Write only
$C30006	Channel B command (8274 register)	Write only
$C30008	Secondary transmit/clock select	Write only
$C30008	Port status	Read only
$C3000A	Reset (Initialize 8274)	Write only

The supervisor programs the 8274 to run in interrupt mode, supplying it with internal interrupt vectors for its 68Ø1Ø interrupt processing. If more than one 8274 exists in the system (such as on a I/O expansion board), use the interrupt priority input/output feature to prioritize interrupts within all 8274s. The Interrupt priority input (IPO) signal from the onboard 8274 goes to the expansion connector specifically for this use.

Tables 4-13 and 4-14 describe the secondary transmit/clock select register and port status register, respectively. Here's a summary of the secondary/transmit clock select register and port status register. **NOTE:** These registers have the same address. The secondary transmit/clock select is write only, and the port status read only.

Secondary transmit/clock select

D7	D6	D5	D4	D3	D2	D1	DØ
X	X	X	X	STDA	STDB	SELA	SELB

Port status

D7	D6	D5	D4	D3	D2	D1	DØ
RIA	DSRA	SRDA	RIB	DSRB-	SRDB-	FCC+	Ø

Table 4-13. Secondary Transmit/Clock Select

Bit	Name	Function
7-4	Not used.	These bits are not used.
3	Secondary transmit data A (STDA)	This bit associates with the A channel and drives pin 14 of the RS-232-C connector. The CPU can write data to an external source via this bit.
2	Secondary transmit data B (STDB)	This bit associates with the B channel and drives pin 14 of the RS-232-C connector. The CPU can write data to an external source via this bit.
1	Select clock A (SELA)	This bit selects between internal and external clocks for port A. SELA = Ø selects the internal clock. SELA = 1 selects the external clock. When the internal clock is selected, the same clock is supplied to the chip for both transmit and receive clocks.
Ø	Select clock B (SELB)	This bit selects between internal and external clocks for port B. SELB = Ø selects the internal clock. SELB = 1 selects the external clock. When the internal clock is selected, the same clock is supplied to the chip for both transmit and receive clocks.

Table 4-14. Port Status

Bit	Name	Function
7	Ring (RIA) indicator A	This is the ring indicator from channel A.
6	Ring (RIB) indicator B	This is the ring indicator from channel B.
5	Data (DSRA-) set ready A	This is the data set ready indication from channel A. DSRA = Ø indicates data is ready. DSRA = 1 indicates data is not ready.
4	Data (DSRB-) set ready B	This is the data set ready indication from channel B. DSRB = Ø indicates data is ready. DSRB = 1 indicates data is not ready.
3	Secondary received data A (SRDA)	This bit associates with the A channel and is driven from pin 16 of the RS-232-C connector. The CPU can read data from an external source via this bit.
2	Secondary received data B (SRDB)	This bit associates with the B channel and is driven from pin 16 of the RS-232-C connector. The CPU can read data from an external source via this bit.
1	Fast (FCC+) Communica-tions carrier detect	This is the fast communications carrier detect signal. When FCC = 1 it indicates a carrier is present. When FCC = Ø, it indicates a carrier is not present.
Ø	Floppy (FDP-) Disk Present	When low, FDP- indicates that a floppy disk drive is connected to the floppy disk port.

Baud Rate Timer Registers

Table 4-15 describes the baud rate timer registers.

Table 4-15. Baud Rate Timer Registers

Address	Register	Function
$C40000	Counter 0	Baud rate for Port A.
$C40002	Counter 1	Baud rate for Port B.
$C40004	Counter 2	Currently not used.
$C40006	Control word	Holds the 8253 control word.

The 8253 timer chip has three counters in it. The input to counters 0, 1, and 2 is a 1,228,800-Hz clock. This frequency divides to obtain the range of internally generated baud rates, from 19,200 baud to 110 baud, using either 16X or 1X clocks. The control word selects the clock divisor.

Line Printer Interface Registers

A peripheral line printer operates through an 8-bit parallel Centronix compatible interface. The line printer interface is interrupt driven and operates in byte mode transfers only. No logic looks at the contents of the data register; the hardware is unaffected by any special characters that may be in the data stream. Any implications for customization would reside in the software device driver which could be adapted to interpret the actual data stream. Table 4-16 lists the three line printer interface registers.

Table 4-16. Line printer Interface Registers

Address	Register	Type
$C50000	Data/status	Write/read
$C50002	Enable Interrupts	Write, address only
$C50004	Disable Interrupts	Write, address only

The line printer status bits are interpreted as follows:

D0 Not used
D1 Line printer buffer full (one byte buffer)
D2 Line printer not present
D3 Line printer out of paper
D4 Line printer selected
D5 Line printer busy
D6 Not used
D7 Not used.

Disk Control Registers

The disk control registers include the disk controller registers, disk bus interface unit registers, and disk DMA registers.

Disk Controller Registers

Table 4-17 lists the disk controller registers.

Table 4-17. Disk Controller Registers

Address	Register	Type
$C60000	Not used; high impedance bus	Hard disk
$C60002	Error flags/write precompensation cylinder	Hard disk
$C60004	Sector Count	Hard disk
$C60006	Sector Number	Hard disk
$C60008	Cylinder Number Low	Hard disk
$C6000A	Cylinder Number High	Hard disk
$C6000C	SDH	Hard disk
$C6000E	Status/Command	Hard disk
$C60010	Status/Command	Floppy disk
$C60012	Track	Floppy disk
$C60014	Sector	Floppy disk
$C60016	Data	Floppy disk

The hard disk controller chip (the WD1Ø1Ø) supports serial data trasfers at 5-Mbits per second; the floppy disk controller chip (the WD2797) at 25Ø kbits per second. Refer to the WD1Ø1Ø and the WD2797 data sheets for a full description of these chips. Both controllers use the same DMA channel. When a disk DMA is in progress, neither chip can be accessed.

Disk Bus Interface Unit Registers

The disk bus interface unit provides the interface between the disk controllers and the processor. All of the interface unit registers are data independent; when the CPU writes one of the registers, a signal asserts or negates, depending on whether the CPU writes the assert address or negate address for the signal. Table 4-18 summarizes the read/write addresses and the effect on the corresponding bus interface unit signal.

Table 4-18. Disk Bus Interface Unit Registers

Address	Register	Function
$C6ØØ2Ø	Floppy disk reset on	Asserts FDRESET-
$C6ØØ22	Floppy disk reset off	Negates FDRESET-
$C6ØØ24	Hard disk reset on	Asserts HDRESET-
$C6ØØ26	Hard disk reset off	Negates HDRESET-
$C6ØØ28	Floppy disk motor on	Asserts FDMOTORON-
$C6ØØ2A	Floppy disk motor off	Negates FDMOTORON-
$C6ØØ2C	Enable hard disk DMA transfer	Asserts HDDMA-
$C6ØØ2E	Enable floppy disk DMA transfer	Negates HDDMA-
$C6ØØ3Ø	Disable DMA transfers	Asserts DSABLDMA-
$C6ØØ32	Select single density, floppy	Negates FDDBLDEN-
$C6ØØ34	Select double density, floppy	Asserts FDDBLDEN-
$C6ØØ36	Reset disk bus interface unit	Asserts BIURESET-

After a power on reset, the state of the disk bus interface unit is as follows:

- The floppy disk controller reset line is asserted.
- The hard disk controller reset line is asserted.
- The floppy disk motor is off.

Disk DMA Controller Registers

The DMA controller registers, described in Table 4-11, generate the memory address, maintain the word count, and determine the direction of transfers during disk DMA transfers.

Table 4-19. Disk DMA Controller Registers

Address	Register	Function
$C80000	DMA Counter	Read/write. The CPU loads this register with the word count at the beginning of a DMA operation. The CPU reads the register for diagnostics or error recovery. The 16-bit word counter allows up to a maximum of 128K Bytes (64K words) to be transferred during one DMA operation. With current disks, the maximum is 8K bytes.
$C80002	DMA Lower Word Address register	Write only. The CPU loads this register with the lower 16 bits of the starting memory address before commencing a DMA operation. All DMA transfers must start on an even byte address. This register increments after every transfer and, when it overflows, sends an increment signal to the upper address register.
$C80004	Not used	This address is not used.
$C80006	DMA Upper Word Address register/ select disk write	Write only. To select a DMA disk write operation (memory to disk) the CPU writes the high six bits of the starting memory address (address bits 21-16) into the lower six bits of this register, via logical data 5-0. The CPU write asserts DSKDMAWRT-.
$C80008	DMA Upper Word Address register/ select disk read	Write only. To select a DMA disk read operation (disk to memory) the CPU writes the high six bits of the starting memory address (address bits 21-16) into the lower six bits of this register, via logical data 5-0. The CPU write negates DSKDMAWRT-.

Setting up Disk DMA

To set up a disk DMA operation, do the following:

1. Load the DMA counter with the 2's complement form of the number of words to be transferred.

2. Load the DMA address low word with the least significant 16 bits of the DMA word address. (The address is incremented from low memory to high memory).

3. Load the DMA address high word with the most significant five bits of the DMA word address by writing to address $C80006 for a DMA write or to address $C80008 for a DMA read.

4. Enable a hard disk DMA transfer by writing to the disk BIU address $C6002C or a floppy disk DMA transfer by writing to address $C6002E (in the case of a floppy disk DMA, be sure the floppy disk motor has been on for the proper amount of time).

5. Issue the proper command sequence to the selected device controller.

Programmable Interrupt Controller A (8259) Registers

Table 4-20 Lists the programmable interrupt controller registers.

Table 4-20. Programmable Interrupt Controller Registers

Address	Register
$C90000	Initialization Control Word 1
$C90002	Operation Control Word 1

ADDRESS SUMMARY

This subsection contains three tables and one figure. Table 4-21 repeats the system memory map. Table 4-22 is a summary of the processor register addresses. Table 4-23 is a memory map of the input/output device registers. Figure 4-1 shows the register summaries of the four processor registers that have individual bit definitions.

Table 4-21. System Virtual Memory Map

Starting Address	Contents	Ending Address
$000000	Mapped RAM area	$3FFFFF
$400000	Page mapping RAMs	$4007FF
$400800	Unused	$40FFFF
$410000	Processor registers	$4FFFFF
$500000	Unused	$7FFFFF
$800000	Boot ROMs	$801FFF
$802000	Unused	$BFFFFF
$C00000	Input/output device address space	$FFFFFF

Table 4-22. Processor Register Address Map

Address	Register
$410000	General Status register
$420000	Clear Status register
$430000	Bus Status register upper bytes
$440000	Bus Status register lower bytes
$450000	General Control register
$460000	System Reset register

Table 4-23. Input/Output Device Address Map

Starting Address	Registers	Ending Address
$C00000	Timer	$C00006
$C10000	Fast communications bus interface unit	$C10036
$C20000	Fast communication DMA controller	$C20008
$C30000	8274AB	$C30008
$C40000	8253B	$C40006
$C50000	Line printer	single
$C60000	Disk bus interface unit	$C60030
$C70100	8259B	$C70102
$C80000	Disk DMA controller	$C80008
$C90000	8259A	$C90002

General Status Register ($410000)

D15	D14	D13	D12	D11	D10	D9	D8
MNP-	LWT-	NPC-	PGF-	--	PIE	PE	UIE-

Bus Status Register Upper ($430000)

D15	D14	D13	D12	D11	D10	D9	D8
MMUERR-	DGNT-	FCGNT-	XGNT-	-	-	-	LDS-

D7	D6	D5	D4	D3	D2	D1	D0
-	-	A21	A20	A19	A18	A17	A16

Bus Status Register Lower ($440000)

D15	D14	D13	D12	D11	D10	D9	D8
A15	A14	A13	A12	A11	A10	A9	A8

D7	D6	D5	D4	D3	D2	D1	D0
A7	A6	A5	A4	A3	A2	A1	LDS-

General Control Register ($450000)

D15	D14	D13	D12	D11	D10	D9	D8
EE	PIE	PS	ROMEN-	DS4	DS3	DS1	DS0

Figure 4-1. Processor Registers Summaries

SECTION 5: CIRCUIT DESCRIPTION

This section describes the MiniFrame Computer circuitry. Signal names appear in uppercase. Active low signal names end with a minus sign (for example, PCLK-), active high signal names end with a plus sign (for example, PCLK+). Sheet numbers in parentheses following signal names or hardware elements refer to the Main Processor board schematic sheets (Appendix D.) **NOTE:** The schematics omit the plus signs in active high signal names.

SYSTEM CONTROL

The system control generates the system clocks, directs system bus arbitration, and generates the system reset.

System Clocks (Sheet 5)

Four clocks, PCLK+, PCLK-, 20MHZ+, and 20MHZ-, control system activity. PCLK+ and PCLK- are 10-MHz clocks, with 50-ns duty cycles. Two flip/flops, one for the 20MHZ clocks and the other for the PCLKs, and a 40-MHz oscillator generate the clocks. By definition, PCLK- trails PCLK+ by 50 ns, and the negative edge of 20MHZ- and the positive edge of 20MHZ+ further divide the 50 ns PCLK pulses in half. (See Figure 5-1). **NOTE:** The 20MHz clocks provide critical timing for the local system.

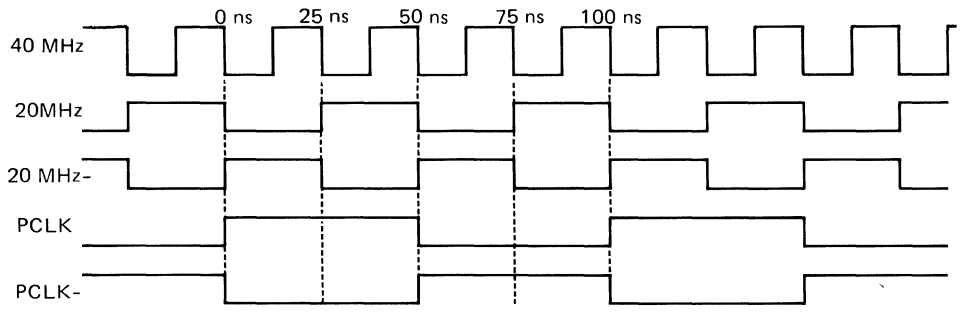


Figure 5-1. System Clocks

Bus Control

The bus control contains the bus arbiter and the miscellaneous logic PAL.

Bus Arbiter (PAL 5, Sheet 4)

The bus arbiter, clocked by PCLK-, controls system operation by deciding which of the input/output control elements (or the CPU) receives control of the local system for the next local transfer cycle. Control of the local system, other than CPU accesses to ROM and the system timers, means control of the logical data bus (LD15-Ø) and logical address bus (LA21-1) and use of the system control timing elements. (The CPU reads the ROM and read/writes the timer via the processor buses.)

The arbiter uses a request/grant handshake method to direct local system activity. Devices requesting bus control for a local transfer activate their bus request lines. The arbiter looks at the active bus requests just after the last PCLK- in the current transfer (the final 5Ø ns of the transfer).

At the next PCLK-, 5Ø ns into the new cycle, the arbiter selects the highest priority request asserted and grants bus control to the device making the request. If the system is inactive, the arbiter performs bus arbitration at each PCLK-.

For fast cycle transfers, the device receiving the grant does the transfer and removes the bus request at 36Ø ns. At the next PCLK- (45Ø, or 5Ø, ns), the arbiter removes the old grant and sends a new grant if a request is pending.

The CPU, with the lowest priority, actually has two different ways to gain bus control, one for a fast cycle and the other for a slow cycle. In both cases, regardless of what is happening with the rest of the system (excluding interrupts or a reset), when the CPU needs to do a read/write cycle, it begins the cycle, then does wait states until it receives a bus grant, whereupon it finishes the cycle.

If the CPU is doing a slow cycle transfer, at 2ØØ ns into the CPU cycle, the negative edge of PCLK- clocks AS-, which assert AAS-. AAS- is the CPU's slow cycle bus request to the arbiter. If no other bus request is pending, at the next rising edge of PCLK-, the arbiter sends the slow cycle grant, LIO-, to the processor control, and the CPU completes the slow cycle transfer. At about 98Ø ns, the processor control removes AAS-.

If AAS- is inactive, and no other requests are pending, at 5Ø ns the arbiter asserts LMEM-, the CPU default grant, and executes what is termed the idle cycle. If the CPU has begun a fast cycle

transfer, at 130 ns LMEM- (NANDed with PA23) generates RAS1-, the fast cycle acknowledge. RAS1- tells the arbiter to lock LMEM- low, while the CPU completes the fast cycle transfer. At 370 ns, the processor control removes RAS1-, freeing the system.

Table 5-1 lists in decreasing order the bus priority of the different elements and their corresponding request and grant signals. The default grant to the CPU is LMEM-, and the CPU acknowledge is either RAS1- (for a fast cycle transfer) or AAS- (for a slow cycle transfer).

Table 5-1. Bus Priority

Control Element	Priority	Request	Grant
External port	Highest	XDMAREQ-	XGNT-
Disk control	.	DSKDMAREQ-	DGNT-
Fast communications	.	FCDMAREQ-	FCGNT-
Refresh control	.	REFREQ-	REFGNT-
	.		
	.		
	.		
CPU (slow cycle)	Lowest	AAS-	LIO-
		Grant	**Ack.**
CPU (fast cycle)	Default	LMEM-	RAS1-

Miscellaneous Logic PAL (PAL 36, Sheet 4)

The miscellaneous logic PAL (PAL 36) is not clocked. The PAL generates two bus status signals, NPCYCLE+ and SETRAS2+, and three memory control signals, B0ROE-, PLLDS+, and PLUDS+.

NPCYCLE+ indicates that the bus arbiter has selected an input/output element for a DMA transfer. SETRAS2+, asserted with NPCYCLE+, sets the RAS2- flip/flop, which generates the fast cycle start signal, RAS2-, for the DMA transfer.

PLUDS+ and PLLDS+, generated from UDS- and BASEMEM-, are strobes for base (onboard) memory cycles. PLUDS+ indicates an upper byte access; PLUDS+ indicates a lower byte access. B0ROE-, active when the current memory cycle is to base (onboard) memory, enables the onboard RAM banks onto the logical data bus.

Similar circuitry exists on each Memory Expansion board. If an expansion board is addressed, the circuitry asserts BXROE-, where the value of X (either 1, 2, or 3) depends on whether bits 20-19 of the latched map address (LMA20-19) equals 01, 10, or 11.

System Reset (Sheet 5)

System reset is hardware generated by powering on the computer or by pressing the reset switch and is software generated by a write to virtual address $460000 (which asserts HARIKIRI-). The three conditions generate RESET- via the reset monostable multivibrator.

The CPU receives RESET- and puts the vector addresses $000000, $000002, $000004, and $000006 on the bus. Although these are RAM addresses, the General Control register, cleared by RESET-, asserts bit 12 (ROMEN-) the ROM enable bit.

With the asserted ROMEN- input, the main address decoder (PAL 1, sheet 2) asserts ROMSEL-, and the lower 13 bits of the processor address (from the vector addresses) address ROM. The ROMs provide the CPU with a new stack pointer and status word, and the CPU vectors to the legitimate ROM address where the bootstrap routine begins. The bootstrap negates ROMEN-, and the CPU can again address the memory RAMs.

PROCESSOR CONTROL

The processor control centers around the M68010 CPU and includes the CPU state machine, processor registers, and processor address decode.

Fast Cycle Latch and CPU State Machine (PAL 31, Sheet 29)

During fast cycle transfers, the CPU state machine, clocked by PCLK-, defines different states by which it synchronizes the processor control to other elements in the system. The machine also generates DTACK- during both fast or slow cycle CPU transfers. The fast cycle latch, negative edge triggered by 20MHZ-, outputs RAS1- during CPU fast cycles.

Fast Cycle Latch

To commence a fast cycle, the CPU outputs the address on PA23-1, with PA23 low. (Note that PA23 is pulled high to keep it from floating low.) ROMEN- is high. (The CPU asserts ROMEN-, via the GCR latch, under restricted conditions. See the "System Reset" subsection, above, and see Section 4 on the "General Control Register.")

If the system is idle, the bus arbiter has asserted LMEM-, the CPU default bus grant. SRAS1- becomes active and, at 125 ns,

sets the fast cycle latch. The latch asserts RAS1-, which triggers a multitap delay (sheet 19). The delay sends timing signals to other portions of the computer.

CPU State Machine -- DTACK Output

DTACK- is the state machine acknowledge to the CPU that a CPU read/write access has been completed. When the CPU commences a cycle, it begins looking for DTACK- at about 200 ns. If the CPU doesn't receive DTACK-, it executes wait states. The CPU always waits during a slow cycle access because some of the input/output devices have a relatively long response time. The CPU also waits if it requests bus control when the local system is already active.

NOTE

During a test and set (read/modify write) cycle, the CPU waits (after doing the read portion of the cycle) until the system is inactive, then does the write portion.

DTACK- is the only CPU state machine output that is not clocked by PCLK-. When the inputs satisfy a DTACK- equation, the machine immediately sends DTACK- to the CPU. During a fast cycle, SRAS1- and T170+ (from the multitap delay, sheet 19) activate DTACK- at about 180 ns.

During a slow cycle, either IODTK- or EDTK- generate DTACK-. If the CPU is doing an input/output device cycle transfer or interrupt acknowledge cycle, PAL 24 (sheet 17) asserts IODTK- at roughly 770 ns, and the CPU state machine asserts DTACK- at about 800 ns. If an error occurs during the I/O cycle, the error control PAL (PAL 25, sheet 17) asserts EDTK-. The timing of EDTK- depends on the type of error.

CPU State Machine -- Fast Cycle Sequencing

At the PCLK- (150 ns) after the CPU receives SRAS1-, the CPU begins sequencing through the fast cycle states. The machine defines the states by the PRS 2-0, which it feeds back to itself. Figure 5-2 illustrates the CPU state machine sequencing diagram, and Table 5-2 describes the states.

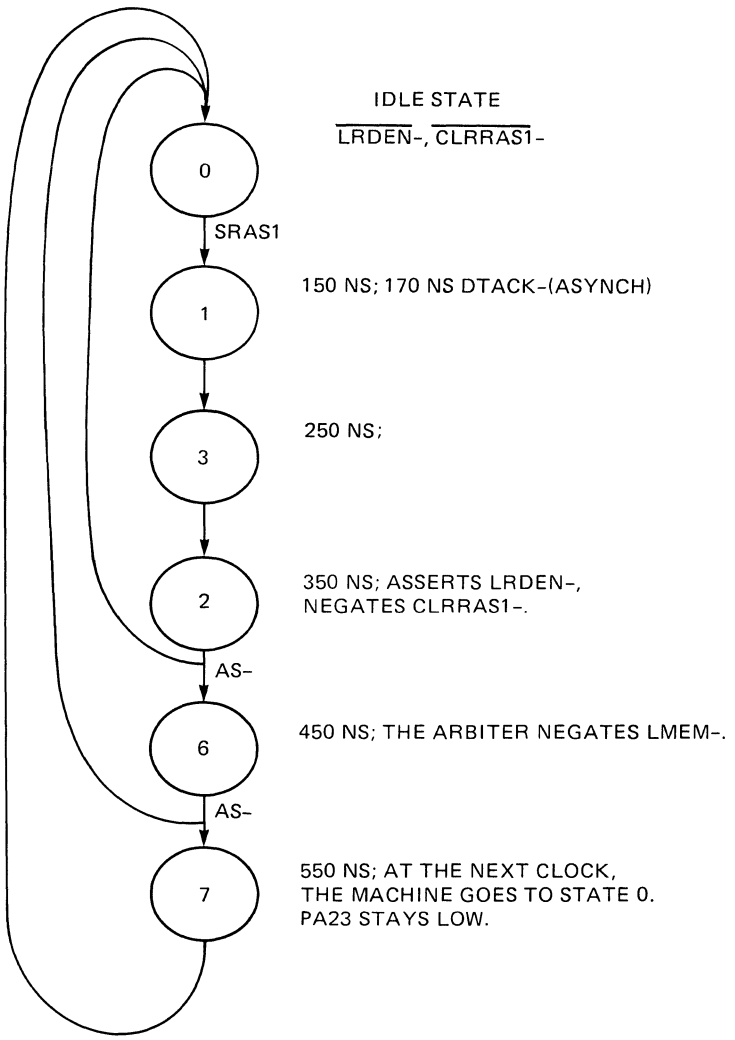


Figure 5-2. CPU State Machine State Sequencing Diagram

Table 5-2. CPU State Machine Fast Cycle States

Time (ns)	State	Function
5Ø	Ø	This state refers to several possible CPU conditions, as follows: 1. The CPU is idle. 2. The CPU is already doing a slow cycle read/write transfer or interrupt acknowledge cycle. 3. The CPU has just begun a fast cycle transfer. (The CPU may also have just finished a fast cycle and is returning from state 2. The MC68Ø1Ø operates with no wait states, so the CPU can begin another cycle immediately.) 4. The CPU has already attempted to start a fast cycle transfer at a previous PCLK-, but the bus arbiter hasn't returned LMEM- yet. 5. The CPU has returned from state 2 and is doing the read portion of a test-and-set cycle. The CPU is waiting for LMEM-. In cases 1 and 2, the machine remains in state Ø (inactive). In cases 3, 4, and 5, if the arbiter asserts LMEM-, it happens at about 6Ø ns and activates SRAS1- (at roughly 11Ø ns). At the next PCLK- (15Ø ns), the machine goes to state 1. If the arbiter doesn't assert LMEM-, the CPU state machine remains in state Ø, and the timing starts over. The machine remains in state Ø until SRAS1-, via LMEM-, and PCLK- advance it to state 1.
15Ø	1	The CPU has bus control for the fast cycle transfer. The state machine outputs DTACK- at 17Ø ns. At the next clock (25Ø ns), the CPU goes to state 3.
25Ø	3	The processor control is doing the read/write cycle. At the next clock (35Ø ns), the machine goes to state 2.

Continued

Table 5-2. Continued

Time (ns)	State	Function
350	2	The machine asserts CLRRAS1-, which clears RAS1-. If the CPU is doing a memory read, the machine also asserts LRDEN-. LRDEN- causes the error control PAL (PAL 25, sheet 17) to check if the memory read generated a parity error. If the read did, LRDEN- causes the PAL to assert NMI-.
		At the next PCLK-, the state machine returns to state 0 (and removes LRDEN- and CLRRAS1-), unless the CPU is doing a test and set cycle. In the CPU is doing a test-and-set cycle, AS- is set, and the machine goes to state 6. CLRRAS1- stays asserted to prevent RAS1- from commencing another cycle.
450	6	The bus arbiter negates LMEM- (at 460 ns). The CPU may remove AS- before 550 ns, in which case the machine goes to state 0 at the next PCLK-; otherwise the machine goes to state 7, keeping CLRRAS1- asserted.
550	7	At the next PCLK-, the machine returns to state 0 (at which time it will negate CLRRAS1-). The CPU keeps PA23 low, to complete the test and set. The test and set will pause in state 0 if the bus is not available (due to memory refresh or DMA), and resume when the bus arbiter returns LMEM-.

Bootstrap ROMs (Sheet 3)

The bootstrap ROMs consist of two 2716/32/64 UV erasable PROMs. The CPU reads the ROMs via a slow cycle when the address decode generates ROMSEL- (PA23-22 = 10) to enable the ROMs for the read. Appendix D describes Boot, the bootstrap program. A portion of the ROM is unused and available to the user for coding.

Procesor Buffers (Sheet 28)

The processor buffers route the CPU address from PA21-1 to LA21-1 and route the bidirectional data bus between PD15-0 and LD15-0. PADENH- and PADENL-, generated by PAL 23, enable the buffers.

Processor Registers

There are five processor registers. Three are multiple bit:

- General Control register (sheet 30)
- General Status register (sheet 17)
- Bus Status register (sheet 17).

The other two processor registers, Clear Status and System Reset, are dynamic signal generated by latches in the CPU register decoder (PAL 2, sheet 2). The latches are data independent write only registers that the CPU writes in order to assert GSRCL-, the clear status register signal and HARIKIRI-, the system reset signal.

The General Status and Bus Status registers are comprised of 373-type, three state, D-latch packages. The registers are in the high impedance state when the CPU is not accessing them, so they don't interfere with the logical data bus. The General Control register is an 8-bit, 273-type, D-latch package.

Processor Address Decode (PALs 1, 2, 3, and 4, Sheet 2)

The processor address decode logic consists four PALs:

- Main address decoder (PAL 1)
- CPU register decoder (PAL 2)
- I/O address decoder #1 (PAL 3)
- I/O address decoder #2 (PAL 4).

When the CPU commences a fast or slow cycle transfer, the main address decoder receives the device select address (PA23-16) and part of the function code (SUPV and PROGRAM) from the CPU, receives AAS-, LIO-, and NPCYCLE- from the bus arbiter, and recevies FCHI- from the memory error monitor (PAL 30). The PAL decodes these signals to generate select signals to different system elements.

The CPU register decoder is addressed by PA19-16, LWT-, UDS-, LDS-, and AAS-. This PAL outputs the control signals for the CPU accesses to the processor registers.

The I/O address decoder #1 decodes PA19-16 and several other signals to assert the select signals to the interrupt controller and slow communications port. The PAL also generates the interrupt acknowledge counter load/count bit.

The I/O address decoder #2 generates selects to the timer, disk bus interface unit, disk DMA controller, fast communications bus interface unit, fast communications DMA controller, line printer, and external port.

Status Indicator Lights (Sheet 30)

There are five status indicator lights; four are associated with the General Control register and one is associated with power on. The processor control turns on and off the four indicators associated with the General Control register by setting and resetting the corresponding register bits. The power on indicator lights up when the computer is powered on.

MEMORY CONTROL

The memory control has the

- Map logic (sheet 27)
- Memory access logic (sheet 19 and 27)
- RAM memory (sheet 22-25)
- Parity (sheet 21)
- Refresh logic (sheet 17 and 20)
- Memory error control (sheets 17 and 27).

Map Logic (Sheets 17 and 27)

The Map logic includes the

- Map control PAL (PAL 29)
- Map RAM.

A discussion of the map logic operation during a CPU/map register access and a CPU/memory or DMA transfer follows.

Map Logic -- CPU Accesses

When the CPU read/writes the map RAMs, the CPU address decode asserts MMUSEL-. MMUSEL- enables the read/write portion of the map control PAL and routes the lower ten processor address bits to logical address 21-12 (sheet 27). MALDEN+, always negated, enables the map RAMs, and the RAMs output the contents of the addressed location in the page table.

If MCWT- (CPU map write) is asserted, PAL 29 asserts MALDEN-, which, with the write signal, routes the processor data, via a buffer, to the map RAMs. When the memory control PAL (PAL 27, sheet 19) asserts CLRRAS- (about 380 ns), the RAMs latch the data. If the CPU does a map read, the map RAMs outputs the data to the logical data bus, and the CPU strobes in the data.

Map Logic -- Memory Accesses

During CPU/memory and DMA accesses, the accessing device puts the map address on bits 21-12 of the logical address bus and asserts or negates logical write (LWT-). The memory control PAL clocks the addressed page address and status bits from the addressed map into the map address register via PADLTCH-. The register sends the page address via latched map address (LMA2Ø-12) to the Memory address multiplexer and sends the status bits to the map control PAL (PAL 29).

At about 26Ø ns in the access, BASEMEM-, from the memory control, asserts UPT- (by PAL 26), causing a page table update. Latched write (MCWT-) tells the map control PAL how to adjust WE+, PS1, and PS2, and the PAL returns the new status to the map RAMs. The update signal strobes the status into the RAMs by controling the upper RAM write enable, PTWTU-.

During a memory refresh cycle, REFGNT- is input to the memory control PAL (PAL 27, sheet 19) and blocks ENCAS-. ENCAS- (high) stops UPT-; so the refresh does not cause a spurious update.

Memory Access Control (Sheets 19 and 27)

The memory access logic has the

- Multitap delay (sheet 19)
- Memory control PAL (PAL 27, sheet 19).

At roughly 13Ø ns in a memory access, the logic receives either RAS1- from the processor control or RAS2- (for DMA or refresh) and asserts RAS+.

Multitap Delay

RAS+ triggers the multitap delay, which sends out delayed waveforms identical to RAS+. The delay outputs are labeled T17Ø+, T2ØØ+, T23Ø+, T26Ø+...T44Ø+, labels that correspond to the time in the access when they are asserted (for example, T17Ø+ is asserted at about 17Ø ns in the access). RAS- goes to the invalid memory control PAL.

Memory Control PAL (PAL 27, Sheet 19)

In addition to the delay signal inputs, the memory control PAL receives five other inputs (RAS, PROCREG-, REFGNT-, MCWT-, and MMUERR-), which define the the type of memory access. This PAL

outputs seven control signals (PADLTCH-, CLRRAS-, CLRREF-, XWE-, ROE-, ENCAS-, and REFRAS-). XWE-, activated by latched write (MCWT-), enables the memory write, unless the memory error control asserts the invalid access signal (MMUERR-). Section 6, "Fast Cycle Transfers," describes PAL 27 operation in greater detail.

Memory and Memory Addressing

This description mentions the following elements:

- RAM memory (sheet 23-26)
- Memory row address (MEMROWADR) decoder (sheet 19)
- Memory address multiplexer (sheet 20)
- Memory column address decoder (sheet 19).

At the beginning of a memory access, the map logic outputs memory board select LMA20-19. This enables either the Main Processor board or one of the Memory Expansion boards. Table 5-3 describes the LMA20-19 decode.

Table 5-3. Board Selection for Memory Access

LMA20	LMA19	Board
0	0	Base (#0) memory
0	1	#1 ME board memory
1	0	#2 ME board memory
1	1	#3 ME board memory

The memory board select equal to 00 selects the Main Processor board and asserts BASEMEM- (sheet 19). Next, the byte offset, LMA10-3, selects one of the 256 memory pages in each of the four memory banks (rows) on the board. The delay tap T170+ is yet inactive, so the memory address multiplexer (sheet 20) routes the page address to all four RAM bank address buses.

The row select portion (LLA2-1) of the latched logical address selects one of the four RAM rows, called MEMROWs 0-3 (sheets 22, 23, 24, and 25, respectively). Each row contains 16 data RAMs and 2 parity RAMs. At roughly 130 ns, RAS+ asserts a RAM row address strobe (RAMRAS0-, RAMRAS1-, RAMRAS2-, or RAMRAS3-), by the MEMROW decode circuit (sheet 19).

The strobe clocks the page address into the selected RAM bank. **NOTE:** During DMA, successive addressing enables rows 0, then 1, then 2, then 3, then 0, and so forth, as the row select goes from 00 to 01 to 10 to 11 to 00 etc.

The page address, LMA18-11, selects one of 256 words (and parity) in the page. At about 170 ns, after RAS+ has triggered the delay, T170+ routes the page address to the four bank address buses, via the memory address multiplexer. Shortly later, the Memory control PAL asserts the RAM column address strobe, ENCAS-.

The column address decoder, selected by BASEMEM- and enabled by ENCAS-, decodes the word select (LLA2-1) and data strobes (UDS- and LDS-), then asserts an upper column and lower column address strobe (one each of UCAS- 3-0 and LCAS- 3-0). If only one byte is addressed, the decoder asserts only the valid upper or lower strobe.

The strobes clock the page address into the RAM column address latches. Depending on the write enable level (MWE- from the memory control PAL), the RAMs do a read or write. Write data must be valid until about 250 ns. Read data for the RAMs becomes valid at about 300 ns.

Parity (Sheet 21)

The parity circuitry contains

- Two parity generator/checkers
- Parity sense buffer
- Parity error gates.

Software normally sets the parity sense signal (PS+) high to select odd parity mode.

Parity Write

During a write cycle, the parity circuit generates odd parity on the parity RAM inputs (RDI17-16) for the upper and lower data bytes (RD15-0) as follows: memory write enable (MWE-) low and parity sense high drive the parity decode (RDO17-16) high; the upper data byte and the high parity drive the upper byte parity chip; the lower data byte and the other high parity drive the lower byte parity chip. (RDO17 with RD15-8 and RDO16 with RD7-0).

If a byte has an even number of ones, the high parity provides odd parity, and the parity chip asserts its odd parity output, forcing the parity RAM input high. If a byte has an odd number of ones, the high parity provides even parity and the chip negates its odd parity output, forcing the parity RAM input low.

In this way, the parity circuit generates odd parity because the byte + parity it sends to the RAMs always has an odd number of ones.

Parity Read

During a read transfer, the parity circuit checks the read data for odd parity. Memory write enable is high, which disables the parity sense buffer (sheet 21).

The read data and parity drive the parity chips. Each read byte + parity should have odd parity. If either chip detects even parity, it asserts its even parity output, SEL+.

The miscellaneous logic PAL (PAL 36, sheet 21) will have already asserted the parity data strobes (PLUDS+, PLLDS+, or both), and these strobes enable the corresponding parity output to parity error (PE+). Parity error is input to the CPU interrupt logic.

Refresh Control (Sheets 17 and 2Ø)

The refresh control contains the

- Refresh address generator (sheet 2Ø)
- Refresh request flip/flop (sheet 17).

Approximately every 13 microseconds, REFCLK, a 76.8-kHz clock from the system clocks, clocks the refresh request flip/flop, which resets the flip/flop and asserts bus request, REFREQ-, from the refresh control.

When it can, the bus arbiter returns the refresh grant, REFGNT-. (The maximum time it should take the arbiter is roughly eight microseconds). REFGNT- enables the refresh address generator, and the generator supplies the 8-bit page address to the memory address multiplexer. REFGNT- also asserts SETRAS2-, via PAL 36 (sheet 4). The next rising edge of 2ØMHZ asserts RAS2- and RAS2. RAS2 causes SETRAS2- to remain asserted until CLRRAS- clears RAS2.

RAS2- starts a memory read cycle. T17Ø+ selects the refresh address, LA 1Ø-3, and routes it to the RAM banks. The memory control PAL outputs REFRAS-, which asserts all four RAMRAS- signals. All four banks of RAMs are enabled, and each bank refreshes one row of bits. (The RAMs only need a row address to refresh) The same occurs on Memory Expansion boards.

REFGNT- (negated) disables and clocks the refresh address generator, causing it to increment the refresh address for the next cycle. All memory (including expansion) is refreshed approximately every 3.3 or 1.65 milliseconds. That is, the time between refreshes for any memory location is 3.3 milliseconds (for RAMs that refresh 256 rows) or 1.65 milliseconds (for RAMs that refresh 128 rows).

Memory Error Control (Sheet 17)

The memory error control consists of three PALs:

- Memory error monitor PAL (PAL 3Ø, sheet 27)
- Error control PAL (PAL 25, sheet 17)
- Memory error latch PAL (PAL 26, sheet 17).

Memory Error Monitor (PAL 3Ø, Sheet 27)

This PAL monitors memory accesses to make sure they are valid. There are five types of invalid memory accesses, as follows:

1. User attempt to write a memory location that, according to the page table, is not write enabled. (MCWT, LMEM-, and RAS- are asserted, PA22 is low, SUPV is negated, and LWE- is asserted.)

2. User attempt to access a memory location which is part of the Kernel. (LMEM-, and RAS- are asserted, PA22 is low, and SUPV is negated. PA21-19 = ØØØ).

3. Nonprocessor attempt to access a memory location on a page not present. (RAS- is asserted, and LMEM- and XGNT- are negated. LPS1-Ø are ØØ.)

4. Processor attempt to access a memory location on a page not present. (RAS-, LMEM- are asserted, and PA22 is low. LPS1-Ø = ØØ.)

5. Processor access to a nonmemory location. (LMEM- and RAS- are high. PA22 is high.)

Conditions 1 through 4 inhibit the access by generating CASINH-, which prevents the memory access control from generating the CAS- signals. Conditions 1 through 4 also generate the memory management unit error, MMUERR-. Condition 5 is actually not an error; rather it prevents writing to the RAM when the processor registers are addressed. It also generates CASINH-. Conditions 3 and 4 generate the page fault error, PGF-.

MMUERR- disables the page table update and generates either a nonmaskable interrupt (for DMA errors) or a bus error (for processor errors), via the error control PAL. If other conditions comply, MMUERR- and PGF- generate a nonmaskable interrupt.

Error Control PAL (PAL 25, Sheet 17)

This PAL generates three signals: nonmaskable interrut (NMI-), bus error (BERR-), and error acknowledge (EDTK-). The CPU enables the PAL by setting error enable (EE) in the General Control register. Referring to Appendix D, the following conditions generate these outputs. A slash means the signal is negated, an asterisk is equivalent to an AND condition, and a plus sign is equivalent to an OR condition. See the previous subsection for a description of MMUERR.

- Nonmaskable interrupt:

  Any access to nonexistent memory location.
  (EE * /REFGNT * MNP * /PGF * /MMUERR)

  Parity error during a CPU/memory read.
  (EE * /REFGNT * /MMUERR * PIE * PE * /LWT * T2ØØ * /UIOER * /NPCYCLE * /PREG)

  Parity error during a DMA read.
  (EE * /REFGNT * /MMUERR * PIE * PE * /LWT * T2ØØ * NPCYCLE)

  Page fault during a DMA read/write. (EE * /REFGNT * PGF * NPCYCLE)

  Memory management unit error a DMA read/write. (EE * /REFGNT * MMUERR * NPCYCLE)

- Bus error:

  Page fault during a CPU access. (EE * /REFGNT * PGF * /NPCYCLE * T2ØØ)

  User attempt to access an input/output device register or the Kernel. (EE * /REFGNT * UIOERR * /NPCYCLE * T2ØØ)

  Memory management unit error during a CPU access.
  (EE * /REFGNT * MMUERR * /NYCYCLE * T2ØØ)

- Error data transfer acknowledge:

  Parity error during a confirmed valid CPU/memory access.
  (EE * /REFGNT * LRDEN * /NPCYCLE * /MMUERR * PIE * PE * T2ØØ * /UIOER * /PREG)

  CPU attempt to access a nonexistent memory location.
  (EE * /REFGNT * MNP * /NPCYCLE)

  Page fault generated during CPU access.
  (EE * /REFGNT * PGF * /NPCYCLE * T2ØØ)

Memory Error Latch (PAL 26, Sheet 17)

This memory error latch causes the Bus Status register to latch on a nonmaskable interrupt (NMI- is asserted), a bus error (BERR- is asserted), or an access a memory location not present (MNP- is asserted). The Bus Status register latch signals have the prefix GSR. The PAL also generates the page table update signal, UPT-.

DISK CONTROL

The Disk control includes three parts:

- Disk DMA controller
- Disk bus interface unit
- Disk controllers.

Disk DMA Controller (Sheet 9)

The disk DMA controller consists of three registers, the disk DMA controller register, Disk Word Address register, and Disk Word Count register. Section 4, "Disk DMA Controller Registers," adequately describes the registers. Section 6, "Disk DMA Operations," describes how the registers operate during disk DMA.

Disk Bus Interface Unit (Sheets 7 and 8)

Section 6, "Disk DMA Operations," describes the disk bus interface unit in considerable detail. The unit includes the disk word state machine, which consists of two PALs, and the disk byte state machine, which consists of three PALs.

During a DMA operation or a CPU/disk controller register access, the state machines sequence through different states. Each state is defined by certain combinations of inputs. In most states, the machines also assert one or more outputs. For both machines, the idle state is state zero (indicated by WØ for the word machine and BØ for the byte machine).

Figures 5-3 and 5-4 are the disk word and disk byte state machine sequencing diagrams. The bubbles show the state the machine is in. The pertinent signals causing a machine to change states are indicated on the lines connecting the bubbles. If a slash is included with the signals, it means the machine asserts (or negates) an output in the state it is about to enter, and the output follows the slash. Any signal with a line over it is negated. (That is, the machine negates the output, or the input must be negated.)

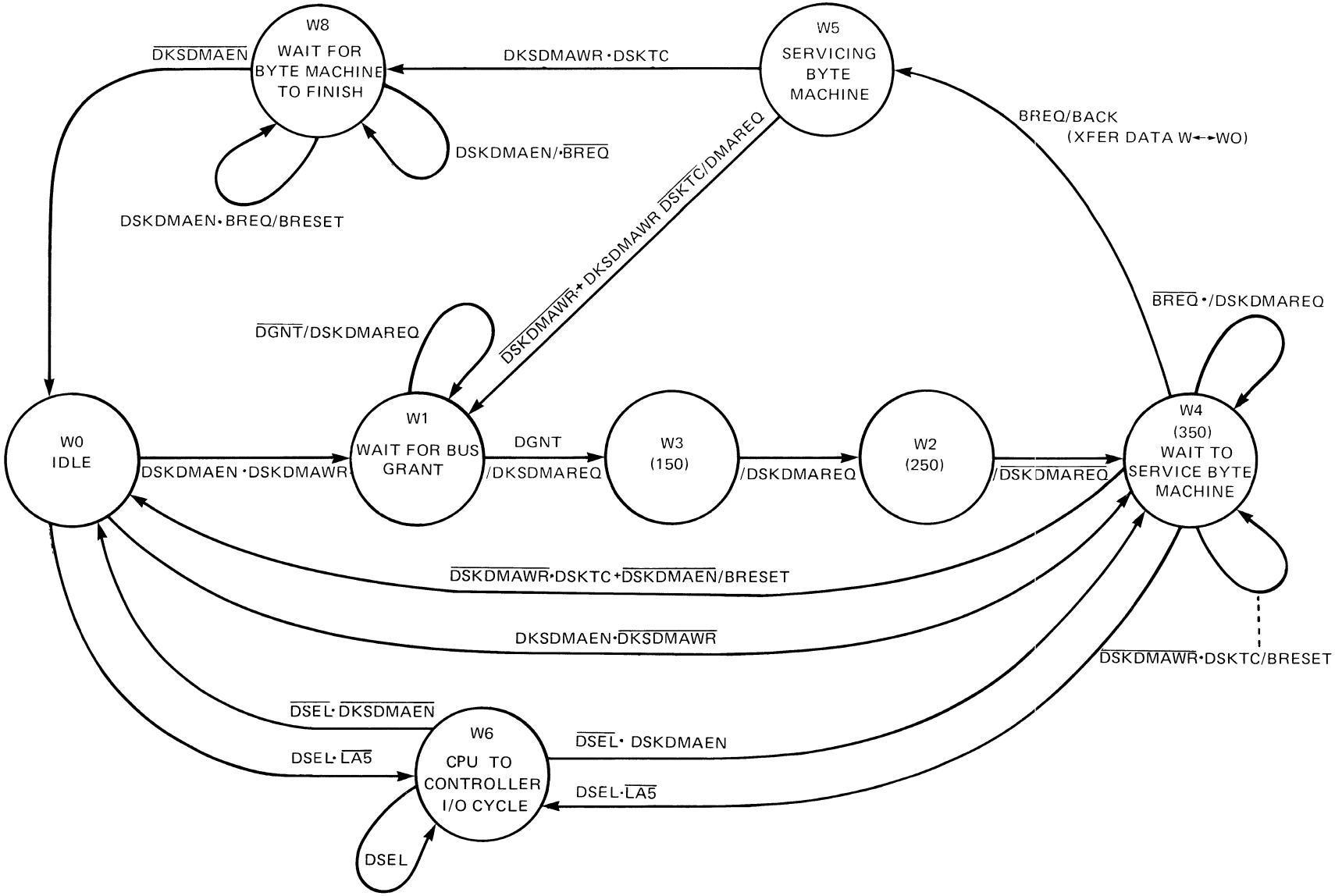


Figure 5-3. Disk Word Machine State Sequencing Diagram

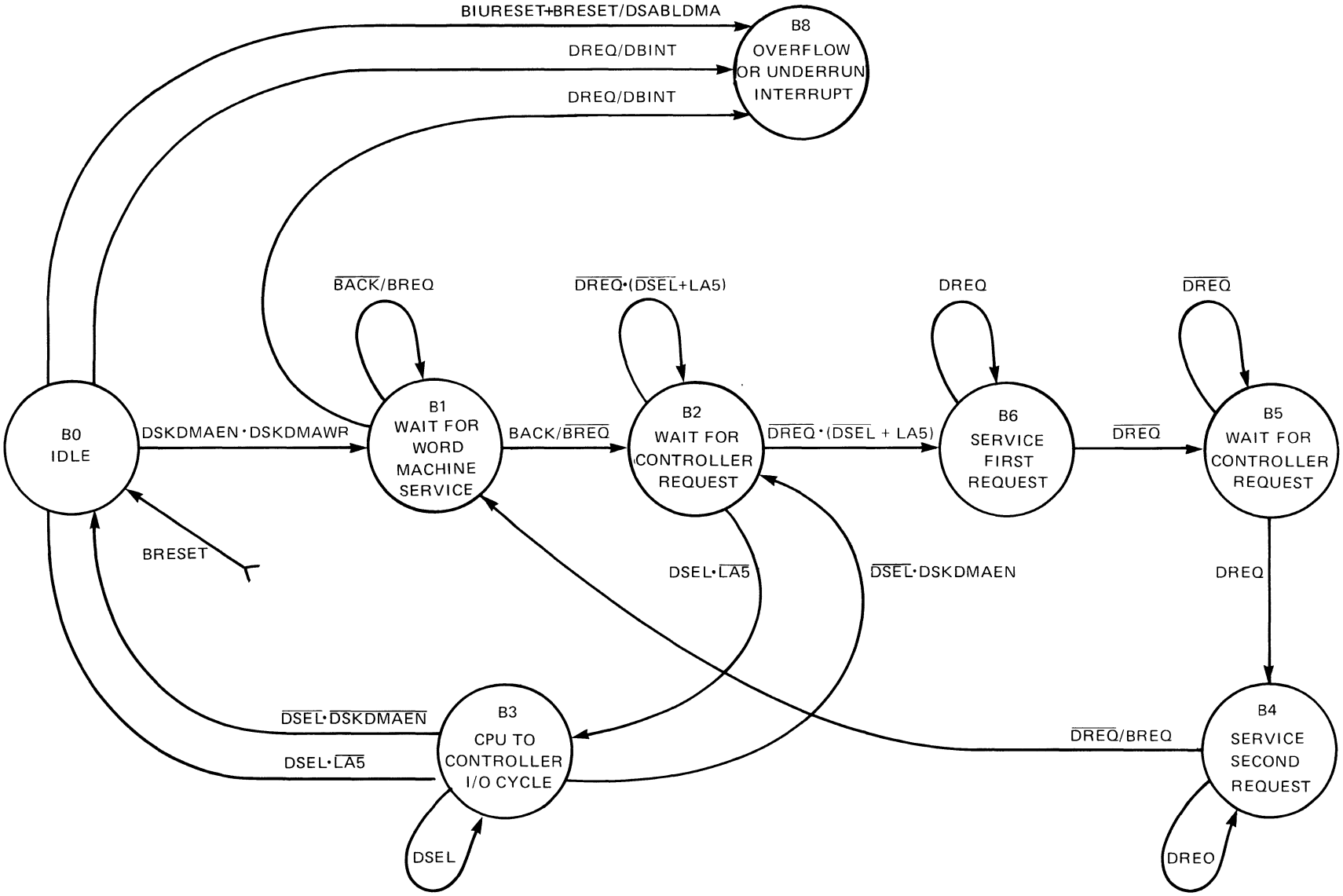


Figure 5-4. Disk Byte Machine State Sequencing Diagram

Hard Disk Controller (Sheets 15 and 16)

The hard disk controller operates in two modes, CPU access (register) and DMA. In both modes, the disk bus interface unit (DBIU) interacts with the controller to peform the register or DMA transfer.

Three controller signals, HDWE-, HDRE-, and HDBCS-, direct the timing and control for either mode of transfer. DDWE- and DDRE- are the DBIU counterparts of HDWE- and HDRE-. The other control signal is DSKDRQ-, which the interrupt PAL (PAL 11, sheet 8) generates.

In CPU access mode, the controller is a bus slave device, and the DBIU, directed by the CPU, controls HDWE- and HDRE- by asserting HDCS-. HDCS- enables the HD RE/WE buffers (sheet 15), which route DDWE- or DDRE- from the byte machine decoder #2 (PAL 7) to the hard disk controller's WE- and RE- inputs. HDCS- tells the controller to prepare for a register access, and HDWE- or HDRE- control the access.

In DMA mode, the controller is a bus master device. At the beginning of a DMA operation, the controller asserts HDBCS-. When it is ready to transfer data, the controller asserts HDWE- or HDRE-.

The controller doesn't wait for any response and assumes the slave device (the DBIU) will latch or output the data. The disk interrupt PAL receives the read/write signal and generates DSKDREQ-, which tells the DBIU to do the transfer. The controller terminates the transfer by removing HDWE- or HDRE-, which negates DSKDREQ-.

NOTE

The disk request latch (sheet 7) generates SYNCDDREQ- from DSKDREQ-. SYNCDDREQ- synchronizes the disk controller request to the DBIU. The disk control subsection of "Input/Output Operations," in Section 6 contains timing diagrams which illustrate the relationship between DSKDRQ- and SYNCDDREQ-.

Floppy Disk Controller (Sheet 14)

Like the hard disk controller, the floppy disk controller operates in two modes, CPU access and DMA, and interacts with the

disk bus interface unit (DBIU) to perform transfers. Unlike the hard disk controller, the floppy disk controller is always a bus slave. FDCS-, DDWE- and DDRE-, FDDRQ- (and DSKDRQ-), and LATCH- are the control signals for the transfers.

In either mode, the disk byte machine state asserts FDCS- at the beginning of the transfer. In DMA mode, the byte machine waits for the controller to assert FDDRQ-, which causes the disk interrupt PAL (PAL 11, sheet 21) to assert DSKDRQ-. DSKDRQ- tells the DBIU the controller is ready to transfer, and the DBIU asserts either DDWE- or DDRE-, which tells the controller to do the transfer.

NOTE

The disk request latch (sheet 7) generates SYNCDDREQ- from DSKDREQ-. SYNCDDREQ- synchronizes the disk controller request to the DBIU. The disk control subsection of "Input/Output Operations," in Section 6 contains timing diagrams which illustrate the relationship between DSKDRQ- and SYNCDDREQ-.

FAST COMMUNICATIONS RS-422 PORT

Fast commmunications includes the fast communications bus interface unit, DMA controller, and 2652 port control.

Fast Communications Bus Interface Unit (Sheet 11)

The fast communications bus interface unit includes the word/byte state machine and the word/byte transceivers. The unit is very similar to the disk bus interface unit, except the word and byte machines are combined, and the word and byte transceivers are combined. The fast communications port requires only a one word buffer because it transfers at a lower rate than the disk control.

The word/byte state machine controls two things: (1) CPU/2652 register accesses and (2) fast communications DMA. Figure 5-5 illustrates the word/byte state machine sequencing during the accesses. Refer to the "Disk Bus Interface Unit" subsection to the exlanation of the disk word state machine and disk byte state machine sequencing diagrams for an equivalent explanation of how to interpret Figure 5-5.

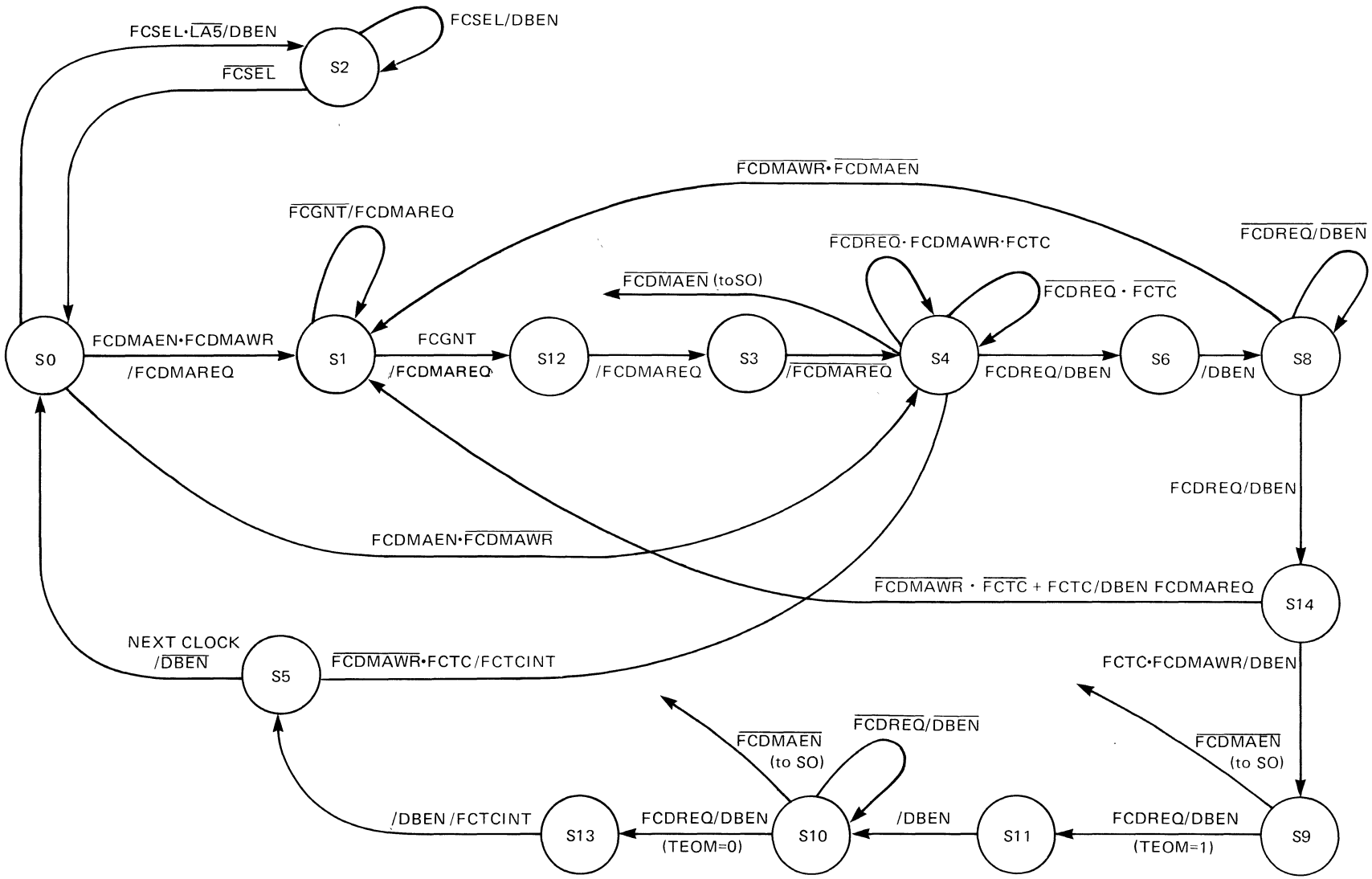


Figure 5-5. Fast Communications Word/Byte Machine Diagram

CPU/2652 Register Accesses

Refer to Figure 5-5, the word/byte state machine sequencing diagram, for an illustration of the machine's inputs and outputs during the CPU/register access. When the CPU read/writes a 2652 register, the processor address decode asserts FCSEL-.

FCSEL- causes the word/byte state machine to disable the upper byte buffer, as only the lower byte from the port controller goes to the word/byte buffer. The CPU obtains the upper byte directly from the controller via the register upper byte buffer (sh. 13).

The write signal, FCWRT-, is input to the 2652 to indicate the direction of transfer. The lower three bits of the logical address, LA3-1, are input to the 2652 register address lines (FC3-1) via a 241 buffer. The 2652 chip enable, FCCE, is normally already asserted.

FCSEL- and LA5 (low) drive the word/byte state machine to state 2, where it stays until the I/O address decode #1 negates FCSEL- The machine bus asserts the data bus enable, FCDBEN-, which is inverted and routed to the DBEN input in the 2652. DBEN tells the 2652 to prepare to read or write the data. The word/byte machine also asserts the word/byte buffer control signals, CBØOE-, CBØASEL-, and CBØBSEL-.

During a register read, the word/byte machine asserts CBØDRVA-, which selects the A-bus (2652 to CPU). At 97Ø ns, the CPU negates the data strobes and latches the upper byte from the register upper byte buffer and the lower byte from the byte/word buffer, both via the logical data and processor data buses. The I/O address decoder #2 removes FCSEL-. The word/byte state machine returns to state Ø and negates DBEN-.

During a register write, the word/byte machine asserts CBØDRVA-, which selects the B-bus (CPU to 2652). At 86Ø ns, the at time sequencer (sheet 4) asserts AT5, which causes the I/O address decoder #2 to remove FCSEL-. The machine returns unit to state Ø and negates DBEN-, which causes the 2652 to latch the lower byte from the word/byte buffer and the upper byte from the register upper byte buffer.

DMA Accesses

The word/byte state machine controls the local portion of the fast communications DMA operation. For a detailed understanding of the machine's sequencing, decode the PAL listings in Appendix A according to the directions included in the "PAL Overview" and as done in the "Disk Control Input/Output" subsection. Figure 5-5 shows the pertinent signals involved in the sequencing.

SLOW COMMUNICATIONS RS-232-C PORT (Sheets 31 and 32)

The slow communications circuitry includes:

- RS-232-C port control
- Slow communications register/clock control

RS-232-C Port Control (Sheet 31)

The 8274 serial controller (sheet 31) performs all communications control, including serial data-in/data-out, bit/byte control, status update, and CPU interrupt. The only reference to the controller internal hardware in this manual are the 8274 internal register addresses provided in Section 4.

Refer to a manufacturer's 8274 specification for information on this chip. **Note:** The 72Ø1 controller performs the identical functions as the 8274 and may be substituted for the 8274.

Slow Communications Register/Clock Control (Sheets 31 and 32)

This circuitry routes data between the CPU and the slow communications control, contains all the slow communications registers other than the controller registers, and does the register address decode and read/write control for slow communications register accesses. The circuitry contains the following elements:

- Slow communications data buffer (sheet 31)
- Slow communications/interrupt controller PAL (PAL 32, sheet 31)
- 8253 timer/baud rate generator (sheet 32)
- Baud rate generator control PAL (PAL 33, sheet 32)
- Slow communications clock/data register (PAL 34, sheet 32).

The slow communications data buffer routes data between the lower byte of the logical data bus (LD7-Ø) and the slow communications data bus, SCD7-Ø. The slow communications data bus also goes to the 8259 interrupt controller.

The CPU loads the slow communications and interrupt controller registers by a normal input/output instruction. At roughly 3ØØ ns into the cycle, the CPU address decode logic asserts SCSEL-. If the CPU is accessing the baud rate generator, the logic

asserts SCTSEL- instead. The slow communications/interrupt controller PAL receives SCSEL-, SCTSEL-, ICASEL-, LA4-1, and LWT-.

If the slow communications/interrupt controller PAL determines that a slow communications or interrupt controller register is addressed, at about 35Ø ns, it both enables the slow communications data buffer, via MOSBOE-, and selects the data direction, via MOSDIR-. If an 8274, 8253, or 8259 register is accessed, the PAL also activates either MOSRD- or MOSWR-. Finally, the PAL activates one of the following signals, for the register or device select:

- 8274 serial port controller: the PAL asserts 8274CS-. LA2-1 determine which register in the 8274 chip is acccessed.

- 8259 interrupt controller chip: the PAL asserts ICACS-. LA1 determines which register in the 8259 is accessed.

- Clock/Data register: the PAL asserts SCEXREGWR-. The PAL receives SCD3-Ø. SCD3-2 carry port A and port B secondary transmit data, respectively. SCD1-Ø select clock external or timer clocks for ports A and B, respectively.

If the 8253 is accessed, the baud rate generator control PAL is enabled by SCTSEL-. The PAL decodes LA4-3 and LWT- to determine if the timer/BAUD rate generator is being accessed. LA2-1 goes directly to the chip to select the interrupt controller register to be read or written. (This allows sufficient address set up time before the PAL asserts the read or write signal.) If LA4-3 = ØØ, the chip is being addressed and at about 35Ø ns, the PAL asserts SCTCS- and, depending on LWT-, either SCTRD- or SCTWR.

Once the slow communications or interrupt controller register selection is done, the timing is the same as for any CPU input/output cycle. The select line asserts at approximately 35Ø ns. During a read, the 8274 serial controller makes the data available at about 55Ø ns; the 8259 interrupt controller and 8253 timer/BAUD rate generator make it available at about 6ØØ ns. At about 86Ø ns, AT5- removes the SCSEL- or SCTSEL- select signal and, at about 9ØØ ns, the read/write line. In the meantime the CPU clocks in the read data.

The timing is similar during a write. The SC signal negates at about 86Ø and the read/write signal at about 9ØØ. The CPU holds the data valid until about 95Ø ns, which gives all the chips adequate time to latch the data. The recovery times are 3ØØ ns (8274 serial controller and 8259 interrupt controller) and 1ØØØ ns (8253 timer/BAUD rate generator.) **NOTE:** The CPU cannot perform successive input/output cycles to the same 8253 chip.

LINE PRINTER INTERFACE (Sheet 26)

The line printer interface is mainly a data and status buffer. The interface consists of:

- LS374 line printer data out buffer
- F244 line printer status in buffer
- Line printer control PAL (PAL 28)
- LS161 line printer strobe generator
- Line printer interrupt flip/flop.

The data and status buffers hold outgoing data bytes or incoming status to or from the printer. To send the printer a byte, the CPU does a write to the data/status address in the line printer control PAL and places the byte on the lower eight bits of the data bus. The CPU writing the data/status address causes the PAL to pulse LPWRITE- active during the write. LPWRITE- clocks the data into the data out buffer.

The LPWRITE- pulse also triggers the strobe generator. The normal status of the generator (after a system reset or after finishing a previous write) has LPSTROBE- negated and the feedback to pin 1Ø low. (RESET- is input to the 161's load pin and causes the 161 to load the hardwired inputs; the feedback input, wired high, is inverted before being fed back to pin 1Ø.)

The LPWRITE- pulse clears the generator, which drives the feedback high. With pins 7 and 1Ø high, the generator is enabled and on the next clock (the system clock COMMCP) begins counting. At the fourth clock, the generator asserts pin 13, which asserts LPSTROBE-. At the next clock, the generator negates LPSTROBE- and once again drives the feedback low. This suspends counting until another LPWRITE- pulse arrives.

The printer receives the LPSTROBE- pulse, takes the data, and returns LPACK-. The line printer interrupt flip/flop, clocked by LPACK-, outputs LPINTR. LPINTR generates an autovector interrupt to alert the CPU that the printer is ready for another byte. When the interrupt control returns INTALP-, the line printer interrupt acknowledge, the line printer control PAL resets the flip/flop. **NOTE:** The CPU can disable LPINTR by writing to the disable interrupt address in the line printer control PAL.

To read the printer status, the CPU reads the data/status address in the line printer control PAL. The PAL asserts LPREAD-, which enables the status inputs through the status-in buffer onto the logical data bus. The CPU strobes in the data.

When the computer is executing printout routine, the CPU provides the memory address for the outgoing bytes, maintains the transfer word count, and routes the data between memory and the printer, via a fast cycle memory read and the just described slow cycle printer write.

INTERRUPT CONTROL

The interrupt control contains the

- Interrupt controller (sheet 18)
- Interrupt level generator (PAL 23, sheet 17)
- Interrupt acknowledge PAL (PAL 24, sheet 17).

A description of these elements follows.

Interrupt Controller (Sheet 18)

Table 5-4 lists the prioritized interrupt requests input to the 8259 interrupt controller:

Table 5-4 Interrupt Controller Interrupt Request Inputs

Priority	Source	Condition Generating Interrupt
Ø (high)	Fast comm.	FCTXU status line asserted.
1	Fast comm.	FCRXSA status line asserted.
2	Fast comm.	FCCARRIER status line asserted.
3	Fast comm.	FCTC status line asserted.
4	Fast timer	Interrupt request from timeout.
5	Disk control	Disk overflow and underrun.
6	Disk control	Hard disk end of transfer.
7 (low)	Disk control	Floppy disk end of transfer.

Each interrupt controller input has a register associated with it that the CPU loads with an interrupt vector address for the condition generating the input. Refer to the section on the slow communications register/clock control to see how the CPU loads the interrupt controller. Refer to the manufacturer's 8253 engineering specification for more information on the interrupt controller.

Interrupt Level Generator (PAL 23, sheet 17)

The interrupt level generator receives interrupts from several places, determines the highest priority device requesting an interrupt, and sends the interrupt control field for that device to the CPU. The interrupt control field is the binary equivalent of the interrupt priority level. Table 5-5 lists, in order of decreasing priority, the interrupt requests to the interrupt level generator and the device generating the interrupt.

Table 5-5 CPU Interrupt Priority Levels

Level	Signal	Source/Function
7	NMI	Nonmaskable autovector interrupts, generated by the error control PAL (PAL 25). All NMIs are generated during a memory access and have the same priority. EE (GCR bit 15) being reset blocks NMIs. The following errors generate NMI: PE: read memory parity error. (PIE, GCR bit 14, must also be set to allow the interrupt.) MNP: memory not present error. Expansion memory transfer where the addressed memory board doesn't return CASACK-. PGF: page fault error. DMA or CPU/memory access with a page not available (as indicated by PS1-Ø). MMUERR: memory management unit error. See memory error PAL, "Memory Control," Section 5.
6	TMR1FF	Timer #1 flip/flop. Autovector interrupt from the system timer. Results from a timer tick.
5	IRQ8274	Autovector interrupt from either half of the 8274 RS-232-C port requesting CPU intervention for a data transfer.
4	IRQ8259A	Normal vector interrupt from the 8259 on-board interrupt controller.
3	IRQ8259B	Normal vector interrupt from the optional expansion port.
2	EIRQ	Autovector interrupt from the expansion port.
1	LPINTR	Autovector interrupt from the line printer requesting CPU intervention for a memory access.

Interrupt Acknowledge PAL (PAL 24, sheet 17)

The interrupt acknowledge PAL, enabled by LIO- and INTAADDR (from the device select PAL) and AAS- (from the CPU) decodes bits PA3-1 to assert the interrupt acknowledge (INTA) to the interrupting device. PA3-1 equals the interrupt priority level the CPU has selected.

If the CPU has selected the programmable interrupt controller, 8259A, the expansion normal vector device (8259B), or the 8274, the interrupt ackowledge PAL must assert the acknowledge pulse twice; the first time to prepare the device and the second time to cause the device to output the vector byte.

Twice Acknowledge Counter

The twice acknowledge counter is normally cleared by the negated AAS+. When AAS+ goes high (roughly 21Ø ns), the counter is enabled. Every 5Ø ns, the counter is clocked by 2ØMHZ-. The counter is configured as a divide by five counter, and QA toggles every five clocks. Thus, at 45Ø ns, INTACLK+ goes high, and at 7ØØ ns, INTACLK+ goes low. AAS+ goes low before 95Ø ns, so INTACLK+ stays low.

Referring to the listing for PAL 24, the effect of INTACLK going high is to pulse the device acknowledge signal low for 25Ø ns (in the middle of the interrupt acknowledge cycle). This, in effect, creates a double acknowledge pulse, which the devices need to output the interrupt vector byte.

At roughly 75Ø ns in an autovector interrupt cycle, AT4+ causes the interrupt acknowledge PAL to send the interrupt acknowledge, IODTK-, to the CPU state machine. The machine converts this to DTACK- for the CPU. The PAL also asserts IODTK- for slow cycle accesses.

SECTION 6: OPERATION

This section describes the MiniFrame Computer System operation.

OVERVIEW OF SYSTEM OPERATION

One model of the MiniFrame Computer system, as Figure 6-1 shows, includes the system control and memory control as central elements, the processor control and input/output devices as local elements, and peripherals as external elements.

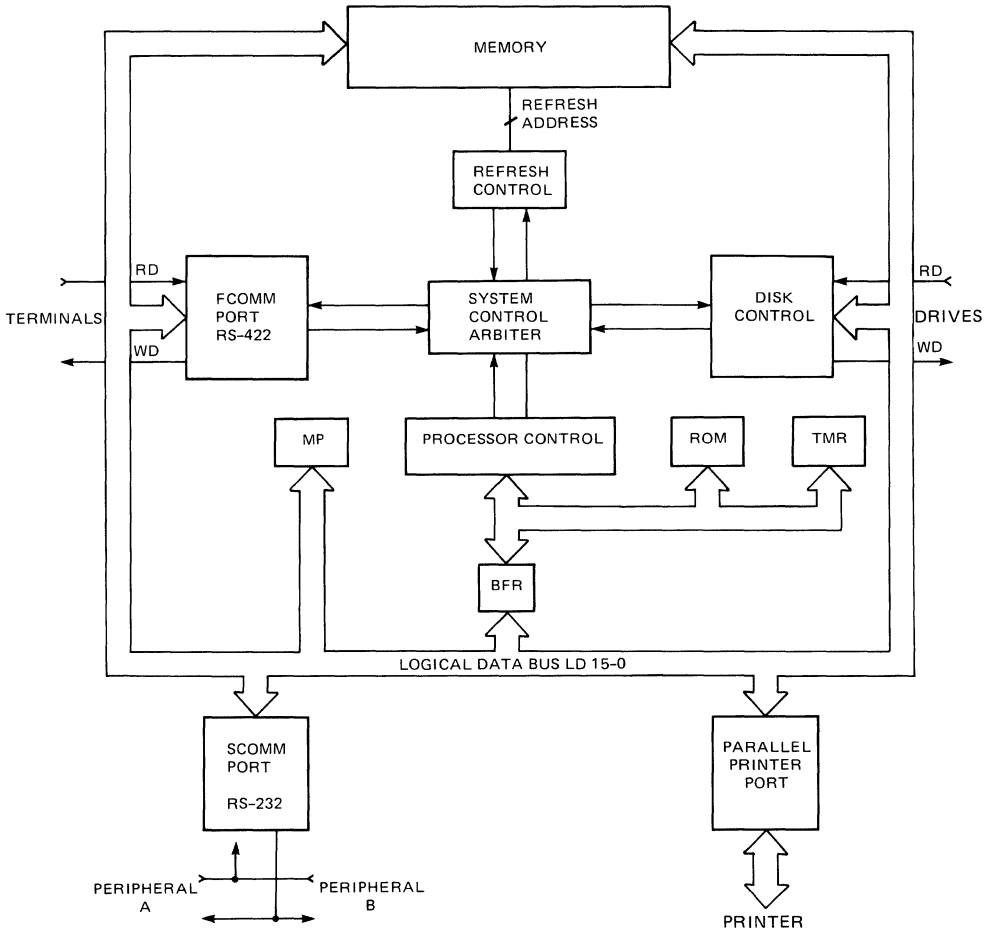


Figure 6-1. System View -- Data Transfers

From this angle, overall system operation appears as several different types of data movement, with CPU processing occuring invisibly. A model of system activity starts with the CPU processing user code. If the CPU addresses a information not in memory, a page map error results and calls the supervisor. The supervisor sets up a disk DMA operation, then returns the CPU to user mode to run a different user process (if one needs to be run). Meanwhile the disk control does the DMA operation.

NOTE

Whenever the computer is powered on, the refresh control, at timed intervals, requests control of the bus. Even when system activity is at a maximum, the refresh control gets a grant in time for it to refresh the next row in each RAM chip.

During the disk DMA operation, whenever the disk control requests bus control for a DMA transfer, the CPU is blocked out, and the disk control takes over the bus. In parallel with the word transfer to or from memory, the disk control reads or writes serial data from or to the disk.

At maximum system activity, all the input/output devices could be simultaneously active; the disk control, external port, and fast communications could do DMA, the line printer interface and slow communications could do interrupt driven memory transfers, and the refresh control could maintain the dynamic memory.

Our model of system activity includes three basic activities:

- Local transfers (between local elements)
- External transfers (to and from peripherals)
- CPU interrupts.

System input/output includes all three activities. The remainder of this section describes local transfers, system input/output, and CPU interrupts.

LOCAL TRANSFERS

The following are local tranfers:

- CPU to/from memory or the processor or map registers
- The memory access portion of DMA operations
- CPU to/from ROM or the input/output device registers
- CPU interrupt acknowledge cycle.

Transfers between the CPU and the processor map registers are fast cycle transfers (400 ns long), as are the memory access portion of DMA transfers. The remaining transfers are slow cycle transfers (1000 ns long).

At the beginning of a transfer, the bus arbiter grants bus control to the highest priority device requesting a transfer. By definition, local transfers commence at the positive edge of PCLK+, and positive transitions of PCLK- trail PCLK+ by 50 ns (that is, they occur at 50 ns, 150 ns, 250 ns ... into the transfer cycle).

Fast Cycle Transfers

This section describes (1) a CPU/memory access, (2) the difference between the CPU/memory acccess and DMA transfer, and (3) a CPU/register access.

CPU/Memory Access

Figure 6-2 is a timing diagram of a CPU/memory access.

To commence a transfer, at 60 ns the CPU outputs the transfer function code on FC0, PROGRAM+, and SUPV+, and if the access is a read, negates WRITE-. At 110 ns, the CPU places the memory address on the processor address bus, PA23-1. PA23-22 equals 00 (pointing to memory). PA23 = 0 indicates the transfer is a fast cycle transfer. At 120 ns, the CPU asserts the address strobe (AS-), and if the access is a read, asserts one or both data strobes, UDS- and LDS-.

The main address decoder (PAL 1, sheet 2), after decoding the processor address and function codes, outputs the processor address enables (PADENH- and PADENL-).

NOTE

If a DMA transfer is in process, one of the bus arbiter DMA grant signals is active. It blocks PADENH- and PADENL-, by asserting NPCYCLE-, via the miscellaneous logic PAL (PAL 36, sheet 4).

The processor address enable signals route the address and bus control signals to the logical address bus (LA21-1, sheet 28) and logical bus control (L control signals, sheet 28).

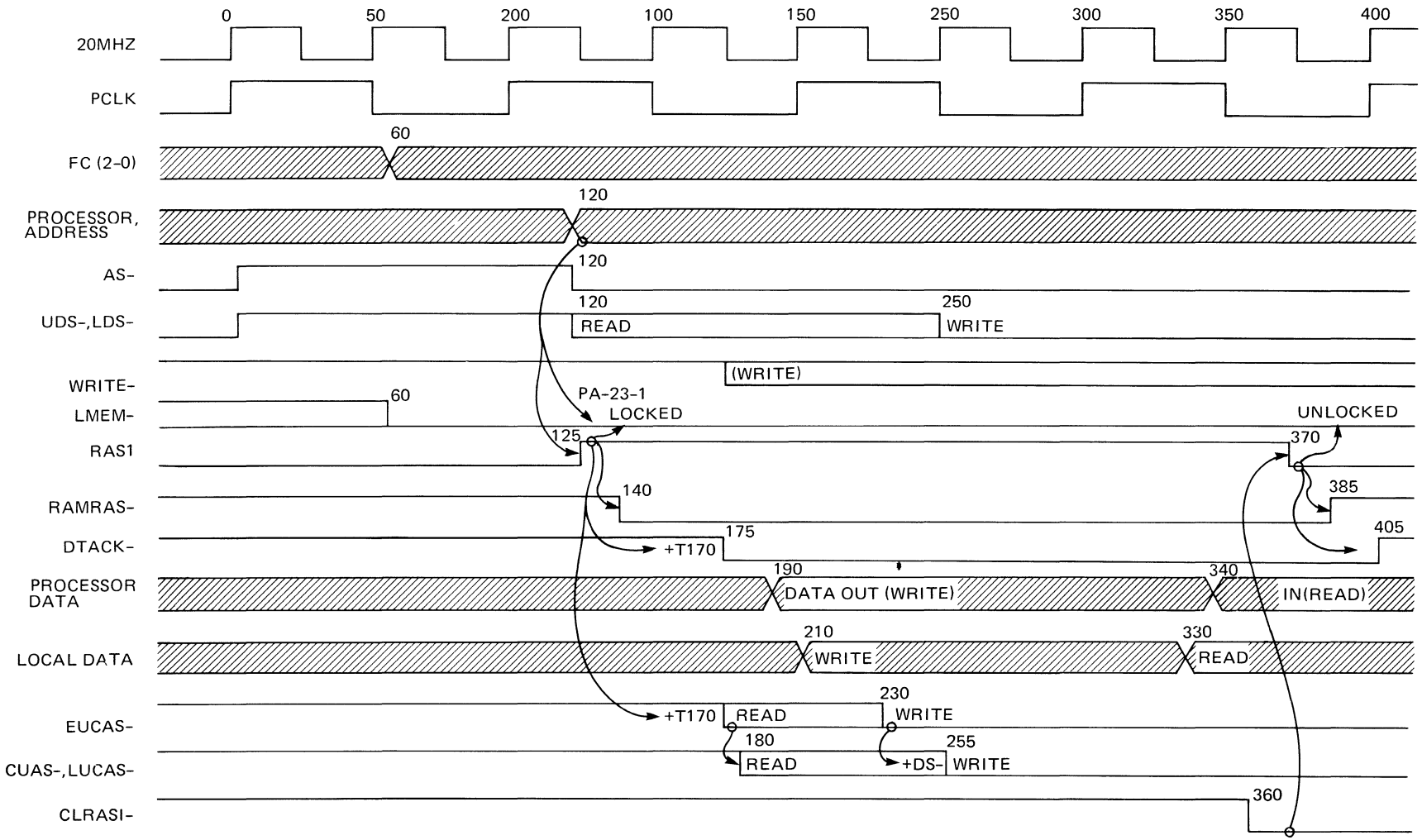


Figure 6-2. Fast Cycle Transfers -- CPU Memory Access

LA21-12 addresses the map RAMs. The memory address latch (sheet 27) receives the map RAM outputs and holds the latched map address (LMA2Ø-12) and the latched logical address bits 2-1 (LLA2-1). If LMA2Ø-19 are ØØ, they generate BASEMEM-, which tells the memory control that onboard memory is being addressed. If expansion memory is being addressed, LMA2Ø-19 select the expansion board: Ø1 for board 1, 1Ø for board 2, 11 for board 3.

At 125 ns, 2ØMH- clocks the RAS1 flip/flop (sheet 29), and with LMEM- and PA23 low, the flip/flop asserts RAS1-. RAS1- tells the bus arbiter to lock the default grant (LMEM-) low and secure the CPU fast cycle selection. RAS1- also synchronizes the CPU state machine (sheet 29), bus control (sheet 4), and memory control PAL (sheet 19) for the fast cycle transfer.

NOTE

When the computer is idle, at every PCLK-, the bus arbiter (PAL 5, sheet 4), sends the default grant, LMEM-, to the CPU state machine (PAL 21, sheet 29). If the system is active (DMA or refresh), the arbiter negates LMEM-, which blocks RAS1-. Without RAS1-, the CPU state machine does not assert the CPU acknowledge, DTACK-. The CPU enters wait states until the bus arbiter asserts LMEM-, which generates RAS1-, and allows the CPU state machine to assert DTACK-.

RAS1-, via RAS+, triggers the multitap delay input to the memory control PAL (PAL 27) and is decoded with LLA2-1 to generate a RAM bank row address strobe (either RAMRAS3-, RAMRAS2-, RAMRAS1-, or RAMRASØ-). At 15Ø ns, RAS1-, via SRAS1-, places the CPU state machine in state 1.

The active row addres strobe does two things: it routes the internal chip row address (LA1Ø-3) through the base memory address multiplexer (sheet 2Ø) to the RAM address buses (W, X, Y, and Z), and it clocks the address into one of the RAM rows (Ø, 1, 2, or 3, sheets 22 through 25).

At 17Ø ns, the memory control multitap delay outputs T17Ø+. T17Ø+ and SRAS1- cause the CPU state machine to output the transfer acknowledge, DTACK-, to the CPU.

From T17Ø+, the Memory Control PAL asserts ROE-. Since BASEMEM- is asserted, ROE- asserts BØROE-, via PAL 36 (sheet 4). BØROE- enables the Main Processor board M-data buffer (sheet 21). If an expansion board was addressed, B1ROE-, B2ROE-, or B3ROE- would be asserted.

If it is a write, the CPU asserts WRITE- at 175 ns. At about 190 ns, the CPU outputs the write data. The processor data buffer, directed by WRITE- and enabled by PDBOE-, routes the write data to the logical data bus, LD15-0. (PAL 33, sheet 32, holds PDBOE- active except during DMA or a CPU ROM/timer access.)

If it is a read, MCWT- (high) causes the memory control PAL to negate XWE-, which negates MWE-. The memory data buffer routes the read data to the logical data bus. If it is a write, the memory data buffer routes the write data to RAM, via RD15-0.

MWE- also controls the write parity buffer (sheet 21). During a read, MWE- disables the parity buffer, forcing high impedance outputs. The parity RAM supplies RDO17-16.

During a write, MWE- enables the buffer, and, with PS+ high (see note), the buffer drives RDO17-16 to 11 (the value of PS+). RDO17-16 are input with RD15-0 (from the CPU) to the parity circuit. The parity chips send SO, their SUM ODD parity outputs, to RDI17-16 for storage in the parity RAMs.

NOTE

Odd parity is the normal state of data in memory, so the software normally sets PS+ to 1. See Section 5, "Parity".

If it is a read, T170+ causes the memory control PAL to activate ENCAS-. The data strobes are already valid. (If it is a write, T230+ activates ENCAS-, and the data strobes are valid at 250 ns.)

ENCAS- enables one of the upper byte column address lines, UCAS- 3-0, and lower byte column address lines, LCAS- 3-0. If the CPU is accessing only the upper or lower byte, ENCAS- enables only the corresponding upper or lower CAS- line.

T170+ also routes the column address, LMA18-11, to RAM. The column address strobes clock the column address into the selected row of chips. During a write, the strobes also cause the RAMs to latch the data (at 260 ns).

At 250 ns, the CPU notes the DTACK- from the CPU state machine, and if it is a write, outputs the data strobes. Also at 250 ns, the CPU state machine advances from processor state 1 to state 3. The data strobes cause the valid CAS- signal to activate.

At 300 ns, if it is a read, the RAMs output the read data and parity to the memory buffer and the parity circuit. The circuit

checks for odd parity. The memory buffer routes RD15-Ø to LD15-Ø at 32Ø ns, which then goes to PD15-Ø. The CPU latches the data at 35Ø ns.

At 35Ø ns, PCLK- clocks the CPU state machine to state 2. The machine outputs LRDEN- and CLRRAS1-. LRDEN- causes the error control PAL (PAL 25, sheet 17) to assert NMI if a parity error has occurred (PE+ is asserted). CLRRAS1- resets the RAS1 flip/flop, which deasserts RAS1-. SRAS1- is deasserted, and, at about 4ØØ ns, so is DTACK-. Also, the CPU removes AS- and the data strobes at about 4ØØ ns.

The CPU state machine is disabled at the next PCLK- (at 45Ø ns, or 5Ø ns into the next cycle), and the processor control is ready to begin another read/write cycle. Also at 45Ø ns, the bus arbiter notes that RAS1- is high, which tells it that the CPU has finished the transfer. If another request is pending, the arbiter sends a grant to that device; otherwise the arbiter keeps LMEM- active, and an idle cycle commences.

If LMEM- remains active, and the CPU returns RAS1- before the next PCLK-, the CPU again gets bus control for another fast cycle transfer. If the CPU starts a fast cycle transfer, after a DMA transfer has commenced, LMEM- and, correspondingly, RAS1- will be inactive. The CPU, not receiving DTACK-, enters wait states until the system is inactive. Then the arbiter reactivates LMEM-, the CPU locks LMEM- high via RAS1-, and the CPU does the fast cycle transfer.

DMA Accesses

The disk control, fast communications port, and optional external port all do DMA transfers. The refresh control does an abbreviated form of a DMA transfer. This section discusses the local transfer portion of DMA accesses. These begin when the input/output element either has the data ready for memory (during a DMA read) or is ready to receive data from memory (during a DMA write).

When a device is ready for a DMA, it sends a DMA bus request to the bus arbiter. If the CPU is performing a fast cycle or slow cycle transfer, the bus arbiter holds off the DMA bus grant until the CPU deasserts RAS1- (fast cycle) or AAS- (slow cycle). If another input/output device is doing a DMA, the arbiter holds off the grant until that device negates its bus request.

At the PCLK- before the system will become idle the arbiter prioritizes the active bus requests. At the next PCLK-, the arbiter returns the DMA grant to the highest priority device requesting bus control. (By definition, this occurs at 5Ø ns in the DMA cycle.)

The DMA grant goes to the miscellaneous logic PAL (PAL 36, sheet 4) and the input/output device. The PAL outputs SETRAS2+, while the device outputs the memory address and read/write signal. At 125 ns, 2ØMHZ sets the RAS2 flip/flop, and RAS2- goes to the memory control.

RAS2- (as RAS1- does in a CPU fast cycle transfer) generates RAS+. Beyond this point, the timing and control is very similar to a CPU memory access. The memory control receives LA21-1 and LWT- and performs the memory access.

The input/output device removes its bus request at 35Ø ns, and the bus arbiter removes the bus grant at 45Ø ns. (The bus arbiter may issue a new grant at the same time.) Table 6-1 lists DMA data bus interfaces and memory address sources belonging to the different input/output control elements.

Table 6-1. DMA Data Interfaces and Memory Address Sources

Device	Data Interface	Address Source
Disk FCOMM Refresh	Disk word buffer (sh. 8) Data buffer (sh. 11)	Disk Word Address (sh. 9) Word Address (sh. 12) Refresh Address (sh. 2Ø)

CPU/Register Transfers

CPU accesses to the processor registers and map registers have the same timing as CPU/memory accesses. Figure 6-2, the timing diagram for a CPU/memory access, is thus applicable to the following desription.

To commence a processor or map register access, at 6Ø ns, the CPU outputs the transfer function code on FCØ, PROGRAM+, and SUPV+, and if the access is a read, negates WRITE-. At 11Ø ns, the CPU places the register address on PA23-1. PA23-22 equals Ø1 to indicate the transfer is a processor or map register transfer. The low PA23 indicates the transfer is a fast cycle transfer.

The main address decoder (PAL 1, sheet 2), after decoding the processor function code and address, outputs PROCREG- and either MPREGSEL- (for an processor register access) or MMUSEL- (for a map access). The interrupt level generator (PAL 23) outputs PADENH- and PADENL-.

The Processor address enable (PAD) signals route the processor address bus and bus control signals to the logical address (LA) bus and logical (L) bus control (sheet 28). PROCREG- informs the memory control to ignore the access.

At 125 ns, the RAS1 flip/flop is clocked and, with PA23 low and LMEM- active, it asserts RAS1-. RAS1-, the CPU fast cycle acknowledge to the bus arbiter, synchronizes the arbiter, CPU state machine (sheet 29), miscellaneous logic PAL (PAL 36, sheet 4) and memory control (sheet 19) for the fast cycle transfer.

The bus arbiter locks LMEM- low, securing the CPU fast cycle selection. RAS1-, via RAS+, triggers the multitap delay (sheet 19) for timing control. At 150 ns, RAS1-, via SRAS1-, is clocked into the CPU state machine and places the machine in processor state 1.

At 155 ns, the CPU asserts address strobe (AS-) and if the access is a read, outputs the data strobes, UDS- and LDS-. At 160 ns, if the access is a write, the CPU asserts WRITE-. At 170 ns, the multitap delay outputs T170+. T170+ and SRAS1- cause the CPU state machine to send DTACK- to the CPU.

If MPREGSEL- is active, the CPU register decoder (PAL 2, sheet 2) decodes PA19-16 and outputs the corresponding register read signal. (Two signals if two bytes of the 4-byte Bus Status register are being read). The addressed processor register outputs its contents to the logical data bus, LD15-0.

If MMUSEL- is active, the map logic (sheet 27) is activated. The map output buffers, enabled by PADLTCH- (high), send MCWT- (from LWT-) to the map control PAL (PAL 29, sheet 27). The map control PAL outputs MALDEN, which enables the map RAMs. The PAL also outputs MALDEN-, which enables the map data buffer. MMUSEL- also routes PA10-1 to LA21-12, which addresses the map RAMs.

NOTE

The map addresses are shifted so that they appear to the CPU to be contiguous. This allows the CPU to load the maps in fast loop mode, provided the page tables in memory are 16-bits wide.

If the access is a write, at about 200 ns, the CPU outputs the write data. The processor data buffer, directed by WRITE- and enabled by PDBOE-, routes the write data to LD15-0. (PAL 33, sheet 32, holds PDBOE- active.)

At 250 ns, the CPU clocks in DTACK-. If the access is a read, the CPU prepares to receive the data. If the access is a write, at 250 ns, the CPU outputs the data strobes. The strobes direct the CPU register decoder or the map control PAL (PAL 29, sheet 27) to output a read or write signal to the addressed register or map RAMs. (If the access is a read, the read signals will already

have been active.) Also, at 25Ø ns the CPU state machine advances from processor state 1 to processor state 3.

At 3ØØ ns, PCLK clocks the CPU. If it is a read, within 4Ø ns the CPU latches the read data. At 35Ø ns, PCLK- clocks the CPU state machine to state 2. The CPU state machine outputs CLRRAS1- and removes PRSØ. CLRRAS1- resets the RAS1 flip/flop, which deasserts RAS1-. SRAS1- is deasserted, and, at about 4ØØ ns, so is DTACK-. Also, the CPU removes AS- and the data strobes at about 4ØØ ns.

The CPU state machine is deactivated at the next PCLK- (at 45Ø, or 5Ø, ns), and the process control is ready to begin another read/write cycle. At the same time, this PCLK- and RAS1- high tell the bus arbiter that the CPU has finished the transfer.

If no other device has a request active at this time, the bus arbiter keeps LMEM- active, and the idle cycle recommences. **NOTE:** If another device has a request active, the arbiter negates LMEM- and instead sends a grant to the other device.

If LMEM- stays active, and the CPU returns RAS1- before the next PCLK-, the CPU again gets bus control for another fast cycle transfer. If the CPU attempts to start a fast cycle transfer after a DMA transfer has commenced, LMEM- and, correspondingly, RAS1- will be inactive. The CPU enters wait states until the system is inactive. When the arbiter reactivates LMEM-, the CPU locks LMEM- high via RAS1-, and the CPU does the fast cycle transfer.

Slow Cycle Transfers

There are two basic types of slow cycle transfers: (1) CPU accesses to ROM, to the timers, or to input/output device registers and (2) the interrupt acknowledge cycle.

CPU Input/Output Cycle Transfers

CPU input/output cycle transfers are accesses to the timers, input/output device registers, or ROM. They are all very similar and are described together here. Figure 6-3 is a timing diagram of an input/output cycle.

To commence the transfer, at 6Ø ns the CPU outputs the transfer function code on FCØ, PROGRAM+, and SUPV+. If the access is a read, the CPU negates WRITE-. At 11Ø ns, the CPU places the address on PA23-1. PA23-22 equal 11Ø for a ROM access and 11 for a timer or input/output device register access. (A special case of ROM access is described in the note below.) PA23 high indicates the access is a slow cycle transfer.

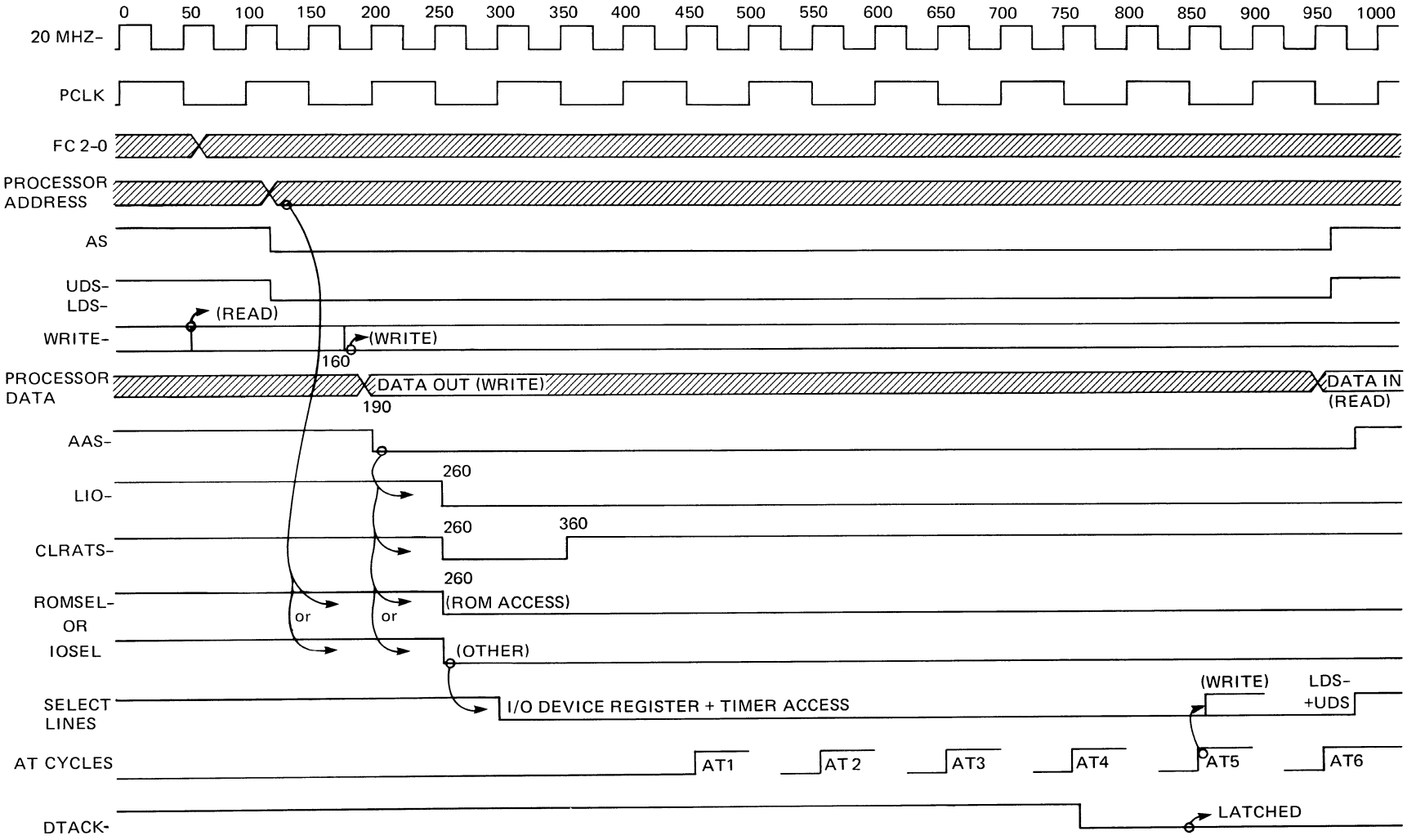


Figure 6-3. CPU Slow Cycle (Input/Output) Transfer

At 12Ø ns, the CPU asserts address strobe (AS-), and if the access is a read, asserts one or both of the data strobes, UDS- and LDS-. The main address decoder outputs the processor address enables, PADENH- and PADENL-. PADENH- and PADENL- route the processor address and bus control signals through the processor address buffers to the logical address (LA) bus and logical (L) bus control (sheet 28).

At roughly 19Ø ns, if the access is a write, the CPU outputs the write data. At 2ØØ ns, AS- and at least one of UDS- or LDS- generate AAS- (sheet 29). AAS-, the slow cycle transfer request to the bus arbiter (PAL 5, sheet 4), synchronizes the bus control and either ROM, timer, or device register read/write control for the slow cycle transfer. The bus arbiter decodes AAS- and ROMEN-. For a slow cycle, either ROMEN- is low (see note below), or AAS- is asserted.

NOTE

A computer power up or reset clears the General Control register (sheet 3Ø), causing it to assert ROMEN- (bit 12). The CPU, receiving RESET, automatically outputs the vector address $ØØØØØØ. ROMEN- causes the bus arbiter to route the lower address bits of $ØØØØØØ to the ROM, rather than RAM memory. At the ROM location, the CPU picks up a stack pointer and program counter, which branch it to a legitimate ROM address. At the legitimate address, the boostrap program resets ROMEN-, allowing the CPU to again access RAM.

At 25Ø ns, the arbiter asserts CLRATS- (which immediately clears the at time sequencer) and LIO-. LIO-, PA23-16, and AAS- go to the Device Select PAL (sheet 2). The PAL sends either ROMSEL- to the ROM (sheet 3) or IOSEL- to the two I/O address decode PALS (PALs 3 and 4, sheet 2).

If ROMSEL- is active (selected by PA23-22 = 1Ø), the ROMs are enabled by PAL 1, and addressed by PA13-1. If IOSEL- is active, PALs 3 and 4 decode PA19-16 and other control signals, then assert one of the input/output device select lines.

At 35Ø ns, PCLK- causes the bus arbiter to deassert CLRATS-, allowing subsequent PCLK-s to assert successive AT signals. At

450 ns, the at time sequencer asserts AT1+. At each succeding PCLK-, the sequencer continues sending the current AT signal and asserts the next one. The AT signals provide timing to the interrupt control (sheet 17) as well as the read/write circuitry.

A delay occurs before the addressed device outputs the read data, or if it is a write, is ready to receive the write data. At 760 ns, AT4+ is input with LIO- to the interrupt acknowledge PAL (PAL24 sheet 17). The PAL outputs IODTK- to the CPU state machine, which then sends DTACK- to the CPU. At 850 ns, the CPU clocks in DTACK-.

If the access is a write to an input/output device register or timer, at 860 ns, AT5+ removes the select line from the enabled I/O address decode PAL; the PAL negates the write signal, and the addressed timer or device register latches the data. If it is a read, the register or timer has already output the data; at 950 ns, the CPU latches the data and removes the data strobes; at 980 ns, PAL 3 or PAL 4 removes the select lines.

At 970 ns, the CPU removes AS-, which clears AAS-. At the next PCLK- (1050 ns), the bus arbiter negates LIO-, and the device select PAL negates ROMSEL- or IOSEL-. The CPU removing AAS- has freed the local system for another local transfer and (at 1050 ns) if a DMA request is pending the bus arbiter outputs a new bus grant. If no request is pending the arbiter outputs LMEM-, the idle cycle default grant to the CPU.

Interrupt Acknowledge Cycle

The interrupt acknowledge cycle is similar to a CPU slow cycle read. The CPU, receiving a nonzero interrupt request (IPL) field, outputs an address with PA23-4 all high and PA3-1 equal to the interrupt request level that it is currently acknowledging. The function code, FC0, PROGRAM+, and SUPV+ equals 111 to indicate the CPU cycle is an interrupt acknowledge cycle.

The main address decoder (PAL 1, sheet 2) decodes PA23-16 and FCHI. These are all high, and the PAL outputs INTAADDR-. At 200 ns, the AAS- flip/flop sets, and INTADDR, PA3-1, and AAS- cause the interrupt acknowledge PAL (PAL 24, sheet 17) to assert the acknowledge signal that corresponds to the interrupting device. If it's a nonmaskable interrupt, the PAL clears the NMI- flip/flop.

If the interrupt is an autovector interrupt, PAL 28 (sheet 26) asserts VPA-. VPA-, which replaces DTACK-, informs the CPU that the interrupt cycle is an autovector cycle. Any other interrupt is a normal vector interrupt, in which case the device requesting the interrupt begins accessing the vector byte.

NOTE

If the interrupt is an autovector interrupt, the interrupt acknowledge cycle has ended. The CPU generates its own vector and branches to the corresponding interrupt routine where it begins servicing the interrupt. The description beyond this point refers to a normal vector interrupt.

At 250 ns, the bus arbiter decodes AAS- and asserts CLRATS- and LIO-. CLRATS- resets the at time sequencer. At 350 ns, the bus arbiter removes CLRATS-. During the next four PCLK-s, while the device is getting the vector byte, the sequencer advances from AT1 to AT4. At 750 ns, AT4 and LIO- cause PAL 24 to assert IODTK-. The CPU state machine, receiving IODTK-, asserts DTACK-.

At roughly 770 ns, the device receiving the interrupt acknowledge places the vector byte on bits 7-0 of the logical data bus (either directly or via SCD7-0). At 850 ns, the CPU latches DTACK-. At 950 ns, the CPU latches the data. At 970 ns, the CPU removes AS-, which clears AAS-. The CPU also removes the address.

At 980 ns, the device select PAL removes INTADDR-. At 990 ns, the interrupt acknowledge PAL removes the acknowledge signal and negates IODTK-. The CPU state machine negates DTACK-. The CPU multiplies the interrupt vector by four to obtain the branch address, then branches to the interrupt routine.

At the next PCLK- (1050 ns), the bus arbiter negates LIO-, and the device select PAL negates INTADDR-. The CPU removing AAS- has freed the local system for another local transfer and (at 1050 ns) if a DMA request is pending the bus arbiter outputs a new bus grant. If no request is pending the arbiter outputs LMEM-, the idle cycle default grant to the CPU.

Double Interrupt Acknowledge Pulse

The 8274 RS-232-C port controller and the 8259 interrupt controller respond to CPU interrupt acknowledge cycles differently than other input/output devices, as they require two

interrupt acknowledge pulses to complete the cycle. For these devices, the interrupt acknowledge signal pulses from active (between 250 and 500 ns) to inactive (between 500 and 750 ns) to active (between 750 and 1000 ns) and back to inactive (after 1000 ns). The first pulse tells the device to get the vector byte ready. The second pulse tells it to output the byte.

The double interrupt acknowledge pulse circuit, described in Section 5, generates the double pulse.

SYSTEM INPUT/OUTPUT OPERATIONS

The following input/output devices do system input/output operations:

- Disk control (DMA)
- Fast communications port (DMA)
- External port (DMA)
- Slow communications port (CPU interrupt driven)
- Line printer interface (CPU interrupt driven).

The remainder of this section describes the system input/output operations.

Disk Control Input/Output

There are four types of disk control input/output operations, as follows:

- Hard disk write
- Floppy disk write
- Hard disk read
- Floppy disk read.

Disk input/output operations involve the system control, processor control, disk control, memory control, and interrupt control. Refer to Section 4, "Setting up a Disk DMA Operation", for a description of how the CPU sets up and starts a disk DMA operation.

Disk input/output operations comprise three types of transfers:

- Local transfers between the disk bus interface unit and the memory control

- Byte transfers between the disk bus interface unit and the selected disk controller

- The serial data stream between the disk controller and the peripheral device.

Refer to the "DMA Accesses" subsection earlier in Section 6 for a description of the memory control portion of the disk/memory local transfer. The rest of the disk/memory local transfer and the transfer to and from the disk controllers are described below in terms of the states that the disk word state machine and disk byte state machine sequence through.

The disk word state machine consists of the disk word state sequencer (PAL 35) and the disk word decoder (PAL 1Ø). The disk byte state machine consists of the disk byte state sequencer (PAL 8), the #1 disk byte decoder (PAL 9), and the #2 disk byte decoder (PAL 7).

The disk word state sequencer outputs bits W3-Ø to define the current state of the disk word state machine. The disk byte state sequencer outputs B3-Ø to define the current state of the disk byte state machine.

In the following descriptions, the state number refers to the value of bits W3-Ø or B3-Ø after PCLK- clocks the state machine. Following the state number is a one or two sentence summary of what occurs during the state. After the summary is a detailed circuit description of the state. All timing is with respect to the PCLK- that latches the new W or B field to start the state.

Disk Write Operation

After the CPU sets up the DMA control for the disk write, the DMA controller sends DSKDMAWR- to the disk word and disk byte state machines, and the Disk Control register (PAL 6, 23L, sheet 7) outputs DSKDMAEN- to the word machine. If it is a hard disk transfer, the Disk Control register also asserts HDDMA-.

Figure 6-4, included in the word state machine description, is a state sequencing/timing diagram depicting the state machine operation as it does a memory read, then transfers the word to the disk byte state machine. Figures 6-5 and 6-6, included in the byte state machine description, are state sequencing/timing diagrams depicting the disk byte state machine operation as it obtains the word from the disk word state machine and routes it, via two byte transfers, to the disk controller. Figure 6-5 pertains to the floppy disk controller, and Figure 6-6 pertains to the hard disk controller.

Disk Word State Machine

The disk word state machine begins in state Ø, the idle state.

STATE Ø: When the CPU enables the DMA write operation, the machine goes to state 1.

W 3-Ø equals ØØØØ. The disk word state machine receives DSKDMAEN- from the Disk Control register and DSKDMAWR- from the DMA controller. At the next PCLK- the machine goes to state 1. (See Figure 6-4.)

STATE 1: The disk word state machine requests bus control from the bus arbiter. When the arbiter grants bus control, the disk word machine sets up the disk word count and disk word address register clocks and goes to state 3. The bus request remains asserted.

W 3-Ø equals ØØØ1. The disk word state machine sends DSKDMAREQ- to the bus arbiter. If a local transfer is in progress, the arbiter ignores the request until the transfer finishes. If a transfer has just finished or the system is idle (and if the optional external port hasn't a request pending), the arbiter sends DGNT- back.

The DMA cycle has commenced and is at 5Ø ns when the arbiter sends DGNT-. DGNT- asserts (drives low) DCNTCLK-, DADLOCLK-, and DADHICLK-, via the DMA controller PAL (PAL 113, 23H, sheet 9). At the next PCLK- (15Ø ns), DGNT- advances the machine to state 3.

STATE 3: The disk word state machine waits one clock, then goes to state 2.

W 3-Ø equals ØØ11. At the next PCLK- (+25Ø ns), the disk word state machine goes to state 2.

STATE 2: The disk word state machine sets up the clock to the A-registers in the disk word buffer and, at the next PCLK-, goes to state 4.

W 3-Ø equals ØØ1Ø. The disk word decoder asserts WBCLKA-. At the next PCLK- (+35Ø ns), the disk word state machine goes to state 4.

STATE 4: The disk word state machine clocks the word from memory into the A-register in the disk word buffer (via LD15-Ø), and removes the bus request. At the next clock, the bus arbiter removes DGNT-, which clocks the word count, register, word address register, and terminal count flip/flop.

The disk word state machine waits for the disk byte state machine to assert BREQ-. (BREQ- may already be asserted.) At the PCLK- after the disk byte state machine asserts BREQ-, the disk word state machine returns BACK- and goes to state 5.

W 3-Ø equals Ø1ØØ. The machine negates WBCLKA-, causing the A-register to clock in the write data. The bus arbiter removes DGNT-. This negates DCNTCLK-, DADLOCLK-, and DADHICLK-, which causes the disk word count and disk word address registers to increment.

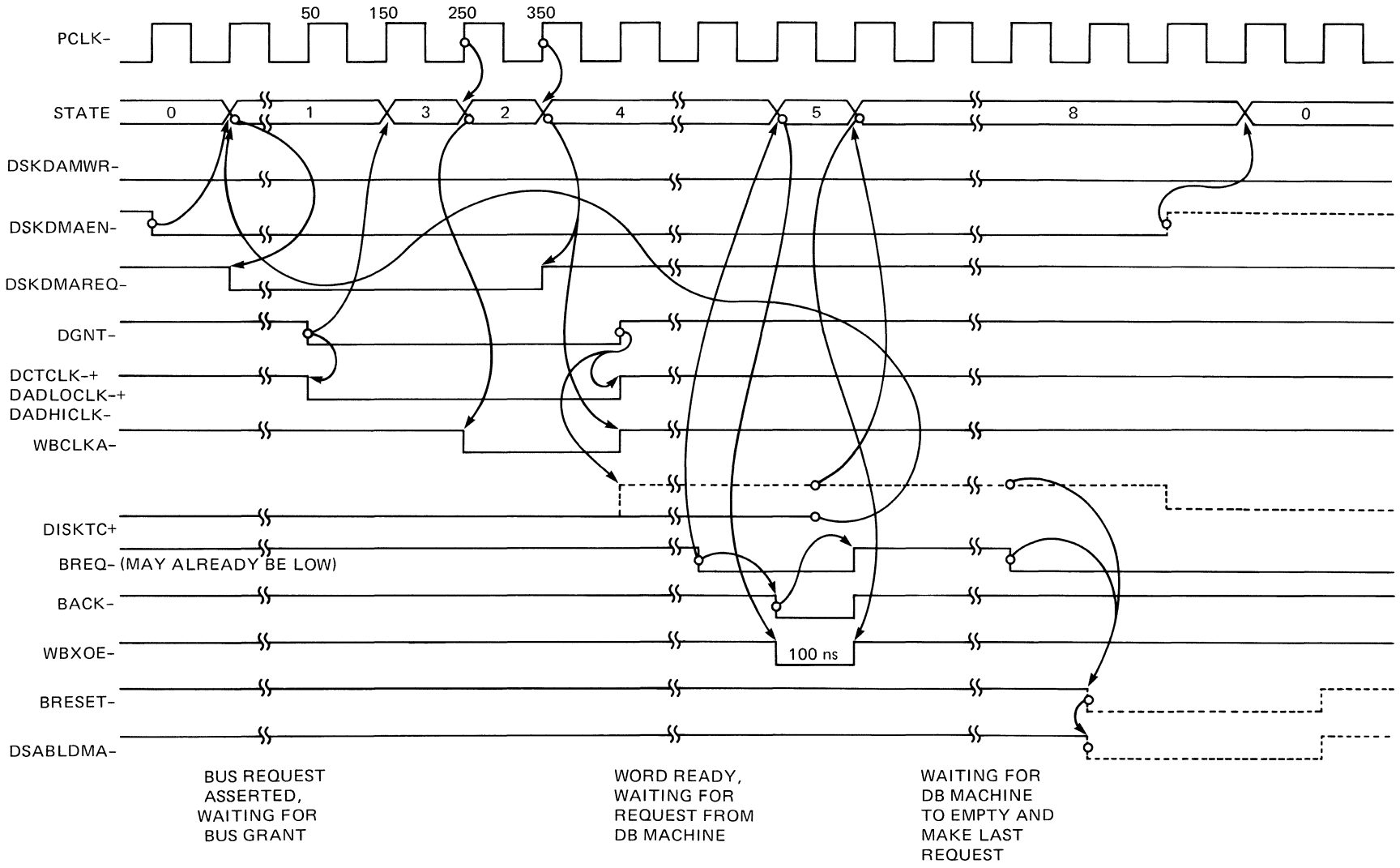


Figure 6-4. Disk Word State Machine -- Disk Write

DGNT- negated also clocks the terminal count flip/flop (25M, sheet 9). If this is the last word to be transferred in the disk write operation, the flip/flop asserts DISKTC+.

On the PCLK- after the disk byte state machine asserts BREQ-, the disk word state machine returns BACK- and goes to state 5. (If BREQ- is already asserted, as it might at the beginning of the DMA operation, state 4 lasts one clock cycle, and the disk word state machine goes to state 5 at the same time that DGNT- goes high.) (See Figures 6-5 and 6-6.)

STATE 5: The disk word state machine transfers the word to the disk byte state machine, then determines if the DMA operation is completed by checking the terminal count signal. If the terminal count signal is asserted, at the next PCLK- the disk word state machine goes to state 8; otherwise it returns to state 1. In either case, at the same time, the disk byte state machine removes the word request, and the disk word state machine removes the acknowledge.

W 3-Ø equals Ø1Ø1. The disk word state machine asserts WBØOE- and WB1OE-, which routes the A-register contents through the B-bus drivers and out to JD 15-Ø.

At the next PCLK-, the disk byte state machine removes BREQ- and the disk word state machine removes BACK-. Also at the next PCLK-, if DISKTC+ from the DMA controller is not set, the disk word state machine advances to state 1 and repeats the sequence. If DISKTC+ is set, the disk word buffer has received the last word in the disk write operation, and the disk word state machine advances to state 8. (See Figure 6-4.)

STATE 8: This is the termination state for the DMA operation. The disk word state machine waits for the disk byte state machine to empty the disk byte buffer and send another request. On the clock after the disk byte state machine sends the request, the disk word state machine sends a reset to the disk byte state machine. The reset causes the disk byte state machine to disable the DMA operation, which drives the disk word state machine back to state Ø.

W 3-Ø equals 1ØØØ. After a delay the disk byte state machine asserts BREQ-, indicating it is empty. At the next PCLK-, the disk word state machine asserts BRESET-. At the next PCLK-, the disk word state machine removes BRESET-, but not before BRESET- causes the disk byte state machine to disable the DMA operation by sending DISABLDMA- to the Disk Control register. DISABLDMA- causes the Disk Control register to negate DSKDMAEN-, which, at the next PCLK-, returns the disk word state machine to state Ø.

Disk Byte State Machine

The disk byte state machine begins in state Ø, the idle state.

STATE Ø: When enabled by the CPU, the machine goes to state 1.

B 3-Ø equals ØØØØ. At the beginning of the disk operation, the CPU sets DSKDMAEN- and DSKDMAWR-. At the next PCLK-, the disk byte state machine goes to state 1.

NOTE

If the disk byte state machine receives SYNCDDREQ- while in states Ø or 1, the hard disk controller has sent a data request before the machine is ready to transfer. The controller generates a data overrun error, while the machine sends DBINT- to the interrupt control (sheet 18), and goes to state 8. The disk byte state machine stays in state 8 until the CPU resets the disk bus interface unit, then returns to state Ø.

STATE 1: The disk byte state machine sets up the clock to the A-registers in the disk byte buffer and requests data from the disk word state machine. When the machine gets an acknowledge from the disk word state machine, it clocks in the data and goes to state 2. If the DMA operation is over, the disk word state machine returns a reset instead of the ackowledge. The disk byte state machine then disables the DMA operation.

B 3-Ø equals ØØØ1. At the next PCLK-, the disk byte state machine asserts DBØ1CLKA- and BREQ-. The disk byte state machine then waits for BACK- from the disk word state machine. At the PCLK- after the disk word state machine issues BACK-, the disk byte state machine goes to state 2 (at 55Ø ns, in the first transfer). (See Figures 6-5 and 6-6).

If the last word has already been transferred, the disk word state machine returns BRESET- instead of BACK-. At the next PCLK- the disk byte state machine asserts DSABLDMA-. At the following PCLK- the byte machine returns to state Ø, and the Disk Control register negates DSKDMAEN-. The negated DSKDMAEN- also returns the disk word state machine to state Ø.

STATE 2: The disk byte state machine clocks in the word from the disk word state machine, then waits for the first byte request from the disk controller. When it receives the request, the machine goes to state 6.

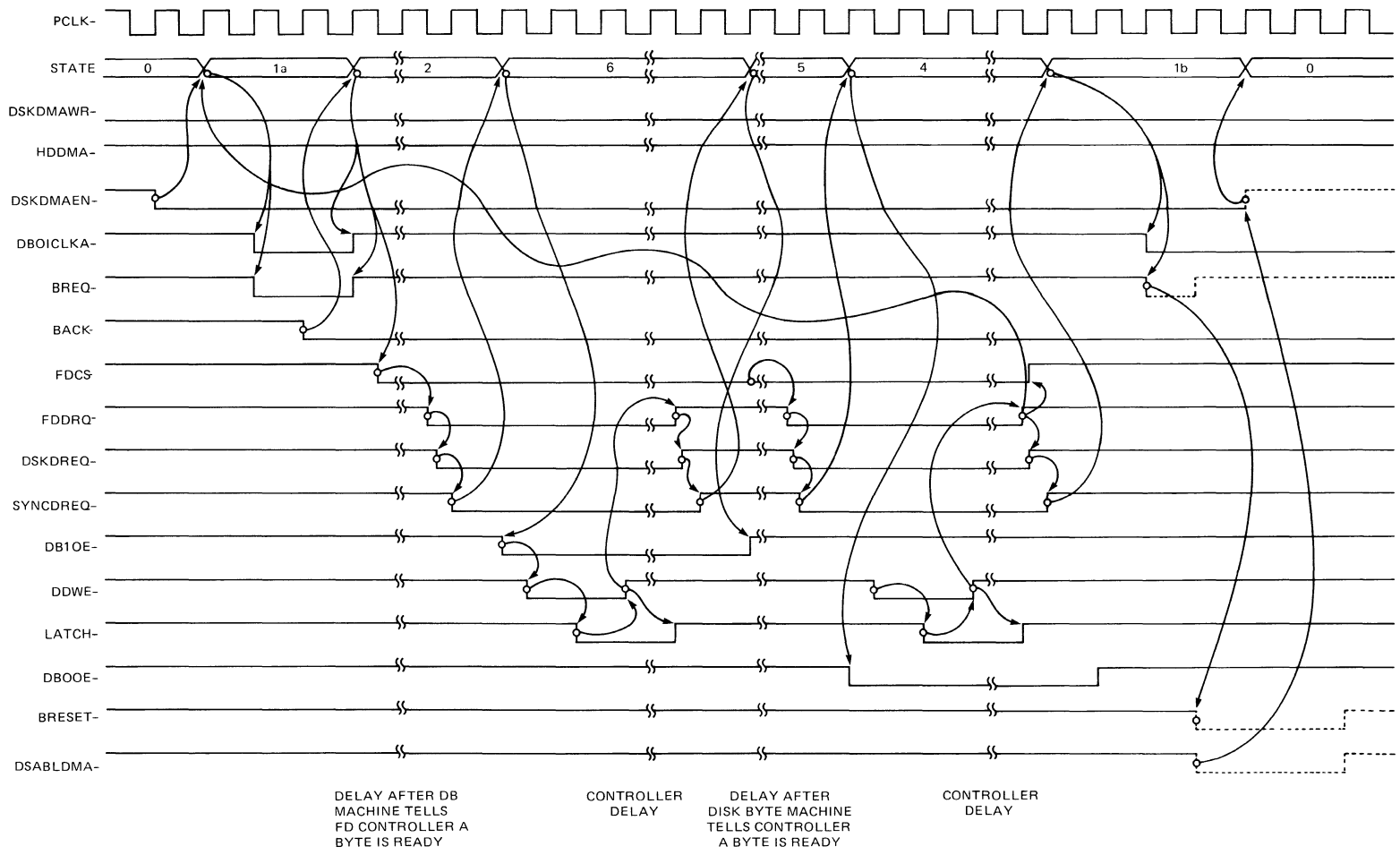


Figure 6-5. Disk Byte State Machine -- Floppy Disk Write

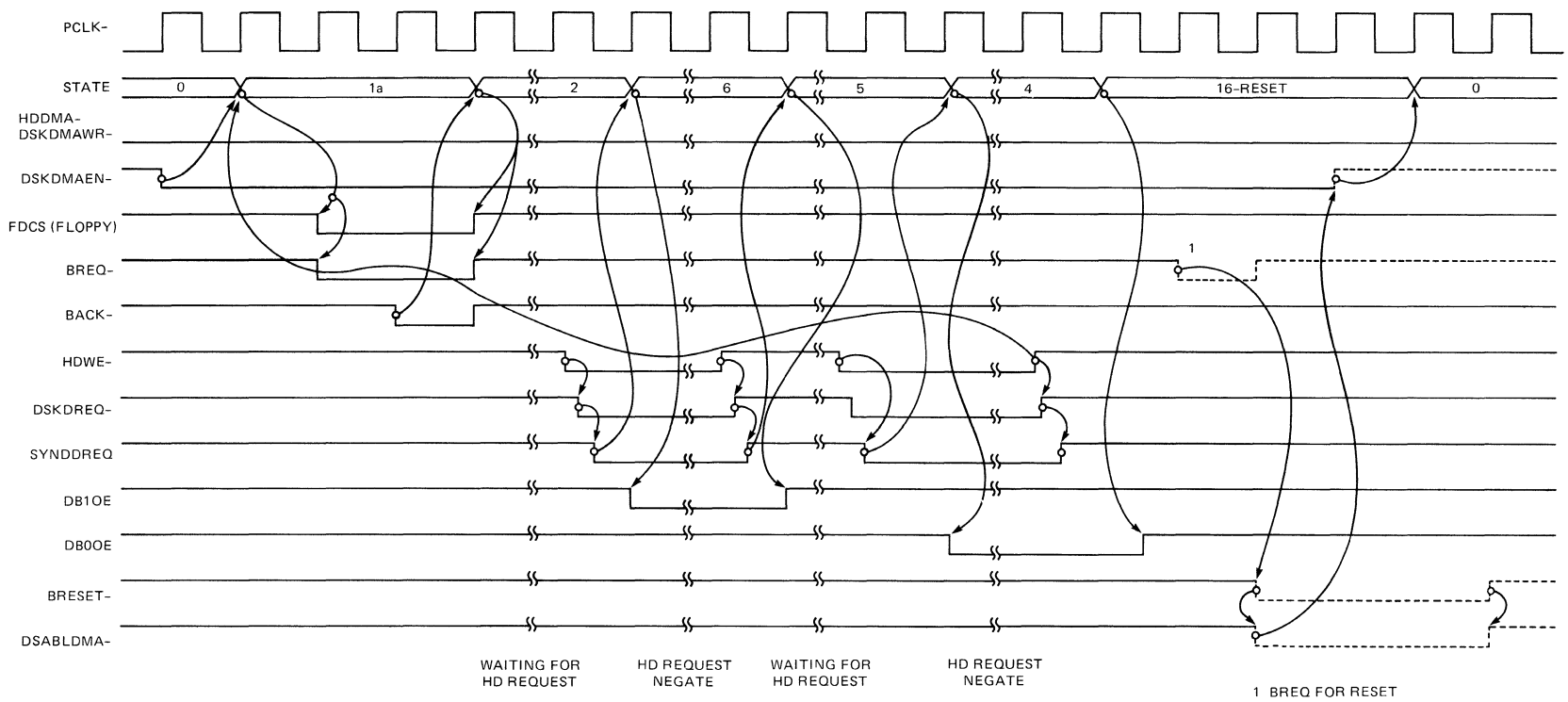


Figure 6-6. Disk Byte State Machine -- Hard Disk Write

B 3-Ø equals ØØ1Ø. The machine negates BREQ-. The machine also negates DBØ1CLKA-, which clocks the word from the A-registers in the disk word buffer into the A-registers in the disk byte buffer.

The disk byte decoder #1 looks at HDDMA-, from the Disk Control register, to see whether the transfer is to the floppy disk or hard disk. If the transfer is to the floppy disk (HDDMA- is negated), 5Ø ns later the decoder asserts FDCS-. (Note that decoder #1 is clocked by PCLK not PCLK-.) (See Figures 6-5 and 6-6.)

When the floppy disk controller requires a byte of data, it asserts FDDRQ-, causing the disk interrupt PAL (PAL 11, sheet 8) to assert DSKDRQ-. DSKDRQ- is synchronized with 2ØMHZ- (22M, sheet 7) to generate SYNCDDREQ-.

If the transfer is to the hard disk (HDDMA- is asserted), after the hard disk controller asserts HDRE-, the disk interrupt PAL asserts DSKDRQ-. DSKDRQ- is synchronized with 2ØMHZ- (22M, sheet 7) to generate SYNCDDREQ-. At the next clock, the disk byte state machine goes to state 6. (If it is a floppy disk write, FDCS- remains asserted.)

STATE 6: The disk byte state machine outputs byte 1 (the high byte) and waits for the disk controller to latch the data and negate the data request. When the disk controller does this, the machine goes to state 5.

B 3-Ø equals Ø11Ø. The byte machine asserts DB1OE-. DB1ØE- (active) and DB1DRVA- (inactive) enable the high byte bus drivers. The upper byte buffer is hardwired to select the A-latch, so the A-latch data is routed through the drivers and out to DD 7-Ø.

If the transfer is to the floppy disk, at PCLK (5Ø ns after the disk byte state machine enters state 6), byte decoder #2 asserts DDWE-, the write enable input to the floppy disk controller. One PCLK later, the decoder asserts LATCH-, a feedback signal.

At the next PCLK, the decoder removes DDWE-. The floppy disk controller clocks in the data and removes FDDRQ- some time later. The disk interrupt PAL removes DSKDRQ-, then 2ØMHZ- resets SYNCDDREQ-. At the next PCLK-, the disk byte state machine goes to state 5. FDCS- remains asserted.

If the transfer is to the hard disk, after a delay, the hard disk controller removes HDWE-. The disk interrupt PAL removes DSKDRQ-, then 2ØMHZ- resets SYNCDDREQ-. At the next PCLK-, the byte machine goes to state 5. FDCS- remains asserted.

STATE 5: The disk byte state machine waits for the second byte request from the disk controller. When it receives the request the machine goes to state 4.

B 3-Ø equals Ø1Ø1. Byte decoder #2 negates DB1OE-. If HDDMA- is negated, after the floppy disk controller asserts FDDRQ-, the disk interrupt PAL (PAL 11, sheet 8) reasserts DSKDRQ, which is synchronized with 2ØMHZ- to assert SYNCDDREQ-. (FDCS- remains asserted.)

If HDDMA- is asserted, after the hard disk controller reasserts HDRE-, the disk interrupt PAL (PAL 11, sheet 8) reasserts DSKDRQ, which is synchronized with 2ØMHZ- to assert SYNCDDREQ-. At the next PCLK-, the disk byte state machine goes to state 4.

STATE 4: The disk byte state machine outputs the second byte. When the disk controller latches the byte and remove the data request, the machine returns to state 1.

B 3-Ø equals Ø1ØØ. Byte decoder #2 asserts DBØOE-, which, with DBØDRVA- inactive, enables the low byte buffer B-bus drivers. DBØASEL- is inactive, so the A-register data is routed through the drivers and out to DD 7-Ø.

If the transfer is to the floppy, 5Ø ns later, the byte decoder #2 asserts DDWE-. Using LATCH- as a feedback, two PCLKs later, the decoder removes DDWE-. The floppy disk controller latches the data and removes FDDRQ-. FDCS- and DBØOE- are negated. If the transfer is to the hard disk, the hard disk controller removes HDWE-.

Either the negated FDDRQ- or HDWE- cause the disk interrupt PAL to negate DSKDRQ-, which, again synchronized with 2ØMHZ-, negates SYNCDDREQ-. At the next PCLK-, the disk byte state machine returns to state 1.

Disk Read Operation

After the CPU sets up the DMA control for the disk read, the DMA controller negates DSKDMAWR-, and the Disk Control register (PAL 6, sheet 7) outputs DSKDMAEN- to the disk word state machine. If it is a hard disk operation, the Disk Control register also asserts HDDMA-, and the hard disk controller asserts HDBCS-.

Figures 6-7 and 6-8, at the end of the disk byte state machine description, are state/timing diagrams of the machine operation as it obtains two bytes from the disk controller, then transfers a word to the disk word state machine. Figure 6-7 pertains to the floppy disk controller, and Figure 6-8 pertains to the hard disk controller. Figure 6-9, at the end of the disk word state machine description, is a state/timing diagram of the disk word state machine operation as it receives the word from the disk byte state machine, then does a memory write.

Disk Byte State Machine

The disk byte state machine begins in state Ø, the idle state.

STATE Ø: When the CPU enables the DMA disk write operation, the byte machine goes to state 2.

B 3-Ø equals ØØØØ. The machine receives DSKDMAEN-. DSKDMAWR- is negated. At the next PCLK-, the byte machine goes to state 2. (See Figures 6-7 and 6-8.)

NOTE

If the disk byte state machine receives SYNCDDREQ- while in states Ø or 1, it means the hard disk controller has sent a data request before the machine is ready to transfer. The controller generates a data overrun error, while the machine sends DBINT- to the interrupt control (sheet 18) and goes to state 8. The machine stays in state 8 until the CPU resets the disk bus interface unit. Then the machine returns to state Ø.

STATE 2: If it is a floppy disk transfer, the disk byte state machine enables the floppy disk controller. (If it is a hard disk transfer, the hard disk controller is already enabled.) When the byte machine receives the high byte request from the disk controller, the byte machine goes to state 6. If the disk byte state machine has just transferred the last word to the disk word state machine, the disk word state machine resets the disk byte state machine, and the disk byte state machine returns it to state Ø.

B 3-Ø equals ØØ11. If HDDMA- is negated, at PCLK the byte decoder #2 asserts FDCS-. Some time later the floppy disk controller returns FDDRQ-, which causes the disk interrupt PAL to assert DSDKRQ-. If HDDMA- is asserted, when the hard disk controller asserts HDWE-, the disk interrupt PAL (PAL 11, sheet 21) asserts DSKDRQ-. DSKDRQ- is synchronized with 2ØMHZ- to generate SYNCDDREQ-. At the next PCLK-, SYNCDDREQ- sends the disk byte state machine to state 6.

If the word machine received the last word during the previous disk byte to disk word transfer, the disk word state machine will have asserted BRESET- before PCLK generated FDCS-. BRESET- returns the disk byte state machine to state Ø.

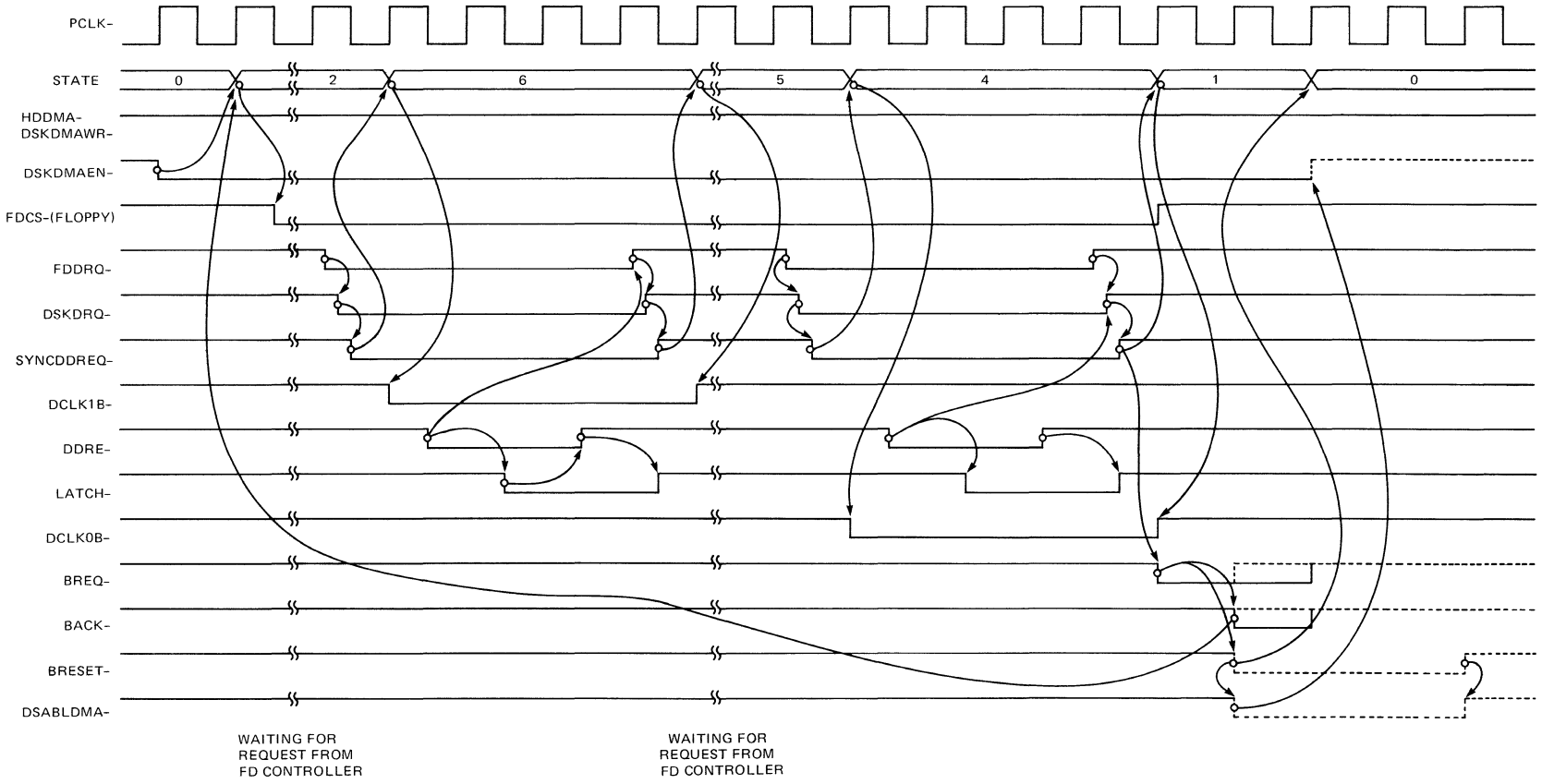


Figure 6-7. Disk Byte State Machine -- Floppy Disk Read

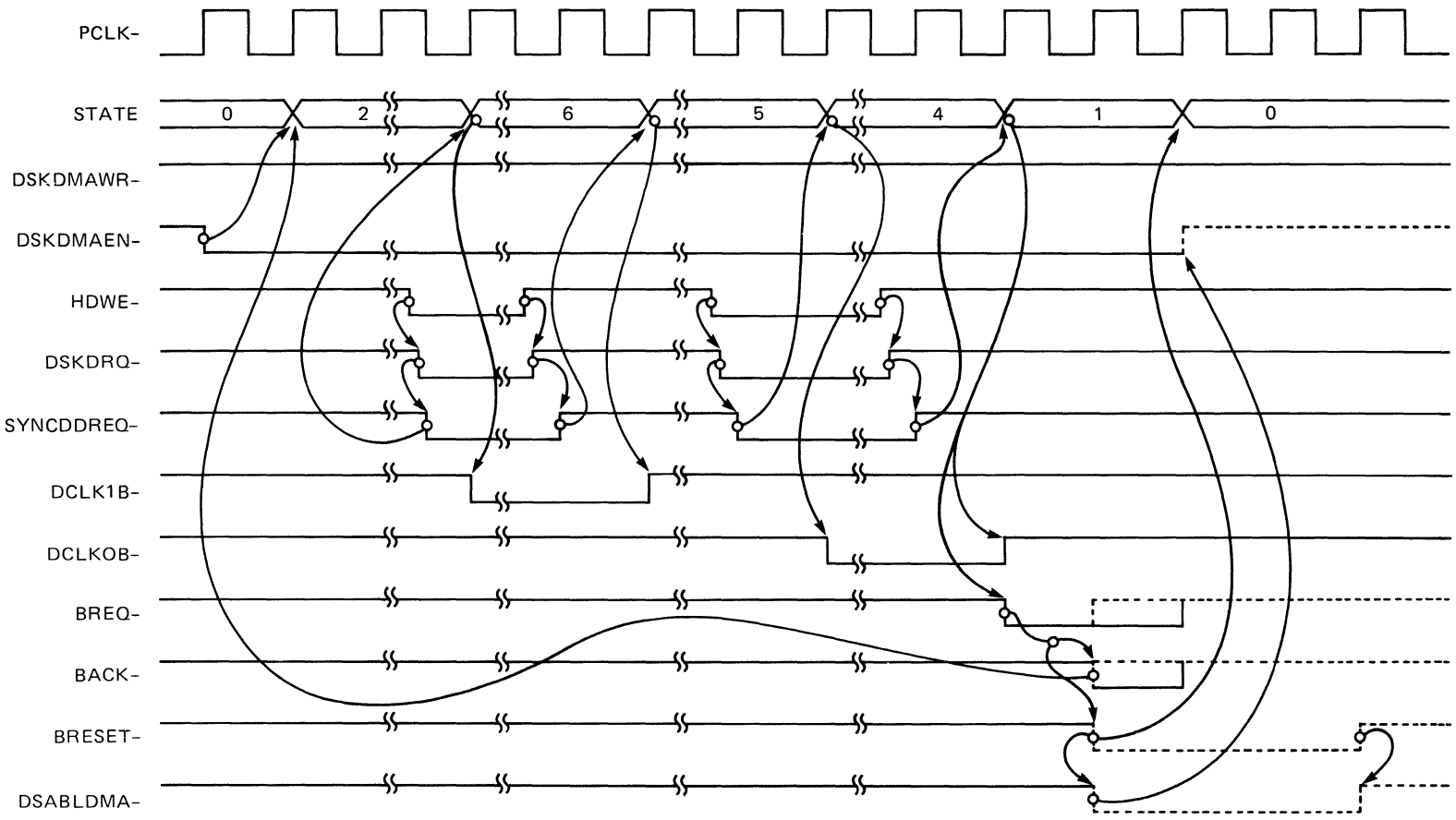


Figure 6-8. Disk Byte State Machine -- DMA Hard Disk Read

STATE 6: The disk byte state machine sets up the clock to the B-registers in the upper disk byte buffer. When the disk controller remove the data request, the machine goes to state 5. (If the transfer ias to the hard disk, the machine latches the high byte.)

B 3-Ø equals Ø11Ø. Byte decoder #1 asserts DB1CLKB-. If HDDMA- is negated, at PCLK (5Ø ns later), byte decoder #2 asserts DDRE-, the read enable input to the floppy disk controller. One PCLK later, byte decoder #2 asserts LATCH- a feedback signal. At the next PCLK, LATCH causes the #2 decoder to remove DDRE-. Shortly later, the floppy disk controller removes FDDRQ-, and, within 5Ø ns, 2ØMHZ- removes SYNCDDREQ-. At the next PCLK-, the machine goes to state 5. FDCS- remains asserted.

If HDDMA- is asserted, when the hard disk controller negates HDWE-, byte decoder #1 negates DB1CLKB-, and the upper B-register in the disk byte buffer clocks in the high byte from the hard disk controller. Within 5Ø ns, 2ØMHZ- removes SYNCDDREQ-, and, at the next PCLK-, the disk byte state machine goes to state 5. (See Figures 6-7 and 6-8.)

STATE 5: If the transfer is to the floppy disk, the byte machine latches the high byte. At the clock after the byte machine receives the low byte request from either the hard disk or floppy disk controller, the machine goes to state 4.

B 3-Ø equals Ø1Ø1. If the transfer is to the floppy disk, byte decoder #1 negates DB1CLKB-, and the upper B-register in the byte buffer clocks in the high byte the from the floppy disk controller. With FDCS- still asserted, the floppy disk controller asserts FDDRQ- when it has byte #Ø assembled. FDDRQ- causes the disk interrupt PAL to assert DSDKRQ-. DSKDRQ- is synchronized with 2ØMHZ- to generate SYNCDDREQ-. At the next PCLK-, the byte machine goes to state 4. FDCS- remains asserted.

If the transfer is to the hard disk, when the hard disk controller has the low byte assembled, it asserts DDWE-, and the disk interrupt PAL (PAL 11, sheet 21) asserts DSKDRQ-. DSKDRQ- is synchronized with 2ØMHZ- to generate SYNCDDREQ-. At the next PCLK-, the byte machine goes to state 4.

STATE 4: The disk byte state machine sets up the clock to the B-register in the lower disk byte buffer and waits for the disk controller to remove the low byte request. When the controller removes the request, the machine clocks the low byte into the B-register in the lower disk byte buffer, sends a data request to the disk word state machine, and goes to state 1.

B 3-Ø equals Ø1ØØ. Byte decoder #1 asserts DBØCLKB-. At PCLK (5Ø ns later), byte decoder #2 asserts DDRE-. One PCLK later, byte decoder #2 asserts LATCH-. At the next PCLK, LATCH

removes DDRE-, and, shortly later, the floppy disk controller removes FDDRQ-. The disk interrupt PAL removes DSKDRQ-, and, within 50 ns, 20MHZ- removes SYNCDDREQ-. The negated SYNCDDREQ- causes the disk byte machine to go to assert BREQ- and, at the next PCLK-, go to state 1.

For a hard disk transfer, when the hard disk controller negates HDWE-, the disk interrupt PAL removes DSKDRQ-, and byte decoder #1 negates DB0CLKB-. The B-register in the lower disk byte buffer clocks in the low byte from the disk controller. Within 50 ns, 20MHZ- removes SYNCDDREQ-. The negated SYNCDDREQ- causes the disk byte state machine to assert BREQ- and, at the next PCLK-, to go to state 1.

STATE 1: The disk byte buffer clocks in the low byte from the disk controller. When the disk word state machine acknowledges the transfer, the disk byte state machine returns to state 2. If the disk word state machine returns a reset instead of an acknowledge, the disk byte state machine disables the DMA operation and returns to state 0.

B 3-0 equals 0001. Byte decoder #1 negates DB0CLKB-, and the B-register in the lower disk byte buffer clocks in the low byte from the disk controller. FDCS- is negated. At the next PCLK- after the disk word state machine sends BACK-, the disk byte state machine returns to state 2.

If the transfer has terminated, the disk word state machine returns BRESET- instead of BACK-. The disk byte state machine asserts DSABLDMA- and returns to state 0. DSABLDMA- causes the disk bus interface unit register to negate DSKDMAEN-.

Disk Word State Machine

The disk word state machine begins in state 0, the idle state.

STATE 0: When the CPU enables the DMA read, the machine goes to state 4.

W 3-0 equals 0000. The machine receives DSKDMAEN- and DSKDMAWR- (high) and, at the next PCLK-, goes to state 4. (See Figure 6-9.)

STATE 4: If the disk word state machine is already doing a transfer (returning from state 2), it removes the bus request. One clock later, the arbiter removes the bus grant, which disables the disk byte buffer and clocks the DMA controller registers. If this was the final transfer, the disk word state machine resets the disk byte state machine and returns to state 0. If this was not the final transfer, at the clock after the disk byte state machine requests a transfer, the disk word state machine returns an acknowledge and goes to state 5.

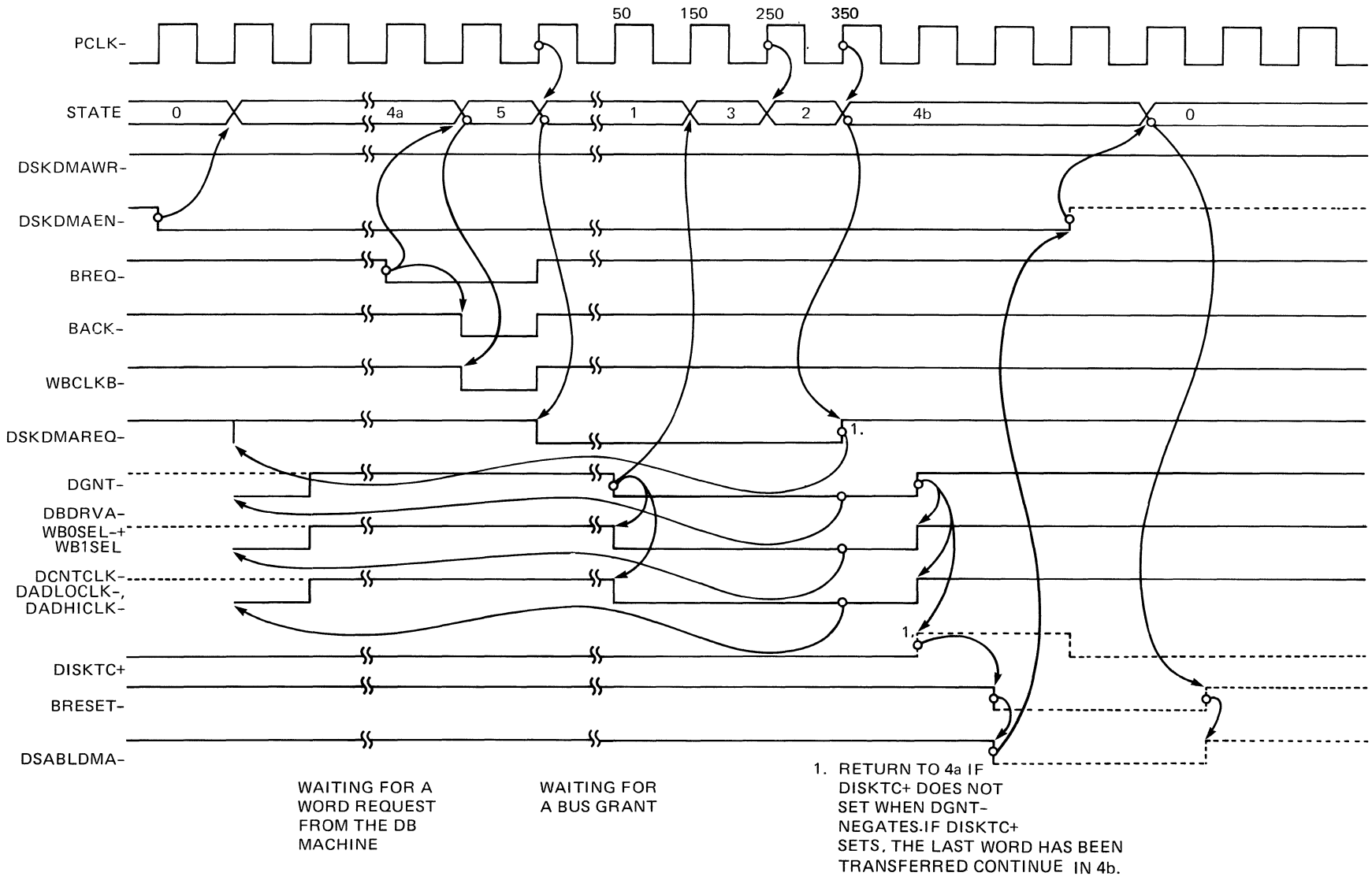


Figure 6-9. Disk Word State Machine -- DMA Disk Read

The disk word decoder removes DSKDMAREQ-. At the next PCLK-, the bus arbiter removes DGNT-, which negates WBDRVA-, WB1OE-, and WBØOE- and clocks the word count register, word address register, and terminal count flip/flop. DISKTC+ is asserted if the disk byte interface unit has received the last word from memory. (See Figure 6-9.)

If DISKTC+ is not set, on the PCLK- after the disk byte state machine asserts BREQ-, the disk word state machine asserts BACK- and goes to state 5. If DISKTC+ is set, at the next PCLK-, the disk word state machine asserts BRESET-. At the next PCLK-, the disk word state machine removes BRESET-, but not before BRESET- causes the disk byte state machine to disable the DMA operation (by sending DISABLDMA- to the Disk bus interface control register). DISABLDMA- causes the register to negate DSKDMAEN-. At the next PCLK-, DKSDMAEN- returns the disk word state machine to state Ø.

STATE 5: The disk word state machine sets up the clock to the B-registers in the disk word buffer and goes to state 1.

W 3-Ø equals Ø1Ø1. The decoder asserts WBCLKB-. At the next PCLK-, the disk byte state machine removes BREQ-, while the disk word state machine removes BACK- and goes to state 1.

STATE 1: The disk word state machine clocks in the data and sends a bus request to the bus arbiter. When the arbiter returns the bus grant, the machine sets up the word count and word address register clocks, enables the data onto the logical data bus and, at the next clock, goes to state 3.

W 3-Ø equals ØØØ1. The disk word state machine asserts DSKDMAREQ- to request bus control. The decoder negates WBCLKB-, causing the disk word buffer to latch the word from the disk byte state machine. When the bus arbiter returns DGNT-, the disk word decoder asserts WBDRVA-, WB2OE- and WBØOE- and negates WBOBSEL-. The disk word buffer drives LD15-Ø with the read data word.

DGNT- also asserts DCNTCLK-, DADLOCLK-, and DADHICLK-, via the disk DMA controller PAL (PAL 13, sheet 9). At the next PCLK- (15Ø ns into the memory write cycle), the disk word state machine goes to state 3. DSKDMAREQ- remains asserted.

STATE 3: The disk word state machine waits one clock, then goes to state 2.

W 3-Ø equals ØØ11. At the next PCLK- (+25Ø ns) the machine goes to state 2.

STATE 2: At the next PCLK-, the machine returns to state 4.
W 3-Ø equals ØØ1Ø. At the next PCLK- (+35Ø ns), the disk word state machine returns to state 4.

Fast Communications Port Input/Output

The fast communications port operates in three basic modes:

- DMA read (RS-422 to memory)
- DMA write (memory to the RS-422 port)
- CPU/Port controller register read/write.

As it does with the disk control, the CPU sets up DMA operations by initializing the fast communications DMA controller, fast communications bus interface unit, and port controller. The CPU does this via several slow cycle writes. After it has set up the fast communications port, the CPU lets the port do the DMA operation. The port interrupts the CPU when the it has finished.

One major difference between the fast communications port and the disk control is that the fast communications port receives data from peripherals without requiring a request from the CPU. Instead, an intelligent terminal or other computer informs the CPU that it needs servicing by sending a packet of information to memory, via the port. When the fast communications port receives a packet, it interrupts the CPU and the CPU retrieves the packet from memory.

Technically, the CPU must request the information packet first by setting up the fast communications port for a DMA read whenever the port is idle. However, unlike during a DMA write, the CPU does not know the length of the packet or when or from whom the packet will come.

The CPU sets up the fast communications port so that it can receive the longest packet possible before it generates the DMA terminal count signal. After it receives a complete packet and routes it to memory, the port generates a CPU interrupt. If the CPU hasn't masked out the interrupt, the CPU looks at the packet (in memory) and, depending on what the packet contains, vectors to one or another service routine.

The current operating system sets up the fast communications port to perform DMA transfers in byte mode. The fast communications bus interface unit reads or writes the lower byte of the Received Data/Status register and Transmit Data/Status register, which contains the DMA data. The unit does not read the upper byte, which contains status.

Fast Communications Write

Once the CPU has set up the fast communications port for a fast communications DMA write, the port takes over. The Fast communications bus interface unit requests bus control and, gaining control, does a memory read. The unit writes one byte to the port controller.

The port controller moves the byte to the serial out register and begins shifting out the serial data stream. At the same time, the bus interface unit writes another byte to the port controller, then does another memory read.

This continues until the DMA controller generates the terminal count signal. The bus interface unit resets the port controller and generates an end of transfer CPU interrupt. When it is able, the CPU services the interrupt and, at some point in the routine, sets up the port in read (receive packet) mode.

Fast Communications Read

Once the CPU has set up the port for a fast communications DMA read, the port takes over. The port controller begins receiving serial data from the RS-422 line. After assembing one byte, the controller alerts the Fast communications bus interface unit, and the unit does a byte read. The unit must do the read before the next byte is assembled, or an error results.

The controller assembles another byte, and the bus interface unit does another read. Now having a word, the unit requests bus control, and when it gains control, writes the word to memory. In the meantime, the controller assembles another byte.

This continues until the DMA controller generates a terminal count signal. The fast communications bus interface unit resets the port controller and generates a CPU interrupt. When it is able, the CPU services the interrupt and, at some point in the routine, sets up the port in read (receive packet) mode.

Odd Byte Read/Writes

If, during a DMA read, the fast communications state machine receives a terminal count signal while it is in an odd byte state, it means the RS-422 port is doing an odd byte boundary DMA read. Instead of resetting the port, the fast communications bus interface unit generates a CPU interrupt and does one more memory write. The bus interface unit then resets as normal.

Slow Communications Port Input/Output

The slow communications port performs input/output operations differently than the disk control or fast communications port; slow commuinications input/output are not DMA, rather CPU interrupt directed.

The operation separates into three parts, as follows:

WRITE	READ
1. CPU interrupt	1. CPU interrupt
2. CPU memory read	2. CPU/port data read
3. CPU/port data write	3. CPU memory write

Before a slow communications external input/output operation begins, if the operation involves transferring data between memory and the port, the CPU sets up a control table in memory. (The transfer may not involve memory if, for example, the CPU is receiving command interpreted or status information from the port.) The control table contains a word counter and memory address register. After setting up the control table, the CPU sets up the port for the transfer.

If it is an external input operation, the port begins reading in data from the peripheral and requests CPU intervention when the port has filled its data register. If it is an external output operation, the port immediately requests CPU intervention; the 8274 asserts IRQ8274-. IRQ8274- goes to the interrupt level generator (PAL 23, sheet 17), which then asserts the level four priority interrupt (IPL field) if there is no higher priority request present.

If the CPU has not masked out level four interrupts, it does an interrupt acknowledge cycle that selects the level four interrupt. The interrupt acknowledge PAL (PAL 24, sheet 17) routes the acknowledge to the 8274 serial controller, which then returns the slow communication port's interrupt vector. The CPU reads the vector, computes the branch address, and branches to the interrupt routine.

If the port is in the process of an external output operation, the CPU does a memory read to get the data and also updates the control table. The CPU then writes the data to the port.

The port serializes and routes the data out to the peripheral, following normal RS-232-C protocol. When it has emptied its data register, the 8274 requests another byte from the CPU by reasserting IRQ8274-. When the CPU determines that all of the data has been transferred (during the control table update), the CPU resets the port control.

If the port is in the process of an external input operation, the port reads in a byte and interrupts the CPU. The CPU reads in the byte, updates the control table, and writes the byte to memory. When the CPU determines that all of the data has been transferred it resets the port control.

Line Printer Interface Input/Output

On the system level, the line printer interface operation is similar to the slow communications port; however, the line printer does not supply the CPU with an interrupt vector. Also, the line printer is a slave device.

To execute a print process, the supervisor sets up a control table in memory. The CPU reads the first byte from memory, then writes it to the line printer interface. This write transfer simultaneously loads the line printer data out buffer and generates LPSTROBE-. After receiving the byte, the printer returns a data acknowledge, LPALP-, which generates an interrupt.

The interrupt control (See "Interrupt Control", Section 5) requests an autovectored interrupt and, when the CPU selects the line printer interface for an interrupt acknowledge cycle, returns INTALP-. During the interrupt routine, the supervisor checks the control table in memory to see if there is still data to be transferred, and if there is, obtains another byte from memory and sends it to the printer. This continues until the CPU has sent all of the data to the printer.

CPU INTERRUPTS

CPU interrupts play two important roles in the MiniFrame system. Transfers between memory and either the slow communications port or line printer interface are initiated by the port via a CPU interrupt. Interrupts also alert the supervisor to an existing system error condition. Section 5 detailed the interrupt priority scheme and interrupt circuitry. This section describes two types of interrupt sequences: autovector and normal vector.

At the beginning of the interrupt sequence, somewhere in the system, an interrupt condition is generated. If the interrupt condition is an input to the error control PAL (PAL 25, sheet 17), the error control PAL determines if it should be allowed (if EE is set), and if so, sends NMI- to the nonmaskable interrupt flip/flop. The flip/flop sends NMIFF- through the interrupt synchronizing latch (sheet 17), which then outputs SNMI- to the interrupt level generator (PAL 23, sheet 18).

If the condition is one that activates an input to the 8259 Interrupt controller (sheet 18), the controller checks if that interrupt has been masked out, and if it hasn't, sends IRQ8259A- to the interrupt control synchronizing latch. The latch sends S59B- to the interrupt level generator. The interrupt synchronizing latch also routes other onboard control device interrupts and the expansion port interrupt to the interrupt level generator.

When the generator receives an active interrupt, it immediately asserts the IPL 2-Ø field corresponding to the interrupt priority level of the input. The CPU receives the interrupt priority field and compares it to the current CPU priority level. If the interrupt level is greater than the CPU's level (or is seven), the CPU finishes the current instruction, then commences an interrupt acknowledge cycle. The cycle is described under "Local Transfers".

APPENDIX A: PAL LISTINGS

This appendix presents an overview of the PAL listings, followed by the listings themselves. Each listing includes a brief description of the PAL.

PAL OVERVIEW

Figure A-1, below, is a sample PAL listing. Following Figure A-1 is an explanation of the listing and some hints on PAL listing decoding.

```
PAL16R4A   PAL DESIGN SPECIFICATION
313A
FOR MP,23H 9/24/83


/PCLK LA1 LA2 LA3 LA4 /DGNT /LWT /DDMASEL /RESET GND /OE /DCNTOE
DADHICLK /DADHICTRL /DADLOCTRL /DCNTCTRL /DSKDMAWR DADLOCLK
DCNTCLK VCC


IF (VCC) /DADLOCLK = DDMASEL * LWT * /LA4 * /LA3 * /LA2 * LA1 +
DGNT

IF (VCC) /DADHICLK = DDMASEL * LWT * /LA4 * /LA3 * LA2 * LA1 +
DDMASEL * LWT * /LA4 *  LA3 * /LA2 * /LA1 +
                       DGNT

DSKDMAWR := DDMASEL * LWT * /LA4 * /LA3 *  LA2 *  LA1 +
            DSKDMAWR * /DDMASEL +
            DSKDMAWR * /LWT +
            DSKDMAWR *  LA4 +
            DSKDMAWR * /LA3 +
            DSKDMAWR *  LA2 +
            DSKDMAWR *  LA1 +
            RESET
```

Figure A-1. Sample PAL Listing

Explanation of Sample PAL Listing

This subsection explains Figure A-1.

PAL Type

PAL16R4A is the PAL type. Refer to the specifications provided from various manufacturers for information on the PAL.

PAL Identifier Code

313A is the PAL identifier code. The leading number is the board revision level. The next two numbers are the PAL number. The letter is the revision level of the PAL for that board. In the body of this manual, PALs are referred simply by the PAL number.

Board Identification and PAL Placement

FOR MP,23H 9/24/83 identifies the PAL as being at location 23H on the Main Processor board. The date is the date of the last board or PAL revision.

Inputs and Outputs

/PCLK LA1 LA2 LA3 LA4 /DGNT /LWT /DDMASEL /RESET GND /OE /DCNTOE DADHICLK /DADHICTRL /DADLOCTRL /DCNTCTRL /DSKDMAWR DADLOCLK DCNTCLK VCC

These are the PAL inputs and outputs, starting at the top left, with pin one, and increasing to the right. No pin is left out. The slash sign / indicates the signal is active low. If the PAL is clocked, the clock is always input to pin 1.

PAL Equations

The PAL equations indicate which combination (or combinations) of inputs (asserted or negated) are required to assert (or negate) an output. The slash indicates the input or output is negated. These equations don't tell whether a signal is logical high or low. (Refer to the inputs and outputs portion of the listing to determine this.) When the output in an equation is negated, the output stays asserted until the equation is satisfied.

Following the equation output is (1) an equal sign, then (2) one or more groups of inputs. Each group of inputs is separated from the preceeding group by a plus sign (and from the next group, if there is one). Each group defines one collection of inputs which asserts (or negates) the output.

There are two types of outputs: register and latched. Unclocked PALs have registered outputs; when the inputs satisfy an equation, the PAL registers this and immediately asserts the corresponding output.

Clocked PALs have latched outputs. (Clocked PALs may also include registered outputs.) When the inputs to a clocked equation are valid at the clock's rising edge, the PAL asserts (or negates) the corresponding output. The output remains unchanged until the next clock, when the PAL adjusts it according to the current input levels. **NOTE:** the CPU asserts or negates many of the PAL outputs by performing data independent writes. Latched outputs have separate assert an negate addresses. Registered outputs have only one CPU addess and are active only during the write cycle.

Sample Registered Output Equation

```
IF (VCC) /DADHICLK = DDMASEL * LWT * /LA4 * /LA3 * LA2 * LA1 +
DDMASEL * LWT * /LA4 *  LA3 * /LA2 * /LA1 +
                    DGNT
```

In this registered output equation, "IF (VCC) /DADHICLK" identifies the output as DADHICLK-. The slash in front of DADHICLK- tells you that the output is normally asserted (high, according to the inputs/outputs portion of the listing) and goes low only when the inputs satisfy the equation.

The two pluses following the equation sign indicate there are three ways to negate DADHICLK-: (a), when the CPU writes address ØØØ1 (LA 4-1) in the disk DMA controller (DDMASEL-, decoded from the higher order address lines, selects the controller), (b) when the CPU writes address Ø1ØØ in the controller, and (c) when the bus arbiter asserts DGNT-.

Sample Latched Output Equation

```
DSKDMAWR := DDMASEL * LWT * /LA4 * /LA3 *  LA2 *  LA1 +
            DSKDMAWR * /DDMASEL +
            DSKDMAWR * /LWT +
            DSKDMAWR *  LA4 +
            DSKDMAWR * /LA3 +
            DSKDMAWR *  LA2 +
            DSKDMAWR *  LA1 +
            RESET
```

In this latched output equation, the first group of inputs indicates that the CPU asserts DSKDMAWR- by writing to address ØØ11 in the disk DMA controller. The remaining groups define the DSKDMAWR- negate address, which turns out to be Ø1ØØ (in the disk DMA controller).

The negate address is Ø1ØØ, because any other situation satisfies one of the remaining groups, and the PAL asserts DSKDMAWR-. When the cpu writes Ø1ØØ (in the disk DMA controller) no equations is satisfied, as DDMASEL- is asserted, LWT- is asserted, LA4 is negated, LA3 is asserted, LA2 is negated, and LA1 is negated; the PAL negates DSKDMAWR-.

The last three bits of the address, 1ØØ, represent the last hex digit (8) of the virtual address, because unmapped virtual addresses are derived directly from their physical addresses by shifting the physical address one bit to the left and adding a zero to the least signifigant bit position.

The address decode PAL asserts DDMASEL- when the CPU writes addresses $C8ØØØØ through $C8ØØØF. Referring to the disk DMA registers in Section 4, the CPU writing $C8ØØØ8 defines a disk DMA operation, with the DMA direction being from the device to memory (disk read). In other words, DSKDMAWR- is negated (high).

AND/EXOR Gate Array PALs

The following equation, taken from PAL 219A, is an example of an AND/EXOR PAL equation.

```
/LA17 :=     FADHICTRL * /LA17          ; COUNT
        +    FADHICTRL * /LA17
       :+: /FADHICTRL * /LDØ            ; LOAD LDØ (LSB)
        +    FADHICTRL * CARRYIN        ; COUNT
```

:+: is the symbol for the EXCLUSIVE OR. Thus, LA17 is negated (low) if FADHICTRL- is asserted and /LA17 is low; unless either FADHICTRL- is negated and LDO is low, or FADHICTRL- and CARRYIN- are asserted. Also, LA17 is negated if either FADHICTRL- is negated or LA 17 is high; and either FADHICTRL- is negated and LDO is low, or FADHICTRL- is asserted and CARRYIN- is asserted.

NOTE: The first half of the equation might also have had two dissimilar parts (as the second half does), which would have made it considerably more complicated.

Hints on Decoding the PAL Listings

The function performed by a particular PAL equation is not always obvious. Use the following tips to decode the equations.

1. Outputs always go to their normal state unless their equation is satisfied, at which time they negate or assert. If an output is normally asserted, it remains asserted unless its equation is satisfied.

   Use Demorgan's theorem on any PAL equation whose output is normally asserted. For example:

   ```
   /SETRAS2 = /REFGNT * /DGNT * /FCGNT * /XGNT
   ```

   works out to:

   ```
   SETRAS2 = REFGNT + DGNT + FCGNT + XGNT
   ```

2. Whenever the equation is for a latched output, the output has two programmable states, an assert address and a negate address. The assert address is usually the first line of the equation. The negate address is obtained by using Demorgan's theorem on the remainder of the equation. (See subsection "Sample Latched Output Equation", above.)

3. When decoding equations for PALs which are part of a state machine, either the sequencer or a decoder, determine the state (number) that a group of inputs refers to, and write this number to the right of the equation. This will aid understanding the state machine sequencing. An example follows: (The state numbers derive from the decimal equivalent of the binary coded B field and are shown in bold)

```
DDRE := /B3 * /B2 *  B1 *  BØ * /LWT *  FDCS  B 3-Ø = 3
      + /B3 * /B2 *  B1 *  BØ * /LWT *  HDCS          3
      + /B3 * /B2 *  B1 *  BØ * /LWT *  DDRE          3
      + /B3 *  B2 *       /BØ * /HDDMA * /LATCH
*/DSKDMAWR                                          *Ø, 2
```

***NOTE:** States Ø and 2 are both valid because B1 can be either high or low.

PAL LISTINGS

The remainder of this appendix contains the PAL listings, each followed by a short description.

PAL 1: Main Address Decoder

This PAL decodes the upper processor address bits, and enables the appropriate processing logic for that type of processor cycle. The PAL is not used during DMA cycles.

```
PAL20L10
301A
FOR MP,17F 12/03/83

PA23 PA22 PA19 PA18 PA17 PA16 /LIO /ROMEN /AAS SUPV PGM GND
/FCHI /RSEL /MUSL /PADENH /PADENL /PREG NPC /IADR /MRSEL /UIOER
/IOSEL VCC

IF(VCC) MUSL = SUPV * /PA23 *  PA22 * /NPC * /FCHI *
                       /PA19 * /PA18 * /PA17 * /PA16

IF(VCC) UIOER  = /SUPV * /NPC *  PA23          +
                 /SUPV * /NPC * /PA23 * PA22

IF(VCC) RSEL = PA23 * /PA22 * SUPV * /NPC * AAS +
               ROMEN * SUPV * /NPC * AAS * /PA23 * /PA22

IF(VCC) IADR = PA23 * PA22 * PA19 * PA18 * PA17 * PA16 * FCHI

IF(VCC) PADENL = /NPC

IF(VCC) PADENH = /NPC * /SUPV +
                 /NPC *  PA23 +
                 /NPC * /PA22

IF(VCC) MRSEL =  /PA23 * PA22 * /NPC * SUPV * /FCHI * PA18 +
                 /PA23 * PA22 * /NPC * SUPV * /FCHI * PA17 +
                 /PA23 * PA22 * /NPC * SUPV * /FCHI * PA16

IF(VCC) IOSEL =   PA23 * PA22 * /NPC * SUPV * LIO * AAS

IF(VCC) PREG  =  /PA23 * PA22 * /NPC * /FCHI
```

PAL 2: CPU Register Decoder

This PAL generates the select signals for the on-board system registers and the system reset function.

```
PAL16L8A   PAL DESIGN SPECIFICATION
3Ø2A
FOR MP,18F 1Ø/31/83

AT2   /MPREGSEL  PA19  PA18  PA17  PA16  /UDS  /LWT  /LDS  GND
/AAS /BSRLRD /HARIKARI /NC1 /GCRWR /BSRMRD /BSRHRD /GSRCL /GSRRD
VCC

IF(VCC) GSRRD     = MPREGSEL * /PA19 * /PA18 * /PA17 *  PA16 *
/AT2 * /LWT

IF(VCC) GCRWR     = MPREGSEL*/PA19*PA18*/PA17*PA16*/AT2*LWT*UDS +
                    MPREGSEL*/PA19*PA18*/PA17*PA16*/AT2*LWT*LDS

IF(VCC) GSRCL     = MPREGSEL*/PA19*/PA18*PA17*/PA16*/AT2*LWT*UDS +
                    MPREGSEL*/PA19*/PA18*PA17*/PA16*/AT2*LWT*LDS

IF(VCC) BSRLRD    = MPREGSEL * /PA19 *  PA18 * /PA17 * /PA16 *
/AT2 * /LWT *
                    LDS

IF(VCC) BSRMRD    = MPREGSEL * /PA19 *  PA18 * /PA17 * /PA16 *
/AT2 * /LWT *
                    UDS

IF(VCC) BSRHRD    = MPREGSEL * /PA19 * /PA18 *  PA17 *  PA16 *
/AT2 * /LWT *
                    LDS

IF(VCC) HARIKARI  = MPREGSEL*/PA19*PA18*PA17*/PA16*/AT2*LWT*UDS +
                    MPREGSEL*/PA19*PA18*PA17*/PA16*/AT2*LWT*LDS
```

PAL 3: I/O Address Decoder #1

This PAL decodes some of the I/O device select addresses. It also performs a miscellaneous function to form the INTA counter load/count bit.

```
PAL16L8A   PAL DESIGN SPECIFICATION
3Ø3A
FOR MP,2ØF 9/24/83

/XDMAREQ  PA19  PA18  PA17  PA16  /IOSEL  AT5  /AS     /DSKDMAREQ
GND
/FCDMAREQ /SCSEL /NC1 /REFREQ /LIO /INTACNTRE /ICASEL /LWT
/SCTSEL VCC

IF(VCC) ICASEL    = IOSEL *  PA19 * /PA18 * /PA17 *  PA16 * /AT5 *
LWT +
                    IOSEL *  PA19 * /PA18 * /PA17 *  PA16 * /LWT

IF(VCC) SCSEL     = IOSEL * /PA19 * /PA18 *  PA17 *  PA16 * /AT5 *
LWT +
                    IOSEL * /PA19 * /PA18 *  PA17 *  PA16 * /LWT

IF(VCC) SCTSEL    = IOSEL * /PA19 *  PA18 * /PA17 * /PA16 * /AT5 *
LWT +
                    IOSEL * /PA19 *  PA18 * /PA17 * /PA16 * /LWT

IF(VCC) INTACNTRE = /AS +
                    DSKDMAREQ * /LIO +
                    FCDMAREQ  * /LIO +
                    XDMAREQ   * /LIO +
                    REFREQ     * /LIO
```

PAL 4: I/O Address Decoder #2

This PAL decodes some of the I/O device select addresses.

```
PAL16L8A   PAL DESIGN SPECIFICATION
304A
FOR MP,21F 9/24/83

/NC1  PA19   PA18   PA17   PA16   /IOSEL   /UDS   /LWT   /LDS
GND
AT5 /TIMRWR /TIMRRD /DSEL /LPSEL /XSEL1 /FCSEL /DDMASEL /FDMASEL
VCC

IF(VCC) TIMRRD   = IOSEL * /PA19 * /PA18 * /PA17 * /PA16 * UDS  *
/LWT +
                   IOSEL * /PA19 * /PA18 * /PA17 * /PA16 * LDS  *
/LWT

IF(VCC) TIMRWR   = IOSEL * /PA19 * /PA18 * /PA17 * /PA16 * /AT5 *
LWT

IF(VCC) DSEL     = IOSEL * /PA19 *  PA18 *  PA17 * /PA16 * /AT5 *
LWT +
                   IOSEL * /PA19 *  PA18 *  PA17 * /PA16 * LDS  *
/LWT +
                   IOSEL * /PA19 *  PA18 *  PA17 * /PA16 * UDS  *
/LWT

IF(VCC) FCSEL    = IOSEL * /PA19 * /PA18 * /PA17 *  PA16 * /AT5 *
LWT +
                   IOSEL * /PA19 * /PA18 * /PA17 *  PA16 * LDS  *
/LWT +
                   IOSEL * /PA19 * /PA18 * /PA17 *  PA16 * UDS  *
/LWT

IF(VCC) LPSEL    = IOSEL * /PA19 *  PA18 * /PA17 *  PA16 * /AT5 *
LWT +
                   IOSEL * /PA19 *  PA18 * /PA17 *  PA16 * LDS  *
/LWT +
                   IOSEL * /PA19 *  PA18 * /PA17 *  PA16 * UDS  *
/LWT

IF(VCC) XSEL1    = IOSEL * /PA19 *  PA18 *  PA17 *  PA16 * /AT5 *
LWT +
                   IOSEL * /PA19 *  PA18 *  PA17 *  PA16 * LDS  *
/LWT +
                   IOSEL * /PA19 *  PA18 *  PA17 *  PA16 * UDS  *
/LWT
```

```
IF(VCC) DDMASEL  = IOSEL *  PA19 * /PA18 * /PA17 * /PA16 * /AT5 *
LWT +
                   IOSEL *  PA19 * /PA18 * /PA17 * /PA16 * LDS  *
/LWT +
                   IOSEL *  PA19 * /PA18 * /PA17 * /PA16 * UDS  *
/LWT

IF(VCC) FDMASEL  = IOSEL * /PA19 * /PA18 *  PA17 * /PA16 * /AT5 *
LWT +
                   IOSEL * /PA19 * /PA18 *  PA17 * /PA16 * UDS  *
/LWT +
                   IOSEL * /PA19 * /PA18 *  PA17 * /PA16 * LDS  *
/LWT
```

DESCRIPTION

PAL 5: Bus Arbiter

This PAL controls access to the system bus. It honors the highest priority request present one clock before the end of the current bus cycle. The priority scheme is

```
     EXTERNAL  DMA      Highest
     DISK      DMA
     FAST COMM DMA
     REFRESH
     PROCESSOR CYCLES   Lowest
```

```
PAL16R8A   PAL DESIGN SPECIFICATION
305B
FOR MP,21E 10/31/83

/PCLK HOLM /AAS /FCDMAREQ /XDMAREQ /ROMEN /DSKDMAREQ /REFREQ
/RAS1 GND
/OE  /LIO  /NC  /LMEM    /REFGNT   /DGNT  /FCGNT     /XGNT   /CLR
VCC

LIO   := LIO * AAS +
         AAS * /REFREQ * /XDMAREQ * /DSKDMAREQ * /FCDMAREQ *
               XGNT * /XDMAREQ  +
         AAS * /REFREQ * /XDMAREQ * /DSKDMAREQ * /FCDMAREQ *
               DGNT * /DSKDMAREQ  +
         AAS * /REFREQ * /XDMAREQ * /DSKDMAREQ * /FCDMAREQ *
               FCGNT * /FCDMAREQ  +
         AAS * /REFREQ * /XDMAREQ * /DSKDMAREQ * /FCDMAREQ *
               REFGNT * /REFREQ  +
         AAS * /REFREQ * /XDMAREQ * /DSKDMAREQ * /FCDMAREQ *
               LMEM * /RAS1  +
         AAS * /REFREQ * /XDMAREQ * /DSKDMAREQ * /FCDMAREQ *
              /XGNT * /DGNT * /FCGNT * /REFGNT * /RAS1 * /LIO


XGNT  := XGNT * XDMAREQ +
         XDMAREQ * DGNT              +
         XDMAREQ * FCGNT * /FCDMAREQ +
         XDMAREQ * REFGNT * /REFREQ  +
         XDMAREQ * LMEM * /RAS1 * /AAS +
         XDMAREQ * LIO * /AAS +
         XDMAREQ * /XGNT * /DGNT * /FCGNT * /REFGNT * /RAS1 *
/LIO
```

```
 DGNT   := DGNT * DSKDMAREQ +
           DSKDMAREQ * /XDMAREQ * XGNT  * /XDMAREQ  +
           DSKDMAREQ * /XDMAREQ * FCGNT * /FCDMAREQ +
           DSKDMAREQ * /XDMAREQ * REFGNT * /REFREQ  +
           DSKDMAREQ * /XDMAREQ * LMEM * /RAS1 * /AAS +
           DSKDMAREQ * /XDMAREQ * LIO * /AAS +
           DSKDMAREQ * /XDMAREQ
*/XGNT*/DGNT*/FCGNT*/REFGNT*/RAS1*/LIO

 FCGNT := FCGNT * FCDMAREQ  +
           FCDMAREQ * /DSKDMAREQ * /XDMAREQ * XGNT * /XDMAREQ +
           FCDMAREQ * /DSKDMAREQ * /XDMAREQ * DGNT             +
           FCDMAREQ * /DSKDMAREQ * /XDMAREQ * REFGNT * /REFREQ  +
           FCDMAREQ * /DSKDMAREQ * /XDMAREQ * LMEM * /RAS1 * /AAS
+
           FCDMAREQ * /DSKDMAREQ * /XDMAREQ * LIO * /AAS +
           FCDMAREQ * /DSKDMAREQ * /XDMAREQ * /XGNT * /DGNT *
                      /FCGNT * /REFGNT * /RAS1 * /LIO

REFGNT := REFGNT * REFREQ +
           REFREQ * /FCDMAREQ * /DSKDMAREQ * /XDMAREQ * DGNT *
/DSKDMAREQ +
           REFREQ * /FCDMAREQ * /DSKDMAREQ * /XDMAREQ * XGNT *
/XDMAREQ +
           REFREQ * /FCDMAREQ * /DSKDMAREQ * /XDMAREQ * FCGNT *
/FCDMAREQ +
           REFREQ * /FCDMAREQ * /DSKDMAREQ * /XDMAREQ * REFGNT *
/REFREQ  +
           REFREQ * /FCDMAREQ * /DSKDMAREQ * /XDMAREQ * LMEM *
/RAS1 * /AAS+
           REFREQ * /FCDMAREQ * /DSKDMAREQ * /XDMAREQ * LIO * /AAS
+
           REFREQ * /FCDMAREQ * /DSKDMAREQ * /XDMAREQ * /XGNT *
/DGNT *
                    /FCGNT * /REFGNT * /RAS1 * /LIO


LMEM   := RAS1 * /ROMEN +
          /REFREQ*/FCDMAREQ*/DSKDMAREQ*/XDMAREQ*/AAS*/ROMEN*/HOLM

CLR    := LMEM   * /RAS1        +
          REFGNT * /REFREQ      +
          DGNT   * /DSKDMAREQ   +
          XGNT   * /XDMAREQ     +
          FCGNT  * /FCDMAREQ    +
          LIO    * /AAS         +
          ROMEN  * /LIO         +
          /LMEM * /REFGNT * /DGNT * /XGNT * /FCGNT * /LIO *
/ROMEN
```

PAL 6: Disk Control Register

This PAL routes control signals to the disk bus interface unit, the disk controllers, and the disk DMA controller. The CPU sets and reset the outputs by writing to the appropriate addresses. After a system reset, the state of this PAL is:

- FDRESET- is asserted. (The floppy disk is reset.)
- HDRESET- is asserted. (This hard is reset.)
- FDMOTORON- is negated. (The floppy motor is off.)
- HDDMA- is asserted. (The hard disk controller is selected.)
- DSKDMAEN- is negated. (The disk bus interface unit is disabled.)

```
PAL16R6A   PAL DESIGN SPECIFICATION
3Ø6A
FOR MP,23L 9/26/83

/PCLK  LA5  LA1  LA2  LA3  LA4  /RESET  /DSABLDMA  /LWT GND
/OE /DSEL /DSKDMAEN /FDRESET /FDMOTORON /HDDMA /FDDBLDEN /HDRESET
/BIURESET VCC

FDRESET := DSEL * LA5 * /LA4 * /LA3 * /LA2 * /LA1 * LWT +
           FDRESET * /LA5 +
           FDRESET *  LA4 +
           FDRESET *  LA3 +
           FDRESET *  LA2 +
           FDRESET * /LA1 +
           FDRESET * /DSEL +
           RESET

HDRESET := DSEL * LA5 * /LA4 * /LA3 * LA2 * /LA1 * LWT +
           HDRESET * /LA5 +
           HDRESET *  LA4 +
           HDRESET *  LA3 +
           HDRESET * /LA2 +
           HDRESET * /LA1 +
           HDRESET * /DSEL +
           RESET
```

```
FDMOTORON := DSEL * LA5 * /LA4 * LA3 * /LA2 * /LA1 * LWT * /RESET
+
             FDMOTORON * /LA5 * /RESET +
             FDMOTORON *  LA4 * /RESET +
             FDMOTORON * /LA3 * /RESET +
             FDMOTORON *  LA2 * /RESET +
             FDMOTORON * /LA1 * /RESET +
             FDMOTORON * /DSEL * /RESET

HDDMA := DSEL * LA5 * /LA4 * LA3 * LA2 * /LA1 * LWT +
         HDDMA * /LA5 +
         HDDMA *  LA4 +
         HDDMA * /LA3 +
         HDDMA * /LA2 +
         HDDMA * /LA1 +
         HDDMA * /DSEL +
         RESET


DSKDMAEN := /DSABLDMA * DSEL * LA5 * /LA4 * LA3 *  LA2 * LWT *
/RESET +
            /DSABLDMA * DSKDMAEN * /LA5 * /RESET +
            /DSABLDMA * DSKDMAEN * /LA4 * /RESET + ; TURN ON =
C6ØØ2C
            /DSABLDMA * DSKDMAEN *  LA3 * /RESET + ;   OR C6ØØ2E
            /DSABLDMA * DSKDMAEN *  LA2 * /RESET + ; TURN OFF =
C6ØØ3Ø
            /DSABLDMA * DSKDMAEN *  LA1 * /RESET +
            /DSABLDMA * DSKDMAEN * /DSEL * /RESET

FDDBLDEN := DSEL *  LA5 *  LA4 * /LA3 *  LA2 * /LA1 * LWT
          + FDDBLDEN * /LA5
          + FDDBLDEN * /LA4
          + FDDBLDEN *  LA3
          + FDDBLDEN *  LA2
          + FDDBLDEN * /LA1
          + FDDBLDEN * /DSEL
          + RESET


IF (VCC) BIURESET = DSEL *  LA5 *  LA4 * /LA3 *  LA2 *  LA1 *
LWT
                  + RESET
```

PAL 7: Disk Byte Decoder #2

This PAL generates the chip selects as well as the read and write enables for the hard disk and the floppy disk controllers. The signal LATCH is used internally to determine when to deassert the enables. This PAL also generates DBØOE.

```
PAL16R6A   PAL DESIGN SPECIFICATION
3Ø7A
FOR MP,24K 9/27/83

PCLK /B3 /LWT  /B2    LA4  /B1   /BØ   /DSKDMAWR /HDDMA GND
/OE  AT  /DDRE /FDAE /HDCS /FDCS /DDWE /LATCH    /DBØOE VCC

HDCS :=  /B3 * /B2 *  B1 *  BØ * /LA4
       + HDCS * DDWE
       + HDCS * DDRE

FDCS := /B3 * /B2 *  B1 *  BØ *  LA4
      + /B3 * /B2 *  B1 * /BØ * /HDDMA
      + /B3 *  B2 * /B1 * /BØ * /HDDMA
      + /B3 *  B2 * /B1 *  BØ * /HDDMA
      + /B3 *  B2 *  B1 * /BØ * /HDDMA
      +  FDCS * DDRE * LATCH

DDRE := /B3 * /B2 *  B1 *  BØ * /LWT *  FDCS * /AT
      + /B3 * /B2 *  B1 *  BØ * /LWT *  HDCS
      + /B3 * /B2 *  B1 *  BØ * /LWT *  DDRE * /AT
      + /B3 *  B2 *       /BØ * /HDDMA * /LATCH * /DSKDMAWR

DDWE := /B3 * /B2 *  B1 *  BØ *  LWT *  FDCS * /AT
      + /B3 * /B2 *  B1 *  BØ *  LWT *  DDWE * /AT
      + /B3 * /B2 *  B1 *  BØ *  LWT *  HDCS * /LATCH
      + /B3 *  B2 *       /BØ * /HDDMA * /LATCH * DSKDMAWR

LATCH := /B3 * /B2 *  B1 *  BØ * DDRE
       + /B3 * /B2 *  B1 *  BØ * DDWE
       + /B3 * /B2 *  B1 *  BØ * LATCH
       + /B3 *  B2 *       /BØ * DDRE
       + /B3 *  B2 *       /BØ * DDWE
       + /B3 *  B2 *       /BØ * LATCH

FDAE := /B3 * /B2 *  B1 *  BØ *  LA4
      +  FDCS * DDRE * LATCH


IF(VCC) DBØOE = /B3 * /B2 * /B1 *  BØ * /DSKDMAWR
              + /B3 * /B2 *  B1 *  BØ
              + /B3 *  B2 * /B1 * /BØ *  DSKDMAWR
              + DBØOE * DDWE * DSKDMAWR
```

PAL 8: Disk Byte State Sequencer

This PAL sequences the disk byte state machine through its different states.

```
PAL16R6A   PAL DESIGN SPECIFICATION
308A
FOR MP,23K 10/4/83

/PCLK /SYNCDDREQ /BIURESET DISKTC /BACK /BRESET LA5 /DSKDMAWR
/DSEL     GND
/OE   /DSKDMAEN  /BREQ    /BØ     /B1   /B3    /B2   DBINT
/DSABLDMA VCC


B3 := /B3 * /B2 * /B1 * /BØ * SYNCDDREQ * /BIURESET
    + /B3 * /B2 * /B1 *  BØ * SYNCDDREQ
    +  B3 * /B2 * /B1 * /BØ * /BIURESET

B2 := /B3 * /B2 *  B1 * /BØ * /BIURESET * /DSEL *  SYNCDDREQ *
/BRESET
    + /B3 * /B2 *  B1 * /BØ * /BIURESET *  LA5  *  SYNCDDREQ *
/BRESET
    + /B3 *  B2 * /B1 * /BØ * /BIURESET *  SYNCDDREQ * /BRESET
    + /B3 *  B2 * /B1 *  BØ * /BIURESET *  /BRESET
    + /B3 *  B2 *  B1 * /BØ * /BIURESET *  /BRESET

B1 := /B3 * /B2 * /B1 * /BØ * /BIURESET *  DSKDMAEN * /DSKDMAWR *
/BRESET
    + /B3 * /B2 *       /BØ * /BIURESET *  DSEL * /LA5  * /BRESET
    + /B3 * /B2 * /B1 *  BØ * /BIURESET *  BACK * /SYNCDDREQ *
/BRESET
    + /B3 * /B2 *  B1 * /BØ * /BIURESET * /DSEL * /BRESET
    + /B3 * /B2 *  B1 * /BØ * /BIURESET *  LA5  * /BRESET
    + /B3 * /B2 *  B1 *  BØ * /BIURESET *  DSEL * /LA5  * /BRESET
    + /B3 * /B2 *  B1 *  BØ * /BIURESET *  DSKDMAEN * /DSEL *
/BRESET
    + /B3 *  B2 *  B1 * /BØ * /BIURESET *  SYNCDDREQ  * /BRESET

BØ := /B3 * /B2 * /B1 * /BØ * /BIURESET *  DSKDMAEN *  DSKDMAWR *
/BRESET
    + /B3 * /B2 * /B1 * /BØ * /BIURESET *  DSEL  * /LA5   *
/BRESET
    + /B3 * /B2 * /B1 *  BØ * /BIURESET * /BACK  * /BRESET
    + /B3 * /B2 *  B1 *       /BIURESET *  DSEL  * /LA5   *
/BRESET
    + /B3 *  B2 * /B1 *       /BIURESET * /SYNCDDREQ * /BRESET
    + /B3 *  B2 *  B1 * /BØ * /BIURESET * /SYNCDDREQ * /BRESET

BREQ := /B3 *  B2 * /B1 * /BØ * /BIURESET * /SYNCDDREQ
      + /B3 * /B2 * /B1 *  BØ * /BIURESET * /BACK
```

```
IF (VCC) DSABLDMA = /B3 * /B2 * /B1 *  BØ * /BIURESET * SYNCDDREQ
                  + BRESET

/DBINT  := /B3
```

PAL 9: Disk Byte Decoder #1

This PAL generates all the signals (except DBØOE) used to control the two LS646 disk byte tranciever/registers.

```
PAL16L8A   PAL DESIGN SPECIFICATION
309A
FOR MP,21K 9/27/83

/B3  /B2    /B1      /BØ     /HDDMA     NC2      /LWT
/DSKDMAWR /HDWE   GND
/NC4 /DB1CLKB /DB1OE /DB1DRVA /DBØ1CLKA /DBØCLKB /DBØBSEL
/DBØASEL /DBØDRVA VCC

IF (VCC) DB1OE = /B3 * /B2 * /B1 *  BØ * /DSKDMAWR
               + /B3 *  B2 *  B1 * /BØ *  DSKDMAWR

IF (VCC) DB1CLKB = /B3 *  B2 *  B1 * /BØ * /DSKDMAWR *  HDDMA *
HDWE
                 + /B3 *  B2 *  B1 * /BØ * /DSKDMAWR * /HDDMA

IF (VCC) DBØASEL = /B3 * /B2 *  B1 *  BØ *  LWT

IF (VCC) DBØDRVA = /B3 * /B2 * /B1 *  BØ * /DSKDMAWR
                 + /B3 * /B2 *  B1 *  BØ * /LWT

IF (VCC) DBØ1CLKA = /B3 * /B2 * /B1 *  BØ *  DSKDMAWR

IF (VCC) DB1DRVA =  /B3 * /B2 * /B1 *  BØ * /DSKDMAWR

IF (VCC) DBØBSEL =  /B3 * /B2 *  B1 *  BØ * /LWT

IF (VCC) DBØCLKB =  /B3 *  B2 * /B1 * /BØ * /DSKDMAWR *  HDDMA *
HDWE
                  + /B3 *  B2 * /B1 * /BØ * /DSKDMAWR * /HDDMA
```

PAL 1Ø: Disk Word Decoder

This PAL generates all the signals used to control the two LS646 disk word tranciever/registers.

```
PAL16L8A   PAL DESIGN SPECIFICATION
310A
FOR MP,22K 9/24/83

NC1 /W3     /W2     /W1       /WØ      NC2     /LWT      /DGNT
/DSKDMAWR GND NC3 /WBCLKB /WB1OE /WBØBSEL /WBØOE /WBDRVA /WBØASEL
/WBCLKA /NC4 VCC

IF (VCC) WBCLKA = /W3 * /W2 *  W1 * /WØ * DSKDMAWR

IF (VCC) WBDRVA =  DGNT * /DSKDMAWR
                 + /W3 *  W2 *  W1 * /WØ * /LWT

IF (VCC) WB1OE  =  DGNT * /DSKDMAWR
                 + /W3 *  W2 * /W1 *  WØ *  DSKDMAWR

IF (VCC) WBCLKB = /W3 *  W2 * /W1 *  WØ * /DSKDMAWR

IF (VCC) WBØOE  =  DGNT * /DSKDMAWR
                 + /W3 *  W2 *  W1 * /WØ
                 + /W3 *  W2 * /W1 *  WØ *  DSKDMAWR

IF (VCC) WBØBSEL =  /W3 *  W2 *  W1 * /WØ * /LWT

IF (VCC) WBØASEL =  /W3 *  W2 *  W1 * /WØ *  LWT
```

PAL 11: Disk Interrupt Controller

This PAL generates disk control interrupt for overflow, underrun, end of transfer, or disk controller error conditions. In doing this, the PAL also prevents the disk controller that is not selected during a DMA from issuing an interrupt.

```
PAL16L8A   PAL DESIGN SPECIFICATION
311A
FOR MP,23M 9/29/83

/HDBCS /HDWE FDDRQ FDINTRQ HDINTRQ HDRWC /HDSEL3 /HITECHSEL
/HDDMA GND
/DSKDMAEN /NC1 /NC2 /NC3 /DSKDRQ RWC_HDSEL3 FDINTREQ /HDRE
HDINTREQ VCC

IF (VCC) /FDINTREQ = DSKDMAEN * HDDMA +
                     /FDINTRQ

IF (VCC) /HDINTREQ = DSKDMAEN * /HDDMA +
                     /HDINTRQ

IF (VCC) /RWC_HDSEL3 = HITECHSEL * /HDSEL3 +
                       /HITECHSEL * /HDRWC

IF (VCC) /DSKDRQ = FDDRQ
                 + HDBCS * HDRE
                 + HDBCS * HDWE
```

PAL 12: Disk DMA Upper Address Counter

During a disk DMA operation, this PAL maintains the five most significant bits of the DMA address and also stores the proper sense of LWT-.

```
PAL2ØX8A   PAL DESIGN SPECIFICATION
312A
FOR MP,22F 9/24/83

DADHICLK LD4 LD3 LD2 LD1 LDØ NC1 LA1 NC2 NC3  DADHICTRL GND
/OE /CARRYIN NC4 /LWT NC5 LA21 LA2Ø LA19 LA18 LA17 NC6 VCC

/LA17 :=     DADHICTRL * /LA17           ; COUNT
        +    DADHICTRL * /LA17
       :+: /DADHICTRL * /LDØ             ; LOAD LDØ (LSB)
        +    DADHICTRL * CARRYIN         ; COUNT

/LA18 :=     DADHICTRL * /LA18           ; COUNT
        +    DADHICTRL * /LA18
       :+: /DADHICTRL * /LD1             ; LOAD LD1
        +    DADHICTRL * CARRYIN * LA17  ; COUNT

/LA19 :=     DADHICTRL * /LA19                          ; COUNT
        +    DADHICTRL * /LA19
       :+: /DADHICTRL * /LD2                            ; LOAD LD2
        +    DADHICTRL * CARRYIN * LA17 * LA18          ; COUNT

/LA2Ø :=     DADHICTRL * /LA2Ø           ; COUNT
        +    DADHICTRL * /LA2Ø
       :+: /DADHICTRL * /LD3             ; LOAD LD3
        +    DADHICTRL * CARRYIN * LA17 * LA18 * LA19    ; COUNT

/LA21 :=     DADHICTRL * /LA21           ; COUNT
        +    DADHICTRL * /LA21
       :+: /DADHICTRL * /LD4             ; LOAD LD4 (MSB)
        +    DADHICTRL * CARRYIN * LA17 * LA18 * LA19 * LA2Ø
;COUNT

LWT    :=    DADHICTRL * LWT             ; HOLD LWT
        +    DADHICTRL * LWT
       :+: /DADHICTRL * /LA1             ; LOAD LWT
        +  /DADHICTRL * /LA1
```

PAL 13: Disk DMA Controller

This PAL generates the control signals for loading the disk DMA word address registers and disk DMA word count register. The PAL also generates the clocks for these registers.

```
PAL16R4A   PAL DESIGN SPECIFICATION
313A
FOR MP,23H 9/24/83

/PCLK LA1 LA2 LA3 LA4 /DGNT /LWT /DDMASEL /RESET GND
/OE /DCNTOE DADHICLK /DADHICTRL /DADLOCTRL /DCNTCTRL /DSKDMAWR
DADLOCLK DCNTCLK VCC

IF (VCC) /DADLOCLK = DDMASEL * LWT * /LA4 * /LA3 * /LA2 * LA1 +
                     DGNT

IF (VCC) /DADHICLK = DDMASEL * LWT * /LA4 * /LA3 *  LA2 *  LA1 +
                     DDMASEL * LWT * /LA4 *  LA3 * /LA2 * /LA1 +
                     DGNT

DSKDMAWR := DDMASEL * LWT * /LA4 * /LA3 *  LA2 *  LA1 +
            DSKDMAWR * /DDMASEL +
            DSKDMAWR * /LWT +
            DSKDMAWR *  LA4 +
            DSKDMAWR * /LA3 +
            DSKDMAWR *  LA2 +
            DSKDMAWR *  LA1 +
            RESET

DADLOCTRL := DDMASEL * LWT * /LA4 * /LA3 * /LA2 *  LA1

DADHICTRL := DDMASEL * LWT * /LA4 * /LA3 *  LA2 *  LA1 +
             DDMASEL * LWT * /LA4 *  LA3 * /LA2 * /LA1

DCNTCTRL  := DDMASEL * LWT * /LA4 * /LA3 * /LA2 * /LA1

IF (VCC) /DCNTCLK = DDMASEL * LWT * /LA4 * /LA3 * /LA2 * /LA1 +
                    DGNT

IF (VCC) DCNTOE    = DDMASEL * /LWT * /LA4 * /LA3 * /LA2 * /LA1
```

PAL 14: Fast Communications Control Register

This PAL routes control signals to the fast communications bus interface unit, the port controller, and the fast communications DMA controller. The CPU sets and resets the outputs by writing to the appropriate addresses.

```
PAL16R6A   PAL DESIGN SPECIFICATION
314A
FOR MP,1ØH 9/24/83

/PCLK LA5 LA4 LA3 LA2 LA1 /FCSEL /RESET /LWT GND
/OE /FCTCINT FCRXE /FCCLKSEL /FCCLKOE /FCDMAEN FCTXE FCRESET
/DSTXE VCC

FCDMAEN := FCSEL * LA5 * /LA4 * /LA3 * LA2 * /LA1 * LWT * /RESET
* /FCTCINT
         + FCDMAEN * /FCSEL * /RESET * /FCTCINT
         + FCDMAEN * /LA5 * /RESET * /FCTCINT
         + FCDMAEN *  LA4 * /RESET * /FCTCINT
         + FCDMAEN *  LA3 * /RESET * /FCTCINT
         + FCDMAEN * /LA2 * /RESET * /FCTCINT
         + FCDMAEN * /LA1 * /RESET * /FCTCINT
         + FCDMAEN * /LWT * /RESET * /FCTCINT

FCCLKOE  := FCSEL *  LA5 *  LA4 * /LA3 * /LA2 * /LA1 * LWT *
/RESET
         + FCCLKOE * /FCSEL * /RESET
         + FCCLKOE * /LA5 * /RESET
         + FCCLKOE * /LA4 * /RESET
         + FCCLKOE *  LA3 * /RESET
         + FCCLKOE *  LA2 * /RESET
         + FCCLKOE * /LA1 * /RESET
         + FCCLKOE * /LWT * /RESET

/FCRXE  := FCSEL *  LA5 * /LA4 * /LA3 * /LA2 *  LA1 * LWT
         + /FCRXE * /FCSEL
         + /FCRXE * /LA5
         + /FCRXE *  LA4
         + /FCRXE *  LA3
         + /FCRXE *  LA2
         + /FCRXE *  LA1
         + RESET

/FCTXE  := /FCTXE * /FCSEL
         + /FCTXE * /LA5
         + /FCTXE *  LA4
         + /FCTXE * /LA3
         + /FCTXE * /LA2
         + /FCTXE *  LA1
         + FCRESET
         + DSTXE
```

```
/FCRESET := /RESET * /FCSEL
          + /RESET * /LA5
          + /RESET * /LA4
          + /RESET *  LA3
          + /RESET * /LA2
          + /RESET *  LA1
          + /RESET * /LWT

FCCLKSEL := FCSEL * LA5 * /LA4 * LA3 * /LA2 * /LA1 * LWT
          + FCCLKSEL * /FCSEL
          + FCCLKSEL * /LA5
          + FCCLKSEL *  LA4
          + FCCLKSEL * /LA3
          + FCCLKSEL *  LA2
          + FCCLKSEL * /LA1
          + RESET
```

PAL 15: Fast Communications Device Decoder

This PAL decodes the state indicators from the fast communications word/byte state sequencer and generates the chip enable, the write, and the data bus enable signals for the 2652. The signal FCLATCH holds the DBEN until after the enables have gone away.

```
PAL16R4A   PAL DESIGN SPECIFICATION
315A
FOR MP,9F  9/29/83

/PCLK /FCDMAWR /LWT /SYNCFCREQ /S3 /S2 /S1 /SØ /FCCLKSEL GND
/OE 2MHZ 3Ø7KHZ FCCE FCWRT /FCDBEN /FCLATCH /NC1 /FCTXCLK VCC

/FCCE  := /S3 * /S2 * /S1 * /SØ
        + /S3 * /S2 * /S1 *  SØ
        + /S3 *  S2 * /S1 *  SØ
        +  S3 *  S2 * /S1 * /SØ

FCDBEN := /S3 * /S2 *  S1 * /SØ * FCLATCH
        + /S3 *  S2 * /S1 * /SØ * SYNCFCREQ
        +        S2 *  S1 * /SØ
        +  S3 * /S2 * /S1 *       SYNCFCREQ
        +  S3 * /S2 *  S1 * /SØ * SYNCFCREQ
        +  S3 * /S2 *  S1 *  SØ
        +  S3 *  S2 * /S1 *  SØ

FCLATCH := /S3 * /S2 *  S1 * /SØ

/FCWRT := /S3 * /S2 *  S1 * /SØ * /LWT
        +  S3 * /FCDMAWR
        +  S2 * /FCDMAWR
        + /S1 * /FCDMAWR
        +  SØ * /FCDMAWR

IF (VCC) FCTXCLK =  FCCLKSEL * 3Ø7KHZ
                 + /FCCLKSEL * 2MHZ
```

PAL 16: Fast Communcations State Sequencer

This PAL provides the main control to the disk bus interface unit during DMA and CPU register transfers.

```
PAL16R6A   PAL DESIGN SPECIFICATION
316A
FOR MP,1ØJ 11/12/83

/PCLK FCRESET FCTC /FCDMAEN /FCDMAWR /FCGNT LA5 /SYNCFCREQ /FCSEL
GND
/OE FCBYTE /S3 /S2 /S1 /SØ /TCINT /FCDMAREQ /CLRFCREQ VCC

IF (VCC) /FCBYTE = FCSEL
                 + FCDMAWR

TCINT := /S3 *  S2 * /S1 * /SØ * /FCRESET * /FCDMAWR * FCTC
       +  S3 *  S2 * /S1 *  SØ * /FCRESET
       +  TCINT * /FCRESET

S3    := /S3 * /S2 * /S1 *  SØ * /FCRESET *  FCGNT
       + /S3 *  S2 *  S1 * /SØ * /FCRESET
       +  S3 * /S2 *                /FCRESET *  FCDMAEN
       +  S3 *  S2 *  S1 * /SØ * /FCRESET *  FCTC

S2    := /S3 * /S2 * /S1 * /SØ * /FCRESET *  FCDMAEN * /FCDMAWR
       + /S3 * /S2 * /S1 *  SØ * /FCRESET *  FCGNT
       + /S3 * /S2 *  S1 *  SØ * /FCRESET
       + /S3 *  S2 * /S1 * /SØ * /FCRESET *  FCDMAEN
       +  S3 * /S2 *       /SØ * /FCRESET *  SYNCFCREQ * FCDMAEN
       +  S3 *  S2 * /S1 *  SØ * /FCRESET

S1    := /S3 * /S2 * /S1 * /SØ * /FCRESET *  FCSEL * /LA5
       + /S3 * /S2 *  S1 * /SØ * /FCRESET *  FCSEL
       + /S3 *  S2 * /S1 * /SØ * /FCRESET *  SYNCFCREQ * FCDMAEN
       +  S3 * /S2 * /S1 *       /FCRESET *  SYNCFCREQ * FCDMAEN
       +  S3 * /S2 *  S1 * /SØ * /FCRESET * /SYNCFCREQ * FCDMAEN
       +  S3 * /S2 *  S1 *  SØ * /FCRESET
       +  S3 *  S2 * /S1 * /SØ * /FCRESET

SØ    := /S3 * /S2 * /S1 * /SØ * /FCRESET *  FCDMAWR *  FCDMAEN
       + /S3 * /S2 * /S1 *  SØ * /FCRESET * /FCGNT
       + /S3 *  S2 * /S1 * /SØ * /FCRESET * /FCDMAWR *  FCTC  *
FCDMAEN
       +  S3 * /S2 * /S1 * /SØ * /FCRESET * /FCDMAEN
       +  S3 * /S2 * /S1 *  SØ * /FCRESET *  FCDMAEN
       +  S3 * /S2 *  S1 * /SØ * /FCRESET *  SYNCFCREQ * FCDMAEN
       +  S3 *  S2 *                /FCRESET
```

```
FCDMAREQ := /S3 * /S2 * /S1 * /SØ * /FCRESET *  FCDMAEN *
FCDMAWR
          + /S3 * /S2 * /S1 *  SØ * /FCRESET
          +  S3 *  S2 * /S1 * /SØ * /FCRESET

IF (VCC) CLRFCREQ = /S3 * /S2 * /S1 * /SØ * /FCRESET *  FCDMAEN *
/FCDMAWR
                  + /S3 *  S2 *  S1 * /SØ * /FCRESET * SYNCFCREQ
                  +  S3 * /S2 *  S1 *  SØ * /FCRESET * SYNCFCREQ
                  +  S3 *  S2 * /S1 *  SØ * /FCRESET * SYNCFCREQ
                  +  S3 *  S2 *  S1 * /SØ * /FCRESET * SYNCFCREQ

DESCRIPTION
```

PAL 17: Fast Communications Buffer Control

This PAL provides the control signals for the DMA word buffer between main memory and the RS422 controller.

```
PAL16R4A   PAL DESIGN SPECIFICATION
317A
FOR MP,9H  11/9/83

/PCLK /S3 /S2 /S1 /SØ /LWT /FCDMAWR /FCGNT /DBEN GND
/OE /CB1OE /CBØOE /CB1DRVA /CB1CLKB /CBØDRVA /CBØCLKB /CBØBSEL
/CBØASEL VCC

IF (VCC) CB1OE =   /S3 *  S2 *       /SØ *  FCDMAWR
                 +  S3 * /S2 * /S1 * /SØ *  FCDMAWR *  DBEN
                 +  S3 *       /S1 *  SØ *  FCDMAWR
                 +  S3 * /S2 *  S1 *        FCDMAWR
                 +  FCGNT * /FCDMAWR


IF (VCC) CBØOE   = /S3 * /S2 *  S1 * /SØ
                 +  S3 * /S2 * /S1 * /SØ *  FCDMAWR * /DBEN
                 +  S3 *  S2 *  S1 * /SØ *  FCDMAWR
                 +       /S2 * /S1 *  SØ *  FCDMAWR *  DBEN
                 +  FCGNT * /FCDMAWR


CBØCLKB :=  S3 *  S2 *  S1 * /SØ * /FCDMAWR

CB1CLKB := /S3 *  S2 *  S1 * /SØ * /FCDMAWR

CBØDRVA := /S3 * /S2 *  S1 * /SØ * /LWT
         +  FCGNT * /FCDMAWR

CB1DRVA :=  FCGNT * /FCDMAWR

IF (VCC) CBØBSEL = /S3 * /S2 *  S1 * /SØ

IF (VCC) CBØASEL = /S3 * /S2 *  S1 * /SØ
```

PAL 18: Fast Communications Device DMA Control

This PAL outputs control signals to the fast communications bus interface unit.

```
PAL16R6A   PAL DESIGN SPECIFICATION
318A
FOR MP,9J  1Ø/3/83

/PCLK /DBEN NC1 /FCDMAWR /LWT /SØ /S1 /S2 /S3 GND
/OE /FCASEL FCD11 FCD1Ø FCD9 FCD8 FCADR1 /DSTXE /CBCLKA VCC

IF (VCC) FCASEL = S3
                + S2
                + SØ

/FCD8 := VCC

/FCD9 := /S3
       +  S2
       + /S1 *  DBEN
       + /SØ * /DBEN

/FCD1Ø := VCC

/FCD11 := VCC

/FCADR1 := /FCDMAWR

DSTXE := S3 * /S2 *  S1 * /SØ

IF (VCC) CBCLKA = /S3 * /S2 *  S1 *  SØ  * FCDMAWR
```

PAL 19: Fast Communications DMA Upper Address

During a fast communations DMA operation, this PAL maintains the five most significant bits of the DMA address and also stores the proper sense of LWT-.

```
PAL20X8A   PAL DESIGN SPECIFICATION
319A
FOR MP,23F 9/29/83

FADHICLK LD4 LD3 LD2 LD1 LDØ NC1 LA1 NC2 NC3  FADHICTRL GND
/OE /CARRYIN NC4 NC5 /LWT LA21 LA2Ø LA19 LA18 LA17 NC6 VCC

/LA17 :=     FADHICTRL * /LA17          ; COUNT
        +    FADHICTRL * /LA17
       :+: /FADHICTRL * /LDØ            ; LOAD LDØ (LSB)
        +    FADHICTRL * CARRYIN        ; COUNT

/LA18 :=     FADHICTRL * /LA18          ; COUNT
        +    FADHICTRL * /LA18
       :+: /FADHICTRL * /LD1            ; LOAD LD1
        +    FADHICTRL * CARRYIN * LA17 ; COUNT

/LA19 :=     FADHICTRL * /LA19          ; COUNT
        +    FADHICTRL * /LA19
       :+: /FADHICTRL * /LD2            ; LOAD LD2
        +    FADHICTRL * CARRYIN * LA17 * LA18          ; COUNT

/LA2Ø :=     FADHICTRL * /LA2Ø          ; COUNT
        +    FADHICTRL * /LA2Ø
       :+: /FADHICTRL * /LD3            ; LOAD LD3
        +    FADHICTRL * CARRYIN * LA17 * LA18 * LA19    ; COUNT

/LA21 :=     FADHICTRL * /LA21          ; COUNT
        +    FADHICTRL * /LA21
       :+: /FADHICTRL * /LD4            ; LOAD LD4 (MSB)
        +    FADHICTRL * CARRYIN * LA17 * LA18 * LA19 * LA2Ø
;COUNT

LWT   :=     FADHICTRL * LWT            ; HOLD LWT
        +    FADHICTRL * LWT
       :+: /FADHICTRL * /LA1            ; LOAD LWT
        +  /FADHICTRL * /LA1
```

PAL 20: Fast Communications DMA Controller

This PAL generates the control signals for loading the disk DMA address registers and disk DMA counter. The PAL generates the DMA clocks to these registers.

```
PAL16R4A   PAL DESIGN SPECIFICATION
320A
FOR MP,26F 9/29/83

/PCLK LA1 LA2 LA3 LA4 /FDMASEL /LWT /RESET /FCGNT GND
/OE FADHICLK FADLOCLK /FADHICTRL /FCDMAWR /FADLOCTRL /FCNTCTRL
FCNTCLK /FCNTOE VCC

IF (VCC) /FADLOCLK = FDMASEL * LWT * /LA4 * /LA3 * /LA2 * LA1 +
                     FCGNT

IF (VCC) /FADHICLK = FDMASEL * LWT * /LA4 * /LA3 *  LA2 *  LA1 +
                     FDMASEL * LWT * /LA4 *  LA3 * /LA2 * /LA1 +
                     FCGNT

FCDMAWR := FDMASEL * LWT * /LA4 * /LA3 *  LA2 *  LA1 +
           FCDMAWR * /FDMASEL +
           FCDMAWR * /LWT +
           FCDMAWR *  LA4 +
           FCDMAWR * /LA3 +
           FCDMAWR *  LA2 +
           FCDMAWR *  LA1 +
           RESET

FADLOCTRL := FDMASEL * LWT * /LA4 * /LA3 * /LA2 *  LA1

FADHICTRL := FDMASEL * LWT * /LA4 * /LA3 *  LA2 *  LA1 +
             FDMASEL * LWT * /LA4 *  LA3 * /LA2 * /LA1

FCNTCTRL  := FDMASEL * LWT * /LA4 * /LA3 * /LA2 * /LA1

IF (VCC) /FCNTCLK  = FDMASEL * LWT * /LA4 * /LA3 * /LA2 * /LA1 +
                     FCGNT

IF (VCC) FCNTOE    = FDMASEL * /LWT * /LA4 * /LA3 * /LA2 * /LA1
```

PAL 21: Hard Disk Drive/Head Select

This PAL controls the use of newer technology disk drives.

```
PAL20X8A   PAL DESIGN SPECIFICATION
321A
FOR MP,20J 11/1/83

/DDWE DD6 DD5 DD4 DD3 DD2 DD1 DD0 LA1 LA2 LA3 GND
/OE /HDCS /DDRIVE1 /DDRIVE0 NC1 /HTSEL /HDSEL3 /HDSEL2 /HDSEL1
/HDSEL0
NC2 VCC

DDRIVE0 := HDCS *  LA3 *  LA2 * /LA1 *  DD4 *          DDRIVE0   ;
2 OR 3 OR
        +  HDCS *  LA3 *  LA2 * /LA1 * /DD4 *  DD3 *  DDRIVE0   ;
1 TURNS OFF
       :+: DDRIVE0
        + /DDRIVE0 * HDCS * LA3 * LA2 * /LA1 * /DD4 * /DD3       ;
0 TURNS ON

DDRIVE1 := HDCS *  LA3 *  LA2 * /LA1 *  DD4 *          DDRIVE1   ;
2 OR 3 OR
        +  HDCS *  LA3 *  LA2 * /LA1 * /DD4 * /DD3 *  DDRIVE1   ;
0 TURNS OFF
       :+: DDRIVE1
        + /DDRIVE1 * HDCS * LA3 * LA2 * /LA1 * /DD4 *  DD3       ;
1 TURNS ON

HTSEL := HDCS *  LA3 * /LA2 *  LA1 *  DD5 * /HTSEL ;
      +  HDCS *  LA3 * /LA2 *  LA1 * /DD5 *  HTSEL ;
     :+: HTSEL

HDSEL3 := HDCS *  LA3 * /LA2 *  LA1 *  DD6 * /HDSEL3    ;
       +  HDCS *  LA3 * /LA2 *  LA1 * /DD6 *  HDSEL3    ;
      :+: HDSEL3

HDSEL2 := HDCS *  LA3 *  LA2 * /LA1 *  DD2 * /HDSEL2    ;
       +  HDCS *  LA3 *  LA2 * /LA1 * /DD2 *  HDSEL2    ;
      :+: HDSEL2

HDSEL1 := HDCS *  LA3 *  LA2 * /LA1 *  DD1 * /HDSEL1    ;
       +  HDCS *  LA3 *  LA2 * /LA1 * /DD1 *  HDSEL1    ;
      :+: HDSEL1

HDSEL0 := HDCS *  LA3 *  LA2 * /LA1 *  DD0 * /HDSEL0    ;
       +  HDCS *  LA3 *  LA2 * /LA1 * /DD0 *  HDSEL0    ;
      :+: HDSEL0
```

PAL 22: Hard Disk Phase-Locked-Loop Control

This PAL generates the pump up and the pump down signals for the hard disk phase-locked-loop voltage controlled oscillator.

```
PAL16R4A   PAL DESIGN SPECIFICATION
322A
FOR MP,8N  9/24/83

/MUXCLK NC1 /DATA NC2 NC3 HDRGATE PCLK VAR REF GND
/OE /PD /PU /FFA /OSCENB /FFB /FFC /CLR /MUX VCC

FFA := HDRGATE

FFB := FFA

FFC := FFB

OSCENB := /HDRGATE * /FFC
        +  HDRGATE *  FFC

IF (VCC) CLR = /OSCENB
             + VAR * REF

IF (VCC) PU = /VAR *  REF

IF (VCC) PD =  VAR * /REF

IF (VCC) MUX =  HDRGATE * DATA
             + /HDRGATE * PCLK
```

PAL 23: Interrupt Level Generator

This PAL decodes interrupt requests and outputs the interrupt priority level (IPL) field to the CPU. The IPL levels are as follows:

7	NMI
6	Time tick
5	8274's, prioritized by their IPI/IPO lines
4	The on-board interrupt controller (8259)
3	(An off-board interrupt controller)
2	(A non-INTA using off-board interrupt (e.g. Ethernet))
1	Line printer

```
PAL16L8A   PAL DESIGN SPECIFICATION
323A
FOR MP,25L 12/Ø3/83

/LIO  TMR1 LP   /SNMI /SI59B /SE   SI59A /SI74  PA23   GND
SUPV /PADENH /IPL2 /PADENL PA22  /LMEM /NC8   /IPL1  /IPLØ  VCC

IF(VCC) IPL2 =  SNMI +
               /SNMI *  TMR1 +
               /SNMI * /TMR1 *  SI74 +
               /SNMI * /TMR1 * /SI74 * SI59A

IF(VCC) IPL1 =  SNMI +
               /SNMI *  TMR1 +
               /SNMI * /TMR1 * /SI74 * /SI59A *  SI59B +
               /SNMI * /TMR1 * /SI74 * /SI59A * /SI59B * SE

IF(VCC) IPLØ =  SNMI +
               /SNMI * /TMR1 *  SI74 +
               /SNMI * /TMR1 * /SI74 * /SI59A *  SI59B +
               /SNMI * /TMR1 * /SI74 * /SI59A * /SI59B * /SE * LP

IF(VCC) PADENL = LIO  +  LMEM

IF(VCC) PADENH = LIO  * /SUPV +
                 LMEM * /SUPV +
                 LIO  *  PA23 +
                 LMEM *  PA23 +
                 LIO  * /PA22 +
                 LMEM * /PA22
```

PAL 24: Interrupt Acknowledge

This PAL decodes the interrupt acknowledge address generated by the CPU, and issues the interrupt acknowledge (INTA) signal to the selected device. The PAL implements a double INTA cycle for the 8274 port controller and 8259 programmable interrupt controller, to allow these chips to operate in vectored interrupt mode.

```
PAL16LA    PAL DESIGN SPECIFICATION
324B
FOR MP,26K 9/24/83

/AAS /LIO /NC1 AT4 INTACLK INTAADDR PA3 PA2 PA1 GND
/RESET    /NMICLR       /INTA8274   /INTA8259A  /ETINTA
/INTALP   /INTA8259B    /INTATMR1    /IODTK      VCC

IF(VCC) NMICLR    = INTAADDR *             PA3 *  PA2 *  PA1 *
/RESET * AAS
+ RESET


IF(VCC) INTATMR1   = INTAADDR            *  PA3 *  PA2 * /PA1 *
/RESET * AAS

IF(VCC) INTA8274  = INTAADDR * /INTACLK *  PA3 * /PA2 *  PA1 *
/RESET * AAS

IF(VCC) INTA8259A = INTAADDR * /INTACLK *  PA3 * /PA2 * /PA1 *
/RESET * AAS

IF(VCC) INTA8259B = INTAADDR * /INTACLK * /PA3 *  PA2 *  PA1 *
/RESET * AAS

IF(VCC) ETINTA    = INTAADDR            * /PA3 *  PA2 * /PA1 *
/RESET * AAS

IF(VCC) INTALP    = INTAADDR            * /PA3 * /PA2 *  PA1 *
/RESET * AAS

IF(VCC) IODTK     = INTAADDR            * LIO * AT4 * /PA3 *  PA2
+
                    INTAADDR            * LIO * AT4 *  PA3 * /PA2
+
                   /INTAADDR            * LIO * AT4
```

PAL 25: Error Control

This PAL generates the error signals for the processor. The CPU enables NMI- and BERR- in this PAL by setting the EE bit in the General Control register. Parity nonmaskable interrupts require PIE (parity interrupt enable, in the General Control register) in addition to EE.

```
PAL16L8A   PAL DESIGN SPECIFICATION
325A
FOR MP,17H 12/03/83

AT2 NPCYCLE /LWT /MNP /NMIFF /RG PE /LRDEN /PREG  GND
/UIOER /NMI T200 /BERR /MMUERR PIE /PGF EE  /EDTK  VCC

IF(VCC) NMI  = EE * /RG *  MNP  * /PGF * /MMUERR            +
               EE * /RG * /MMUERR * PIE * PE * /LWT *
                    T200 * /UIOER * /NPCYCLE * /PREG       +
               EE * /RG * /MMUERR * PIE * PE * /LWT * T200 *
NPCYCLE+
               EE * /RG *  PGF    * NPCYCLE                +
               EE * /RG *  MMUERR * NPCYCLE                +
               NMIFF

IF(VCC) EDTK = EE * /RG * LRDEN * /NPCYCLE * /MMUERR  * PIE *
                    PE * T200 * /UIOER * /PREG +
               EE * /RG * MNP   * /NPCYCLE              +
               EE * /RG * PGF   * /NPCYCLE * T200

IF(VCC) BERR = EE * /RG * PGF   *  /NPCYCLE * T200    +
               EE * /RG * UIOER *  /NPCYCLE * T200    +
               EE * /RG * UIOER *  /NPCYCLE * AT2     +
               EE * /RG * MMUERR * /NPCYCLE * T200
```

PAL 26: Memory Error Latch

This PAL controls the latching of the Bus Status register and also generates the signal to update the page table. The latching of the Bus Status register is controlled by an RS latch implemented by the PAL equations. The register latches when a nonmaskable interrupt, bus error, or memory not present condition occurs and asserts NMI-, BERR-, or MNP-, respectively.

```
PAL16L8A   PAL DESIGN SPECIFICATION
326A
FOR MP,15J 9/24/83

/GSRCL /BASEMEM /BERR T26Ø /NMI T38Ø /ENCAS /CASACK T32Ø GND
/EDTK /MNP RESET /GSRHL /GSRHH PA23 /GSRLH /GSRLL /UPT VCC

IF(VCC) MNP    = ENCAS * /BASEMEM * T32Ø * T26Ø * /CASACK

IF(VCC) GSRHL = /GSRHH +
                EDTK  * T32Ø                    * /RESET * /GSRCL
+
                BERR  * T32Ø */PA23             * /RESET * /GSRCL
+
                BERR         * PA23             * /RESET * /GSRCL
+
                NMI   * T32Ø                    * /RESET * /GSRCL
+
                ENCAS * /BASEMEM * T32Ø * T26Ø * /CASACK

IF(VCC) GSRHH = /GSRHL +
                 GSRCL+
                 RESET

IF(VCC) GSRLL = /GSRLH +
                EDTK  * T32Ø                    * /RESET * /GSRCL
+
                BERR  * T32Ø */PA23             * /RESET * /GSRCL
+
                BERR         * PA23             * /RESET * /GSRCL
+
                NMI   * T32Ø                    * /RESET * /GSRCL
+
                ENCAS * /BASEMEM * T32Ø * T26Ø * /CASACK

IF(VCC) GSRLH = /GSRLL +
                 GSRCL +
                 RESET

IF(VCC) UPT    = T26Ø * /T32Ø * ENCAS * BASEMEM  +
                 T26Ø * /T32Ø * CASACK
```

PAL 27: Memory Control

This PAL controls memory accesses and CPU fast cycle register accesses. The PAL also assists in the page table update timing. CLRRAS clears the RAS2 flop. PADLTCH- controls the latching of the upper address bits of the processor address.

```
PAL16L8A   PAL DESIGN SPECIFICATION
327A
FOR MP,19E 12/Ø2/83

/MCWT T17Ø T2ØØ T23Ø T29Ø T38Ø T44Ø RAS /MMUERR        GND
/REFGNT /CLRREF /CLRRAS /REFRAS /PREG /XWE /ROE /ENCAS /PADLTCH
VCC

IF(VCC) CLRRAS = T38Ø * /T44Ø + REFGNT * /REFRAS * T17Ø

IF(VCC) CLRREF = REFGNT * T29Ø * /T38Ø

IF(VCC) XWE    = MCWT

IF(VCC) ROE    = /REFGNT * /MMUERR *  T17Ø * /PREG +
                 /REFGNT * /MMUERR *  T2ØØ * /PREG

IF(VCC) ENCAS  = /REFGNT * /MCWT * T17Ø * /PREG * RAS +
                 /REFGNT *  MCWT * T23Ø * T17Ø * /PREG * RAS

IF(VCC) REFRAS = REFGNT * /T38Ø +
                 REFGNT * /T29Ø +
                 REFGNT * /T23Ø +
                 REFGNT * /T44Ø

IF(VCC) PADLTCH = /REFGNT * /MCWT * T17Ø * /PREG  +
                  /REFGNT *  MCWT * T29Ø * T17Ø * /PREG
```

PAL 28: Line Printer Controller

This PAL controls (1) reading the status register, (2) writing data, and (3) printer interrupts. It also performs an OR function to generate VPA- for the CPU.

```
PAL16R4A   PAL DESIGN SPECIFICATION
328B
FOR MP,5H  9/24/83

/PCLK /LWT /INTATMR1 LA2 /NMICLR /RESET LA1 /INTALP /LPSEL  GND
/OE /LPWRITE /XINTA NC7 NC8 /LPINTEN /LPRSTINTR /VPA /LPREAD VCC


IF (VCC) LPREAD = LPSEL * /LWT * /LA2 * /LA1

IF (VCC) LPWRITE = LPSEL * LWT * /LA2 * /LA1

LPRSTINTR := RESET
           + /LPINTEN
           + INTALP

LPINTEN := LPSEL * LWT * /LA2 *  LA1 * /RESET
         + LPINTEN * /LPSEL * /RESET
         + LPINTEN * /LWT   * /RESET
         + LPINTEN * /LA2   * /RESET
         + LPINTEN *  LA1   * /RESET

IF (VCC) VPA = INTALP
             + NMICLR
             + INTATMR1
             + XINTA
```

PAL 29: Map Control

This PAL controls the map RAMs and buffers during loading and reading. It also handles the updating of the page status bits during each access.

```
PAL2ØL1Ø   PAL DESIGN SPECIFICATION
329A
FOR MP,13H 9/24/83

LMA2Ø /UDS /UPT LPS1 /MMUERR LPSØ /MCWT /CLRRAS LWE /MMUSEL /LDS
GND
/NC1 MA2Ø /PTWTU /PTWTL WE PSØ PS1 MALDEN2 /MALDEN /NC2  /NC3 VCC

IF(UPT*/MMUERR) /MA2Ø  = /LMA2Ø

IF(UPT*/MMUERR) /PS1   = /LPS1 * /LPSØ

IF(UPT*/MMUERR) /PSØ   = /LPSØ * /LPS1  +
                         /LPSØ * /MCWT  +
                         /LPS1 * /MCWT

IF(UPT*/MMUERR) /WE    = /LWE

IF(VCC) PTWTL  = MMUSEL * /CLRRAS * LDS * MCWT

IF(VCC) PTWTU  = UPT * /MMUERR +
                 MMUSEL * /CLRRAS * UDS * MCWT

IF(VCC) MALDEN = MMUSEL *  MCWT   * /CLRRAS +
                 MMUSEL * /MCWT

IF(VCC) /MALDEN2 = VCC
```

PAL 30: Memory Error Monitor

This PAL implements the memory protection mechanism. It generates the memory access error signals, as well as CASINH- (the signal that inhibits the generation of the CAS signal). In addition, it ANDs together the CPU function code outputs to detect a CPU space cycle.

```
PAL16L8A   PAL DESIGN SPECIFICATION
330A
FOR MP,18H 9/24/83

LPS1 LPSØ LWE /XGNT SUPV PA19 FCØ PROGRAM PA21  GND
PA22 /FCHI /MCWT /RAS PA2Ø /MMUERR /LMEM /PGF /CASINH VCC

IF(VCC) FCHI    = SUPV * PROGRAM * FCØ

IF(VCC) CASINH = MCWT * /LWE * /SUPV * RAS * LMEM * /PA22 +
                 MCWT * /LPS1 * /LPSØ * RAS * /XGNT +
                 /PA22 * /PA21 * /PA2Ø * /PA19 * /SUPV * RAS *
LMEM +
                 RAS * LMEM * PA22

IF(VCC) PGF     = RAS * /LPS1 * /LPSØ * /XGNT * /LMEM +
                  RAS * /LPS1 * /LPSØ * /PA22 *  LMEM

IF(VCC) MMUERR = MCWT * /LWE * /SUPV * RAS * LMEM * /PA22 +
                 RAS * /LPS1 * /LPSØ * /XGNT * /LMEM +
                 RAS * /LPS1 * /LPSØ * /PA22 *  LMEM +
                 /PA22 * /PA21 * /PA2Ø * /PA19 * /SUPV * RAS *
LMEM
```

PAL 31: CPU State Machine

This PAL synchronizes the bus activity with the processor cycles, and also is used to generate the DTACK signal for the processor.

```
PAL16R6A   PAL DESIGN SPECIFICATION
331A
FOR MP,26E 9/24/83

/PCLK /WRITE /EDTK T17Ø /IODTK /PREG SRAS1 /LMEM /AAS   GND
/OE /DTACK /PRS2 /PRS1 /PRSØ /CLRRAS1 /LRDEN /LWREN RESET VCC

PRS2 := PRS1 * /PRSØ * AAS

PRS1 := /PRS2 *  PRSØ +
         PRS1 * /PRSØ * AAS

PRSØ := /PRS2 * /PRS1 * /PRSØ * SRAS1 +
        /PRS2 * /PRS1 *  PRSØ         +
         PRS2 *  PRS1 * /PRSØ * AAS

LRDEN := /PRS2 *  PRS1 *  PRSØ * /WRITE * /PREG

LWREN := /PRS2 *  PRS1 *  PRSØ *  WRITE

CLRRAS1 := /PRS2 *  PRS1 *  PRSØ        +
           /PRS2 *  PRS1 * /PRSØ * AAS  +
            PRS2 *  PRS1 * /PRSØ * AAS +
            RESET

IF(VCC) DTACK =  SRAS1 * T17Ø +
                 IODTK +
                 EDTK
```

PAL 32: Slow Communications Control

This PAL controls the slow communications bus interface. The PAL manages the MOS bus that connects the logical data bus to the 8274 port controller, 8253 timer, and 8259 programmable interrupt controller chips.

```
PAL20L10   PAL DESIGN SPECIFICATION
332A
FOR MP,9K  9/24/83

NPCYCLE RESET /LWT LA4 LA3 LA2 LA1 /SCSEL /ICASEL /IA74A /SCTSEL
GND
/IA59A /MOE /MWR /MRD /CS8274 /MDIR /MB1 /R8274 /ICACS /EXTRD
/EXWR VCC

IF(VCC) R8274  = RESET +
                 SCSEL * LWT * /LA4 *  LA3 * /LA2 *  LA1

IF(VCC) MRD  = SCSEL  * /LWT * /LA4 * /LA3 +
               SCTSEL * /LWT * /LA4 * /LA3 +
               ICASEL * /LWT * /LA4 * /LA3

IF(VCC) MWR  = SCSEL  *  LWT * /LA4 * /LA3 +
               SCTSEL *  LWT * /LA4 * /LA3 +
               ICASEL *  LWT * /LA4 * /LA3

IF(VCC) CS8274 = SCSEL * /LA4 * /LA3

IF(VCC) EXTRD  = SCSEL * /LA4 *  LA3 * /LA2 * /LA1 * /LWT

IF(VCC) MOE = MB1          +
               SCTSEL      +
               ICASEL

IF(VCC) MB1    = SCSEL          +
               IA74A * /NPCYCLE     +
               IA59A * /NPCYCLE

IF(VCC) MDIR = LWT

IF(VCC) EXWR = SCSEL * /LA4 *  LA3 * /LA2 * /LA1 * LWT

IF(VCC) ICACS     = ICASEL * /LA4 * /LA3
```

PAL 33: Baud Rate Generator Control

This PAL controls the slow communications 8253 timer chip used to generate baud rates. It also generates the data strobes for the nonprocessor bus cycles. All DMA cycles are word-mode only.

```
PAL16L8A   PAL DESIGN SPECIFICATION
333A
FOR MP,11N 9/24/83

/ROMSEL /SCTSEL /LWT  NC1  LA1  LA2  LA3  LA4  NPCYCLE    GND
/TIMRRD /W8253 /R8253 /C8253 /PDBOE /NC5 /NC6 /LUDS /LLDS VCC

IF (VCC) R8253 = SCTSEL * /LWT * /LA4 * /LA3

IF (VCC) W8253 = SCTSEL *  LWT * /LA4 * /LA3

IF (VCC) C8253 = SCTSEL * /LA4 * /LA3

IF (VCC) PDBOE = /NPCYCLE * /ROMSEL * /TIMRRD

IF (NPCYCLE) LUDS = VCC

IF (NPCYCLE) LLDS = VCC
```

PAL 34: Slow Communications Select Register

This PAL has two functions: (1) it acts as a register to store the user selection of clock source (SELA, SELB) and the secondary transmit data (STDA, STDB); (2) it acts as multiplexer to route the clock source (internal or external), based on the SEL bits.

```
PAL20X4     PAL DESIGN SPECIFICATION
334A
FOR MP,6M  9/24/83

/SCEXREGWR   NØA  N1A  NØB  N1B  T1  TØ  SCD3   SCD2   SCD1
SCDØ  GND
/OE /TXCA /RXCA /TXCB /SCISTDB /SCISTDA /SELA /SELB /RXCB /NC1
/NC2 VCC

SCISTDA := SCD3

SCISTDB := SCD2

SELA    := SCD1

SELB    := SCDØ

IF(VCC) TXCA =  SELA * NØA +
               /SELA * TØ

IF(VCC) TXCB =  SELA * NØB +
               /SELA * T1

IF(VCC) RXCA =  SELA * N1A +
               /SELA * TØ

IF(VCC) RXCB =  SELA * N1B +
               /SELA * T1
```

PAL 35: Disk Word State Sequencer

This PAL generates the controls for the word buffer. The word buffer is the interface between the byte buffer and main memory.

```
PAL16R6A   PAL DESIGN SPECIFICATION
335A
FOR MP,22L 9/26/83

/PCLK /NMIFF  /DGNT /BREQ /DSKDMAWR /BIURESET /DSKDMAEN  /DSEL
LA5    GND
/OE   DISKTC /WØ   /W1   /W2       /W3       /DSKDMAREQ /BACK
/BRESET VCC


W3 := /W3 *  W2 * /W1 *  WØ * /BIURESET *  DSKDMAWR * DISKTC
    +  W3 * /W2 * /W1 * /WØ * /BIURESET *  DSKDMAEN

W2 := /W3 *       /W1 * /WØ * /BIURESET *  DSEL * /LA5
    + /W3 * /W2 * /W1 * /WØ * /BIURESET *  DSKDMAEN * /DSKDMAWR
    + /W3 * /W2 *  W1 * /WØ * /BIURESET
    + /W3 *  W2 * /W1 * /WØ * /BIURESET *  DSKDMAEN * /NMIFF
    + /W3 *  W2 *  W1 * /WØ * /BIURESET *  DSEL
    + /W3 *  W2 *  W1 * /WØ * /BIURESET *  DSKDMAEN * /DSEL

W1 := /W3 * /W2 * /W1 * /WØ * /BIURESET *  DSEL * /LA5
    + /W3 * /W2 * /W1 *  WØ * /BIURESET *  DGNT
    + /W3 * /W2 *  W1 *  WØ * /BIURESET
    + /W3 *  W2 * /W1 * /WØ * /BIURESET *  DSEL * /LA5
    + /W3 *  W2 *  W1 * /WØ * /BIURESET *  DSEL

WØ := /W3 * /W2 * /W1 * /WØ * /BIURESET *  DSKDMAEN * DSKDMAWR
    + /W3 * /W2 * /W1 *  WØ * /BIURESET
    + /W3 *  W2 * /W1 * /WØ * /BIURESET *  DSKDMAEN * BREQ
    + /W3 *  W2 * /W1 *  WØ * /BIURESET *  DSKDMAWR * /DISKTC
    + /W3 *  W2 * /W1 *  WØ * /BIURESET * /DSKDMAWR

DSKDMAREQ := /W3 * /W2 * /W1 * /WØ * /BIURESET *  DSKDMAEN *
DSKDMAWR
           + /W3 * /W2 * /W1 *  WØ * /BIURESET
           + /W3 * /W2 *  W1 *  WØ * /BIURESET
           + /W3 *  W2 * /W1 *  WØ * /BIURESET *  DSKDMAWR *
/DISKTC
           + /W3 *  W2 * /W1 *  WØ * /BIURESET * /DSKDMAWR

BACK := /W3 *  W2 * /W1 * /WØ * /BIURESET * DSKDMAEN * BREQ

BRESET := /W3 *  W2 * /W1 * /WØ * /BIURESET * /DSKDMAWR * /DSEL *
DISKTC
        + /W3 *  W2 * /W1 * /WØ * /BIURESET * /DSKDMAEN
        + /W3 *  W2 * /W1 * /WØ * /BIURESET *  NMIFF
        +  W3 * /W2 * /W1 * /WØ * /BIURESET *  DSKDMAEN *  BREQ
```

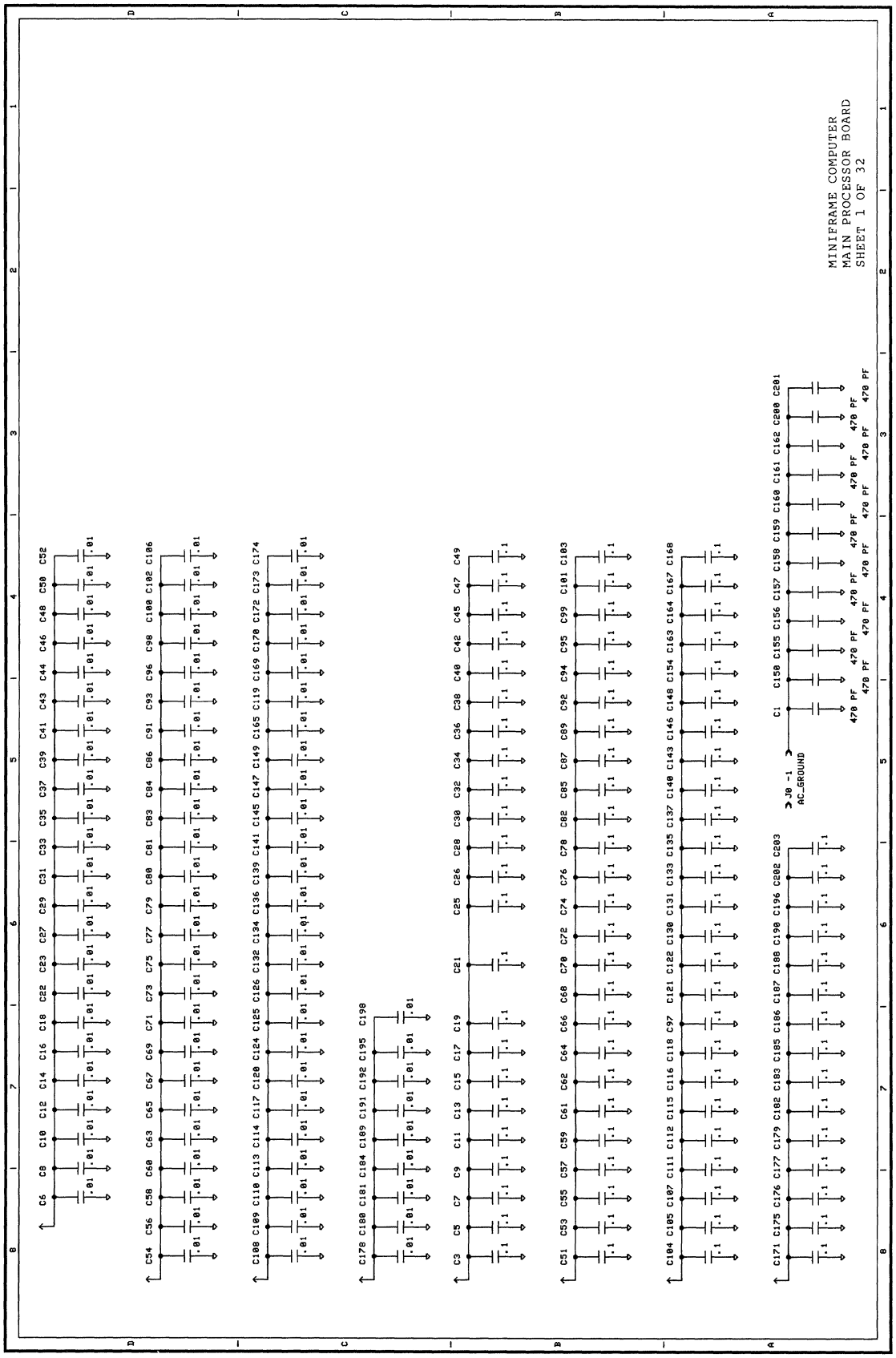
PAL 36: Miscellaneous Logic

This PAL performs miscellaneous logical functions. It generates two bus status signals (NPCYCLE, SETRAS2) and three memory control signals (BØROE, PLLDS, and PLUDS).

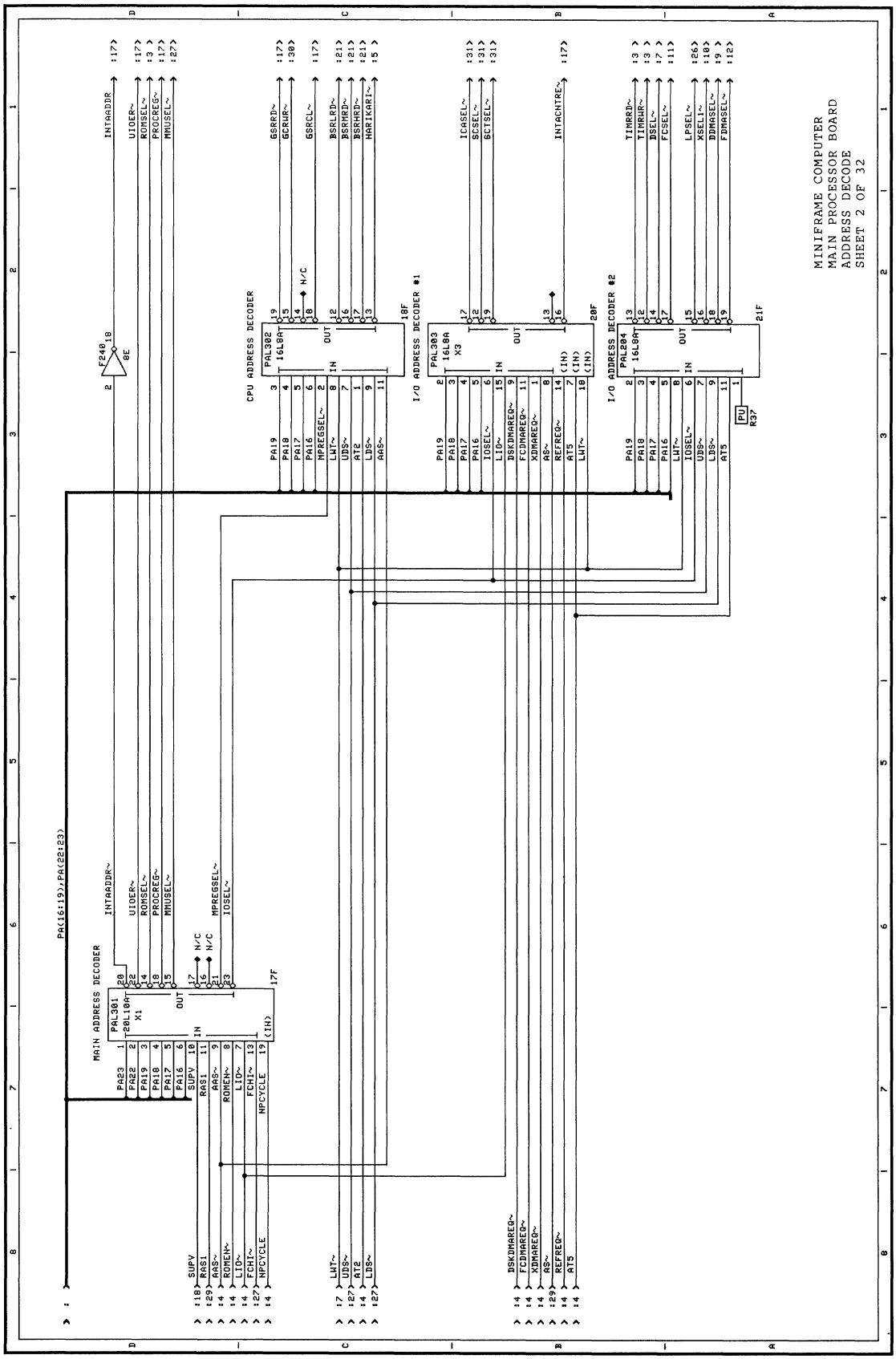
```
PAL16L8A   PAL DESIGN SPECIFICATION
336A
FOR MP,2ØE 9/24/83

/REFGNT /DGNT  /FCGNT  /XGNT  PCLK  RAS2  /ROE  /LUDS /LLDS GND
/BASEMEM NPCYCLE PLLDS PLUDS /BØROE /NC1 /NC2 /NC3 SETRAS2 VCC

IF (VCC) BØROE = ROE * BASEMEM

IF (VCC) /PLUDS = /LUDS +
                  /BASEMEM

IF (VCC) /PLLDS = /LLDS +
                  /BASEMEM

IF (VCC) /NPCYCLE =  /REFGNT * /DGNT * /FCGNT * /XGNT

IF (VCC) /SETRAS2 =  /REFGNT * /DGNT * /FCGNT * /XGNT +
                     /PCLK * /RAS2
```

APPENDIX B: SCHEMATICS

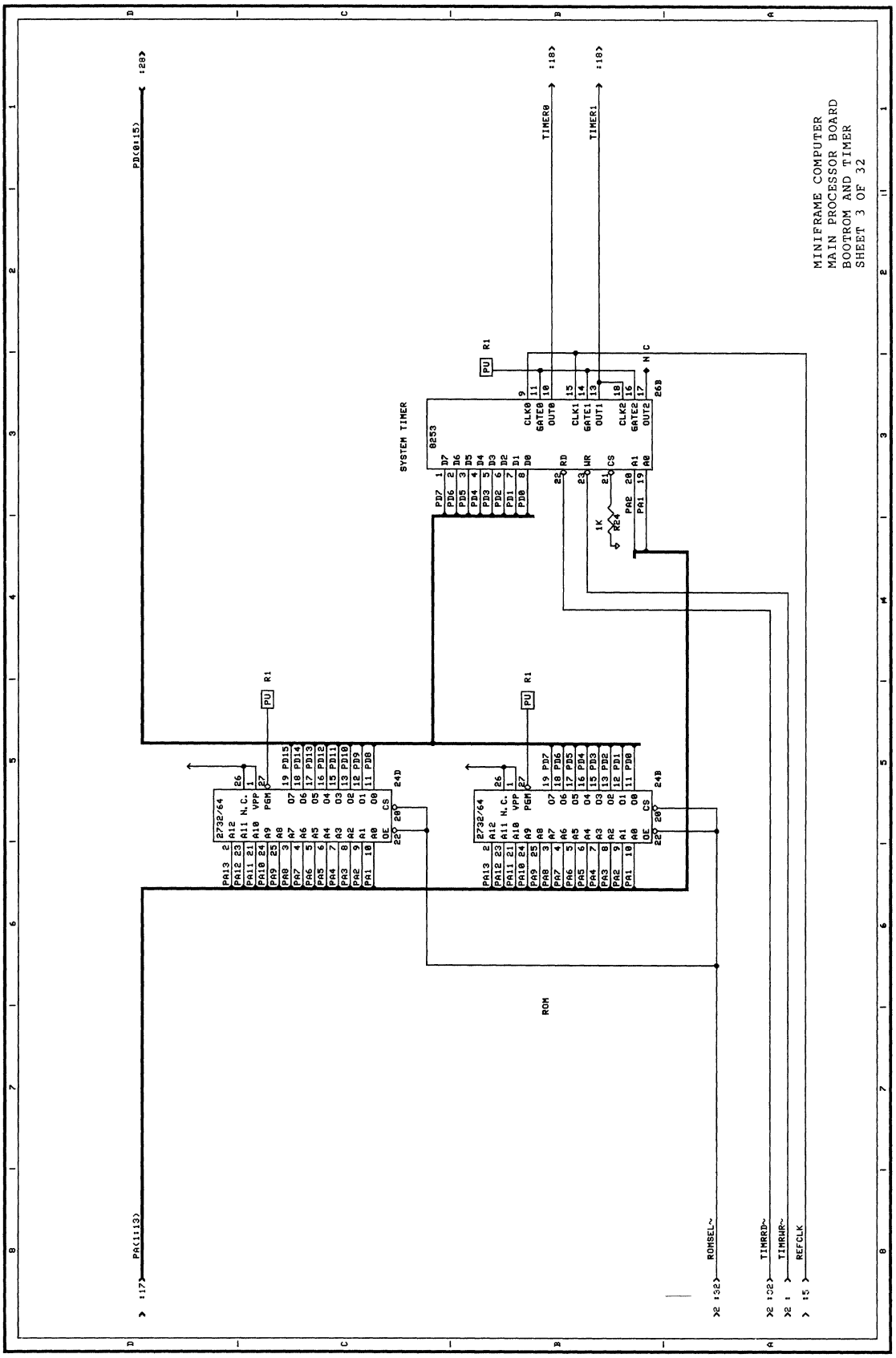
This appendix contains two sets of schematics sheets: one for the Main Processor board and the other for a Memory Expansion board.



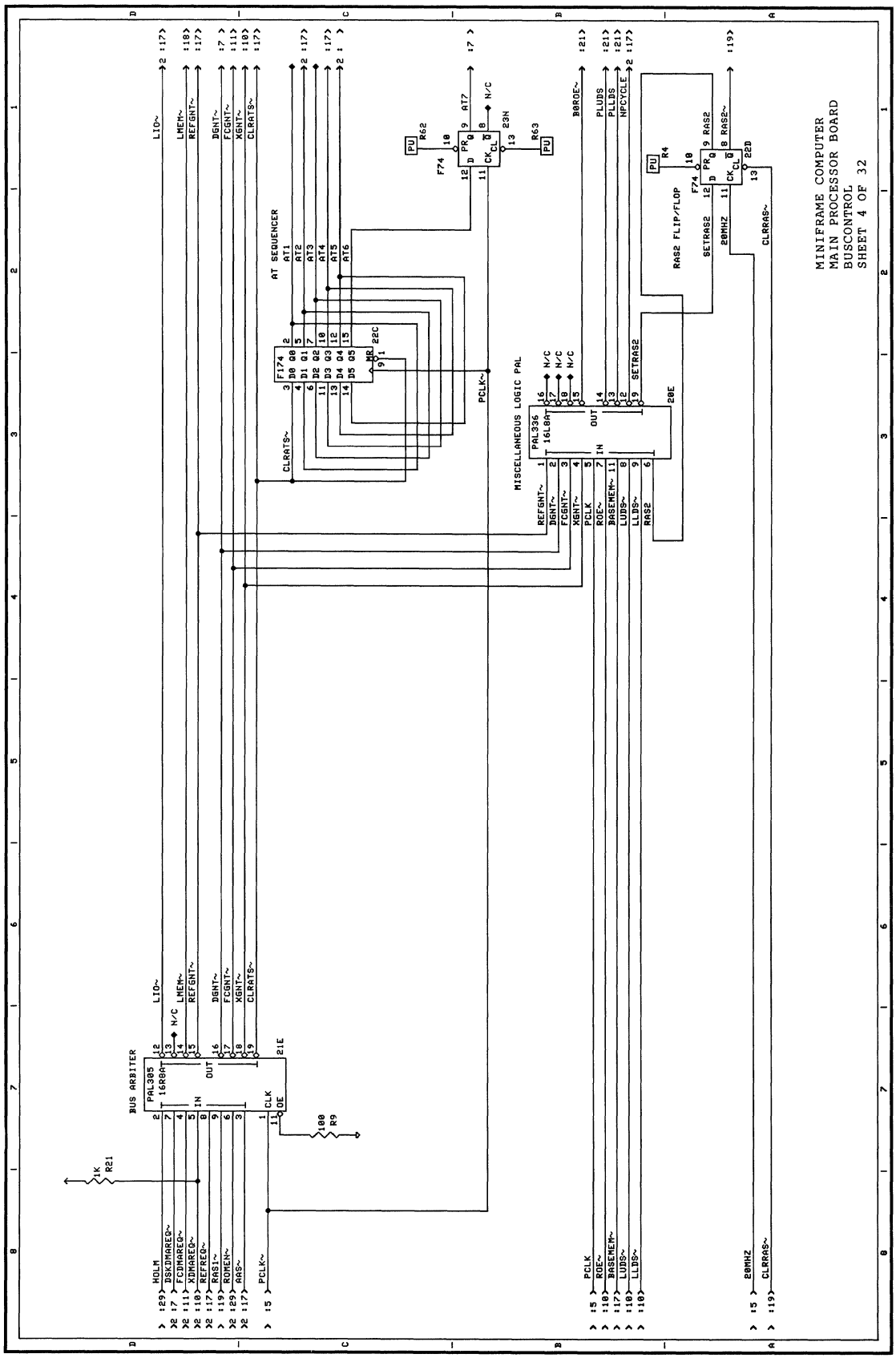
Sheet 1 of 32: Main Processor Board



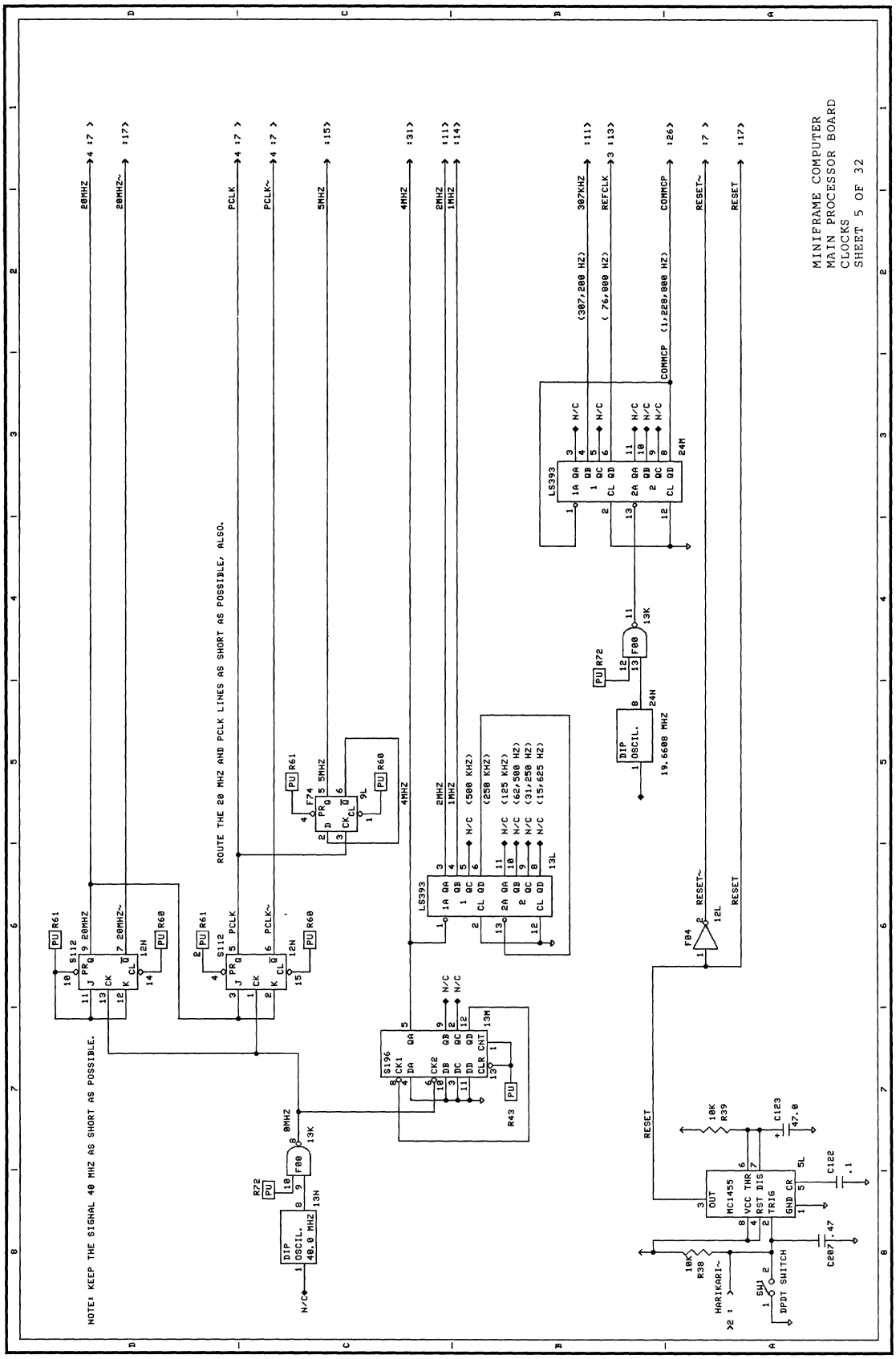
Sheet 2 of 32: Address Decode



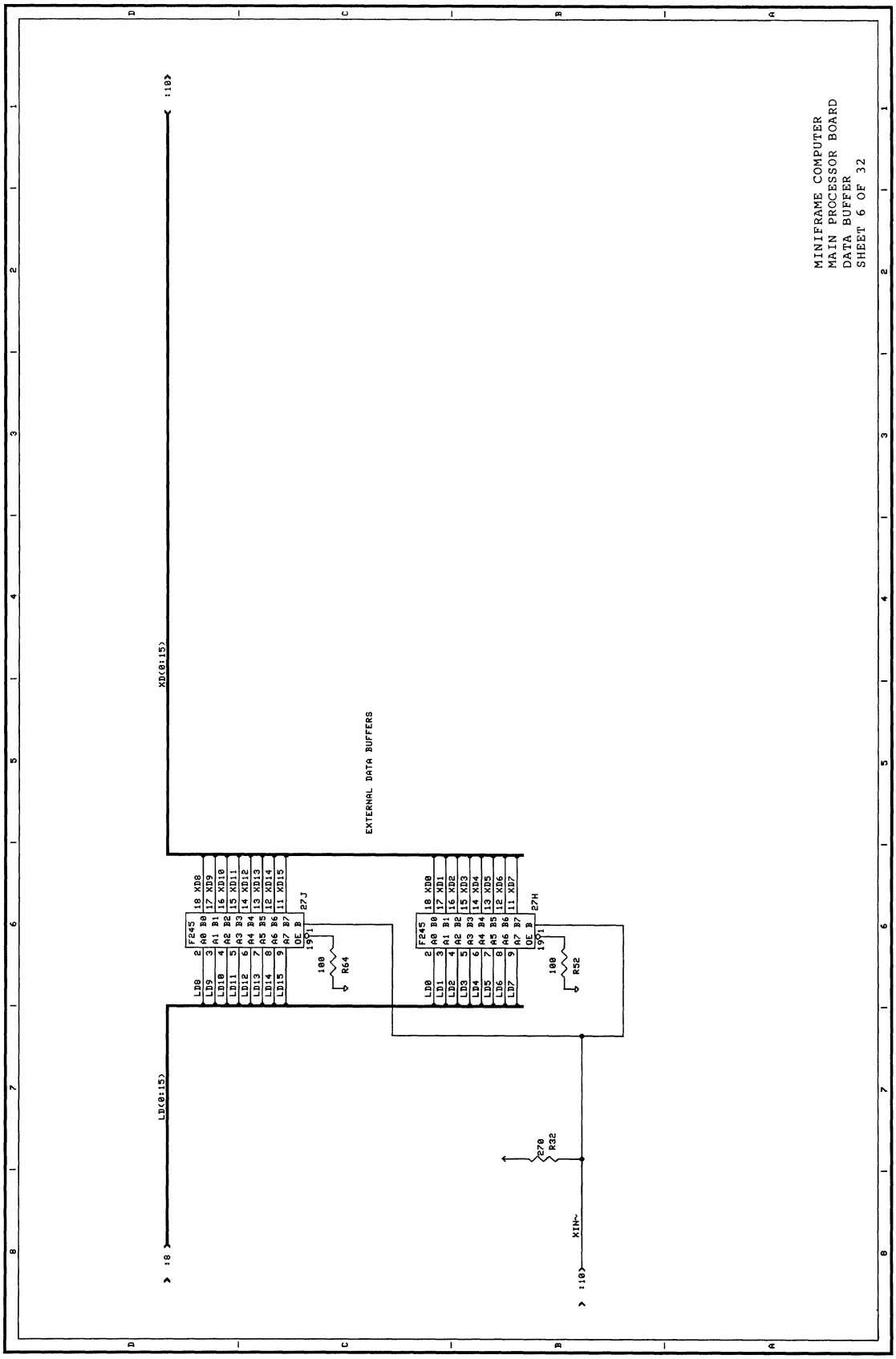
Sheet 3 of 32: ROM and Timer



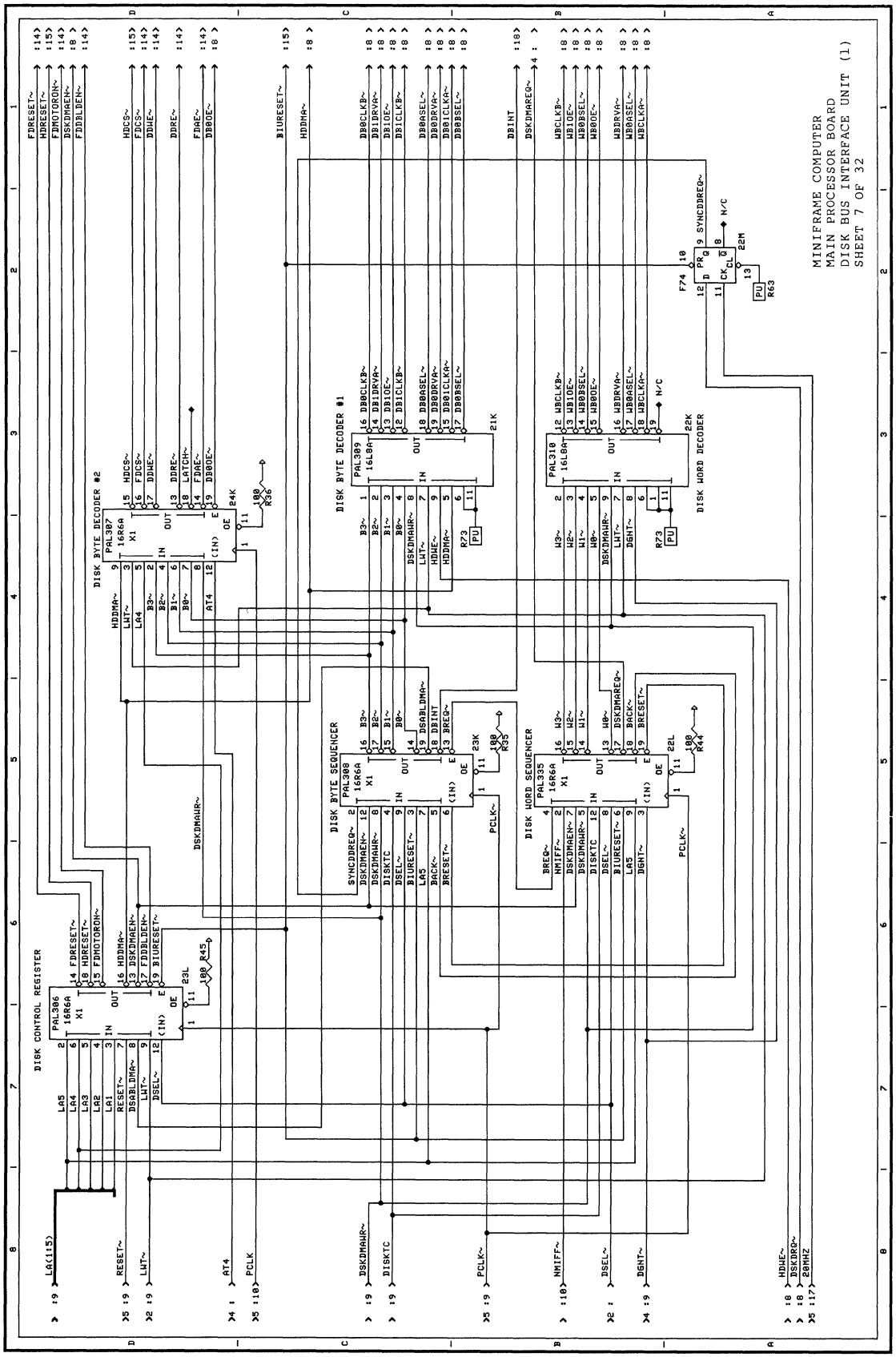
Sheet 4 of 32: Bus Control



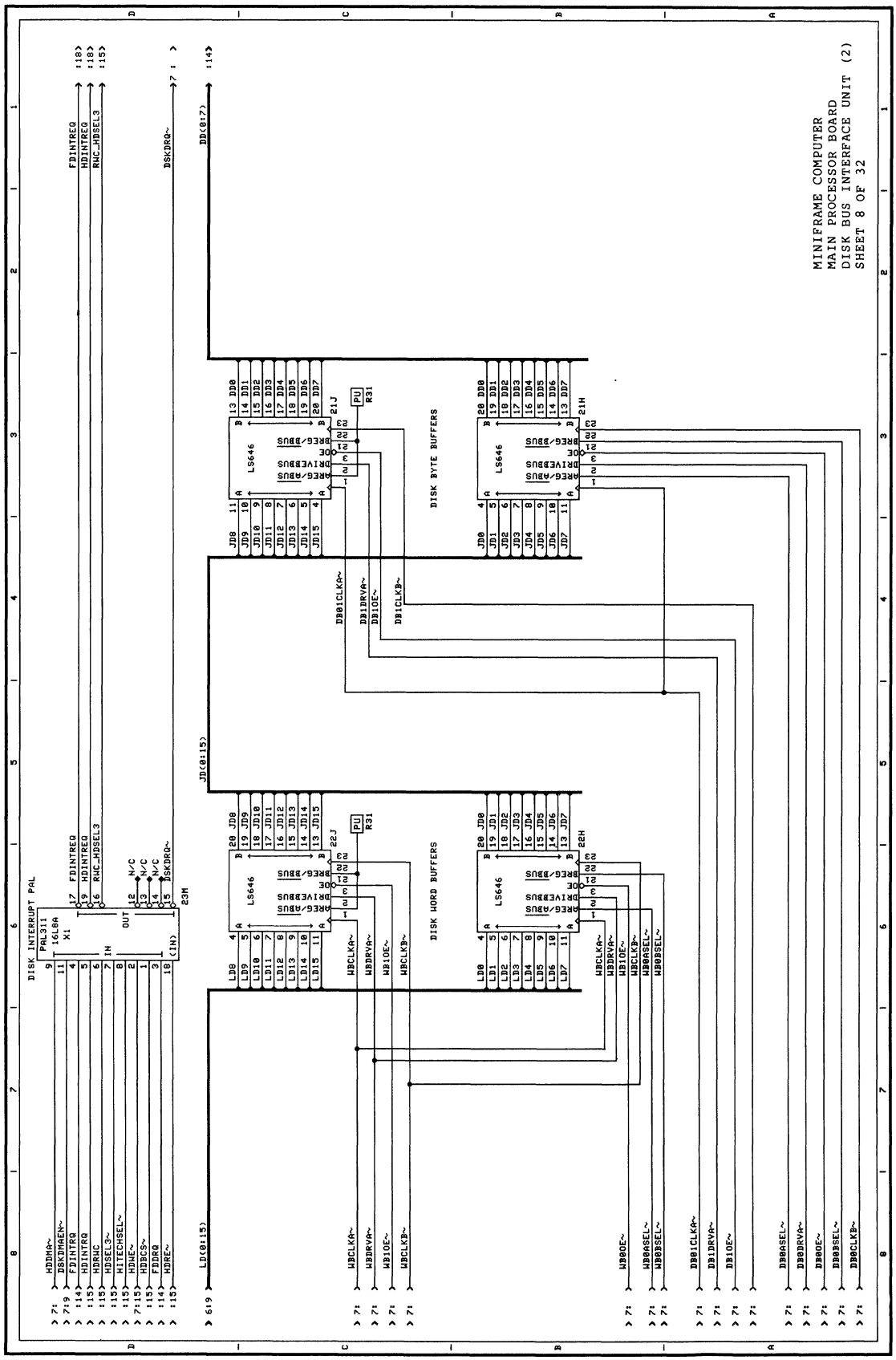
Sheet 5 of 32: Clocks



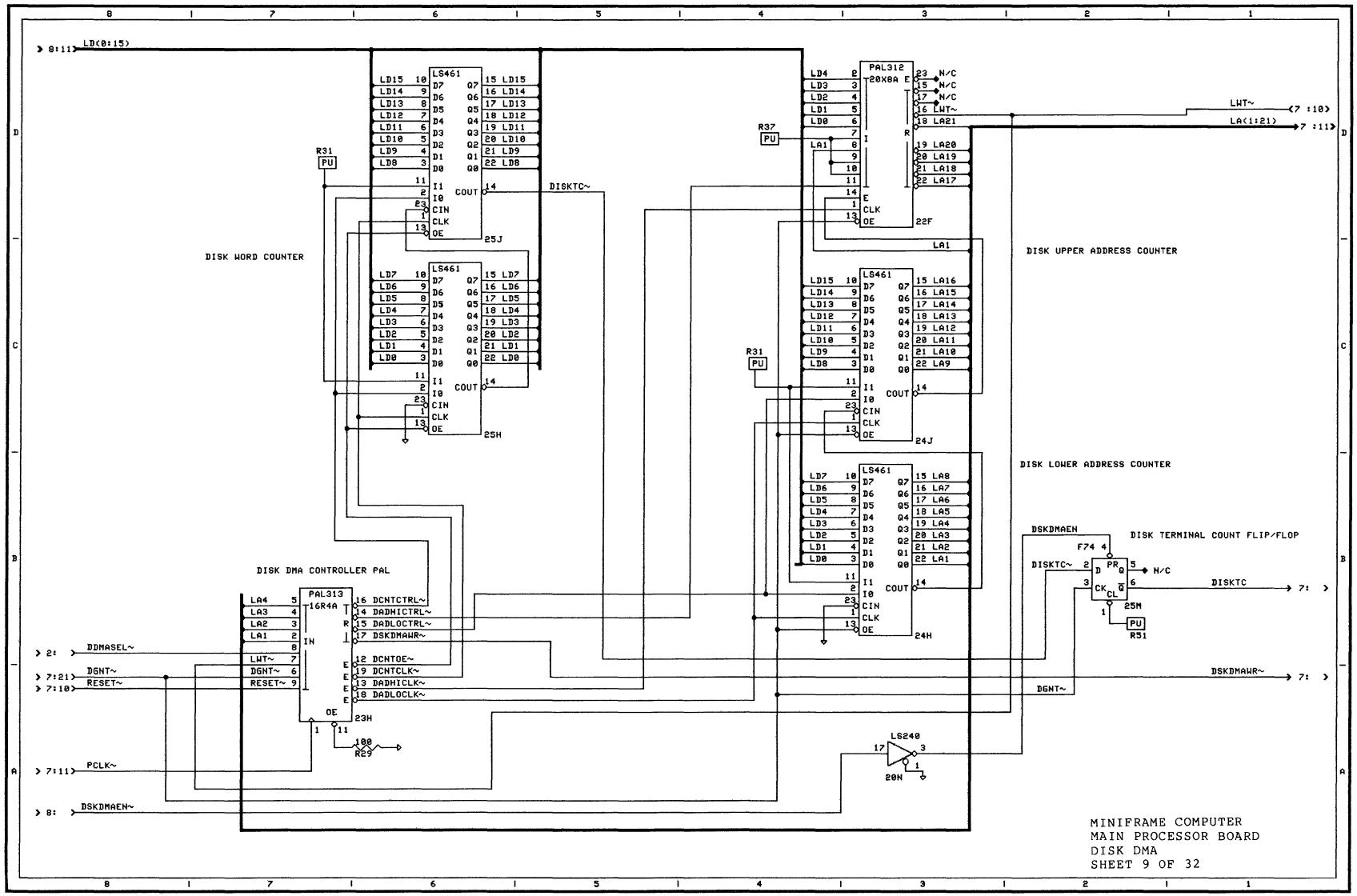
Sheet 6 of 32: Data Buffer



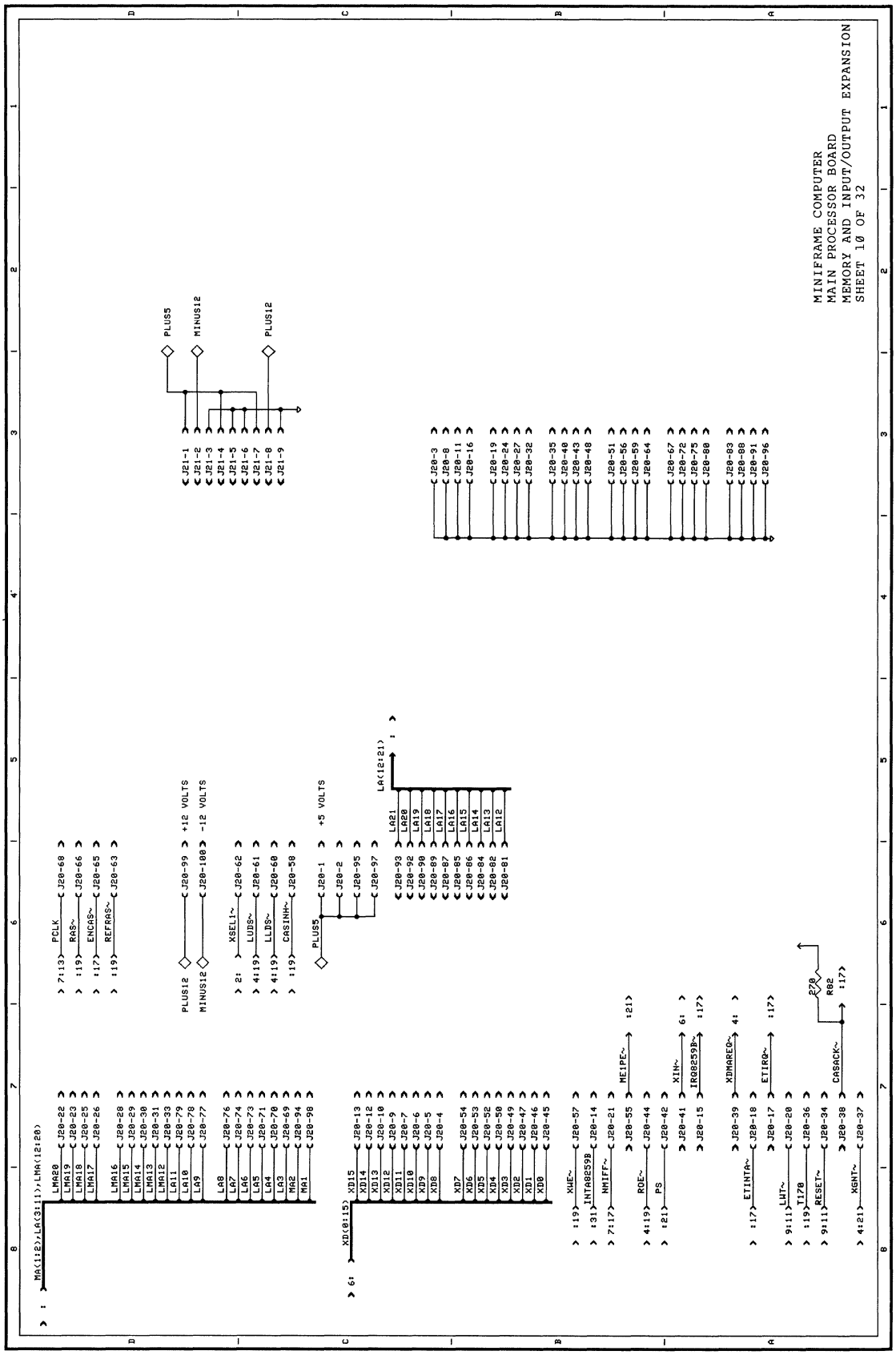
Sheet 7 of 32: Disk Bus Interface Unit (1)



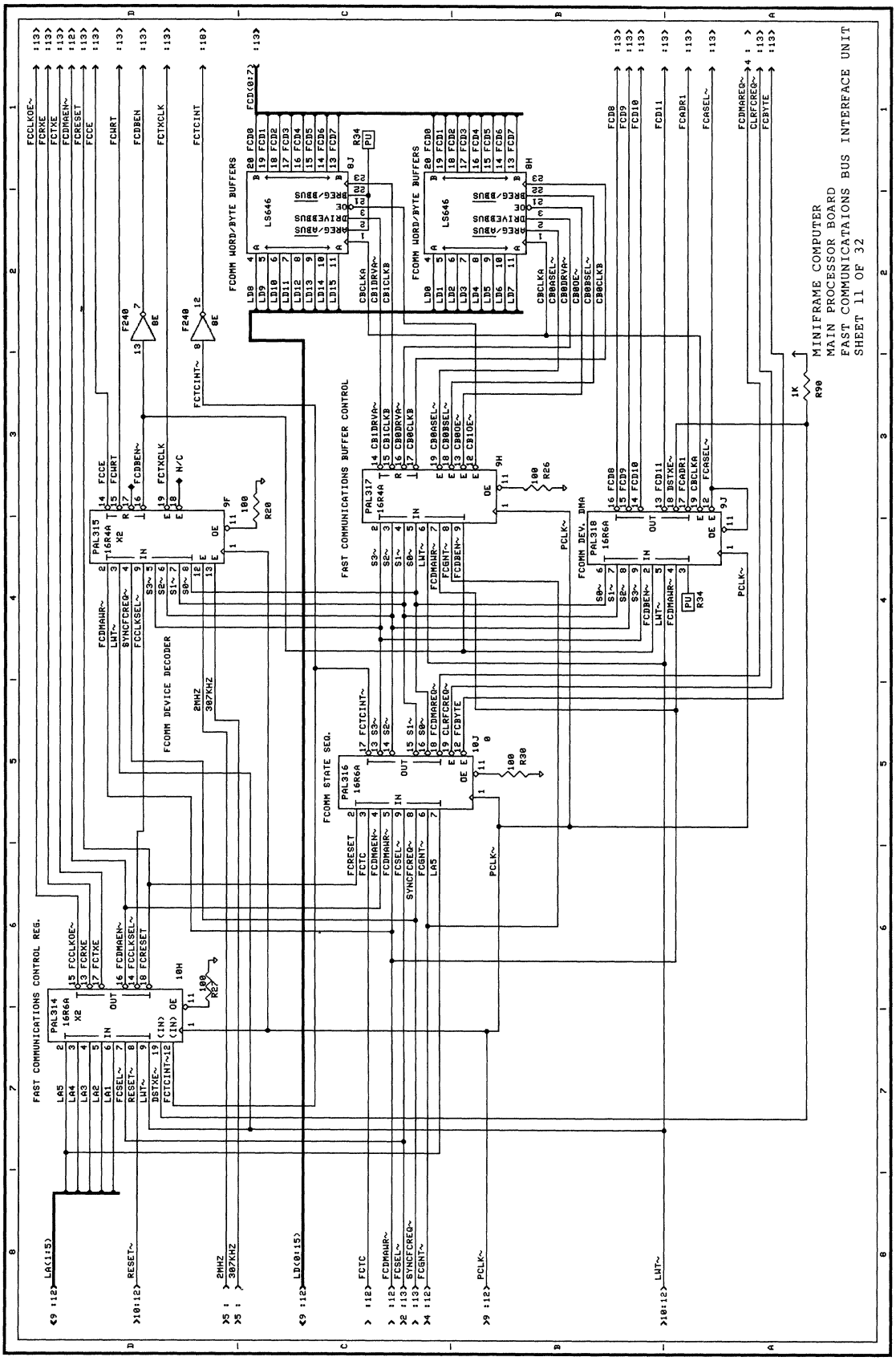
Sheet 8 of 32: Disk Bus Interface Unit (2)



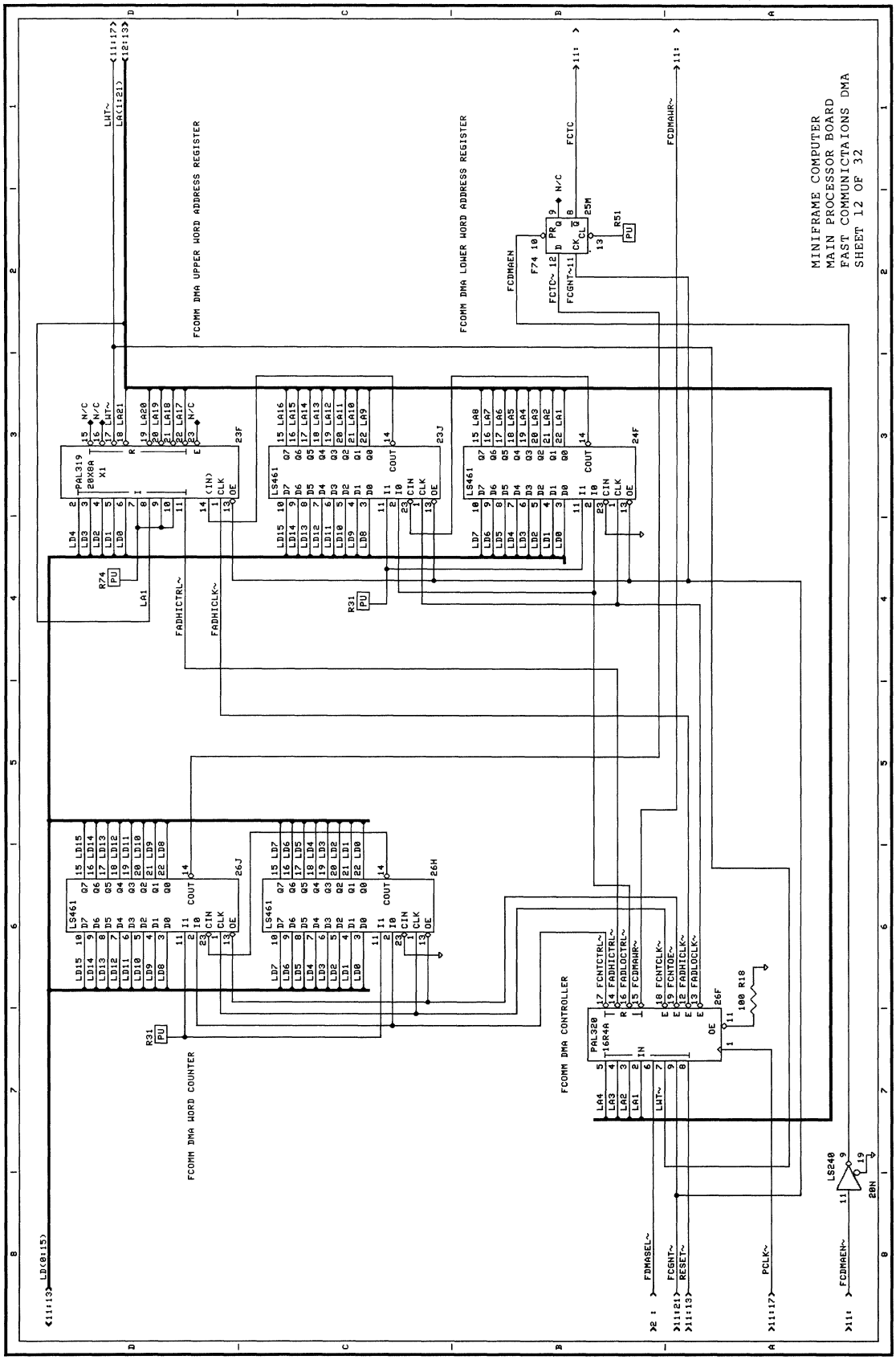
Sheet 9 of 32: Disk DMA



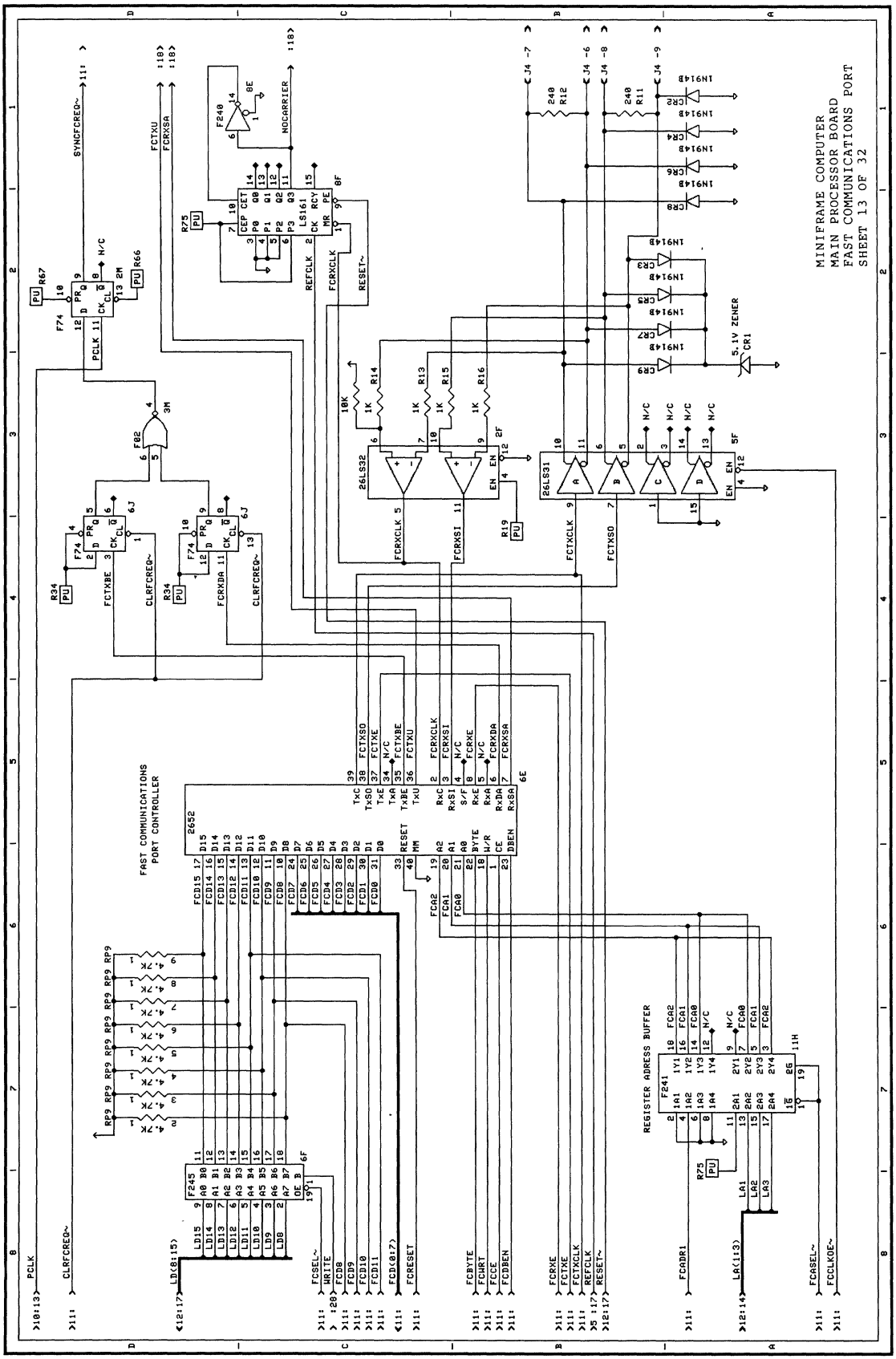
Sheet 10 of 32: Expansion Memory and Input/Output



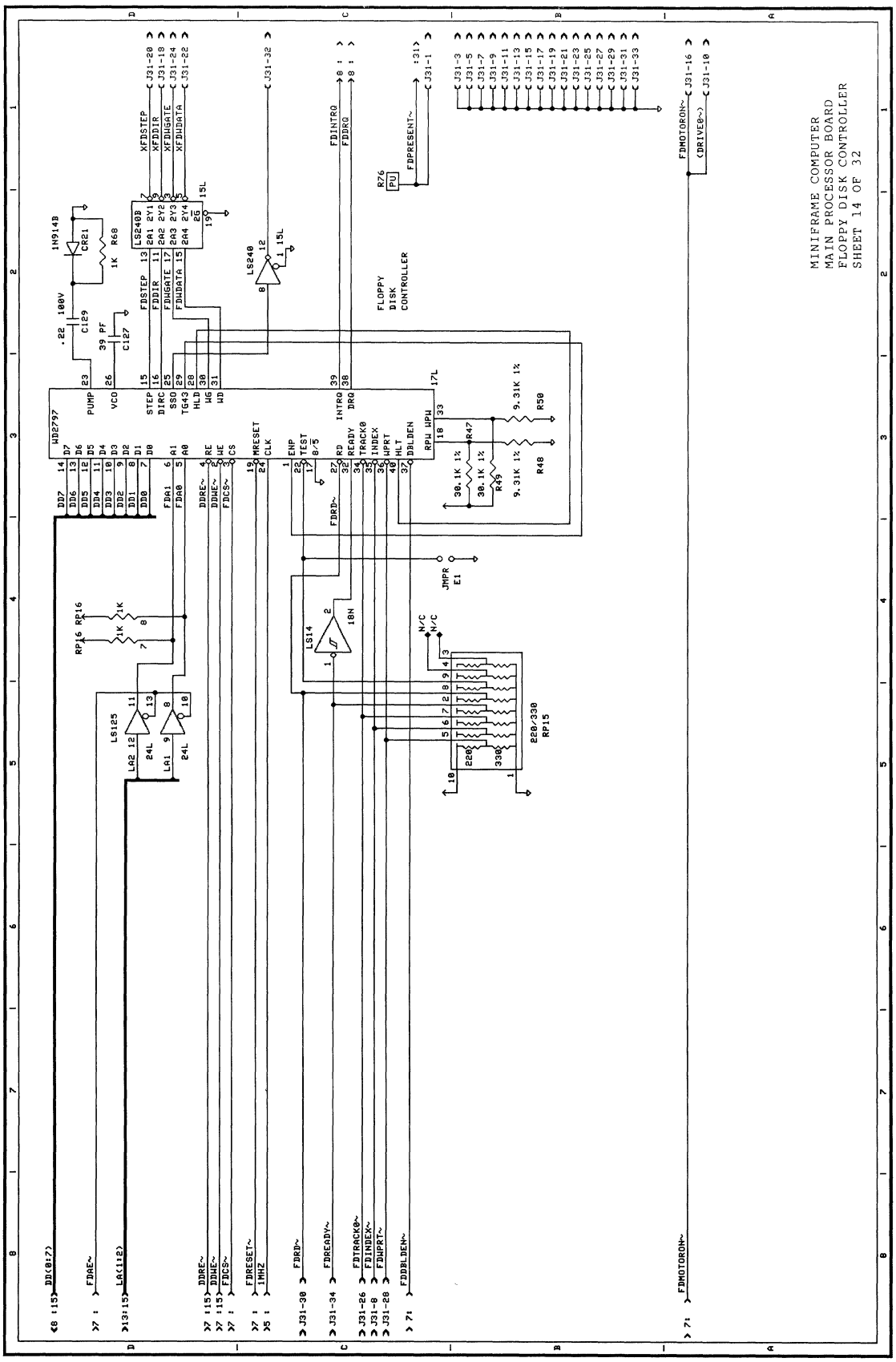
Sheet 11 of 32: Fast Communications Bus Interface Unit



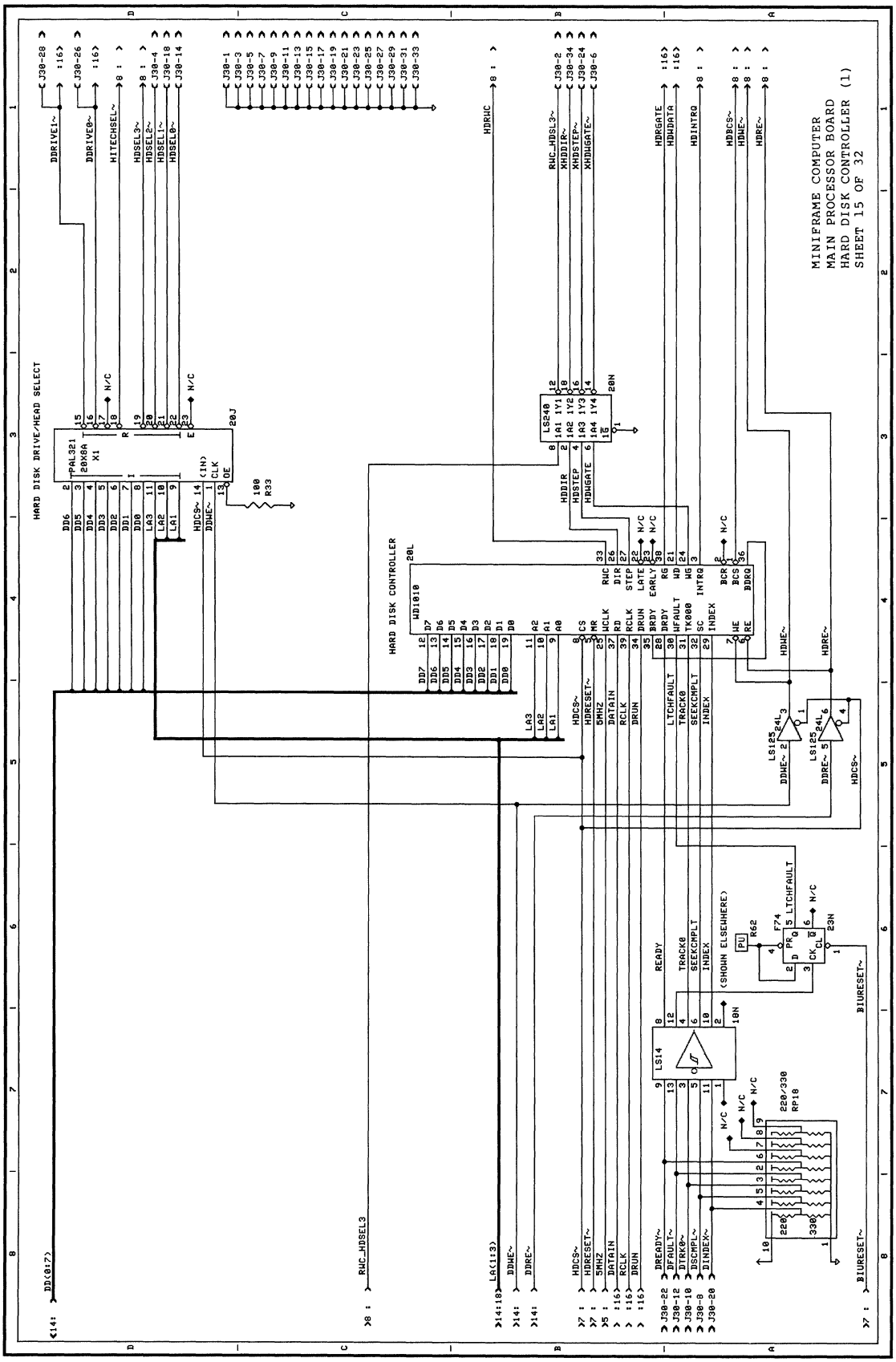
Sheet 12 of 32: Fast Communications DMA



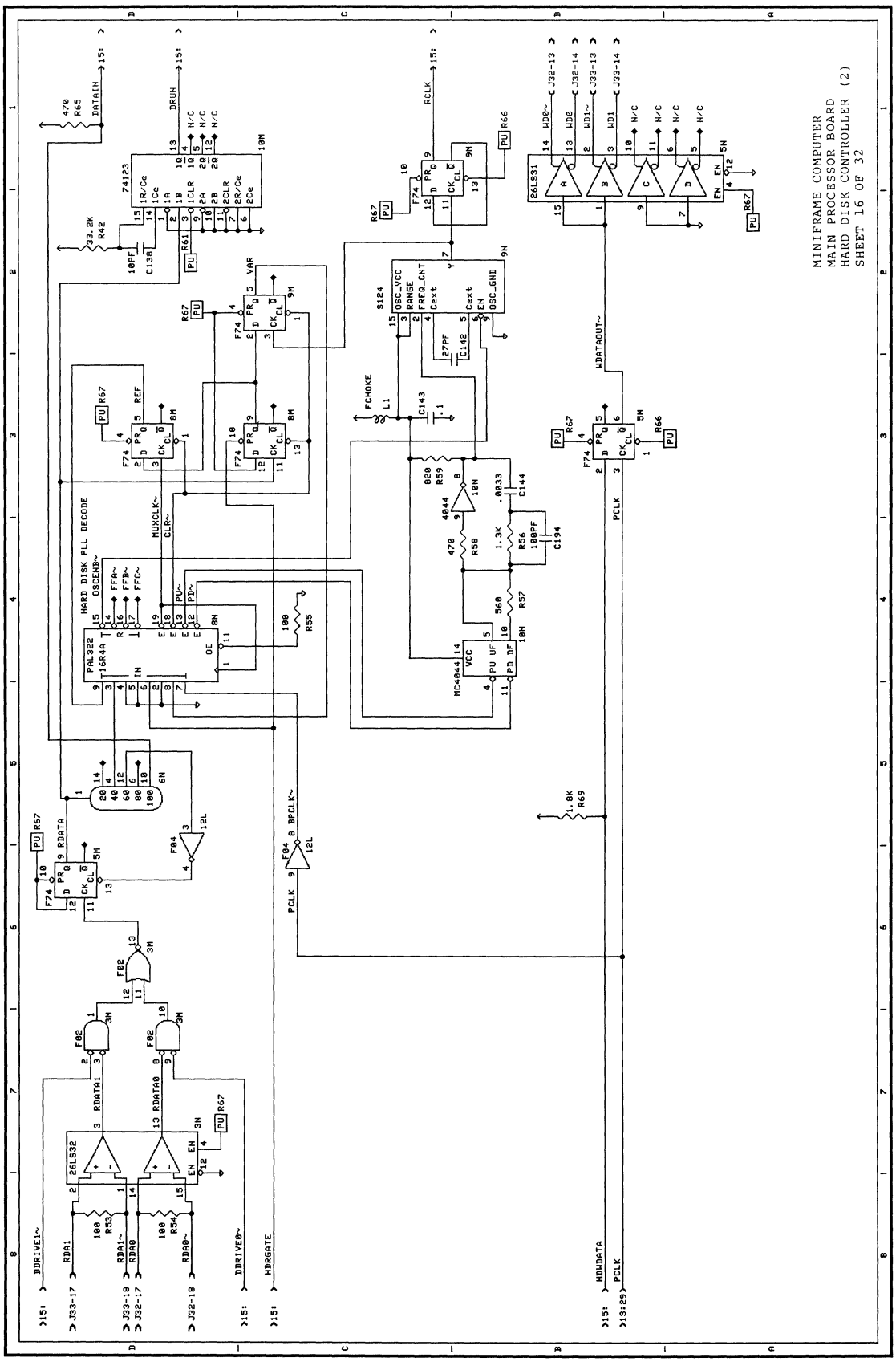
Sheet 13 of 32: Fast Communications Port



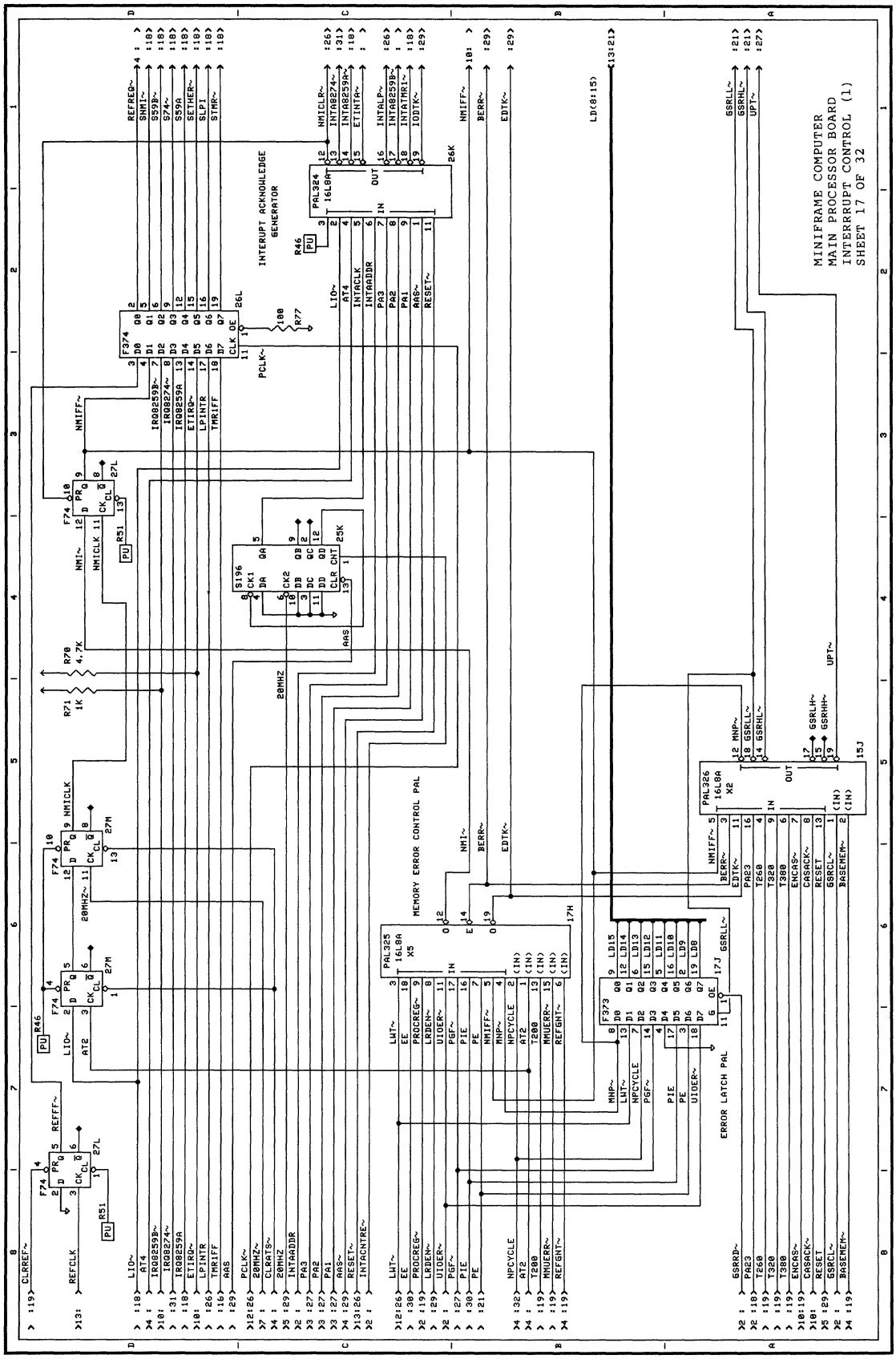
Sheet 14 of 32: Floppy Disk Controller



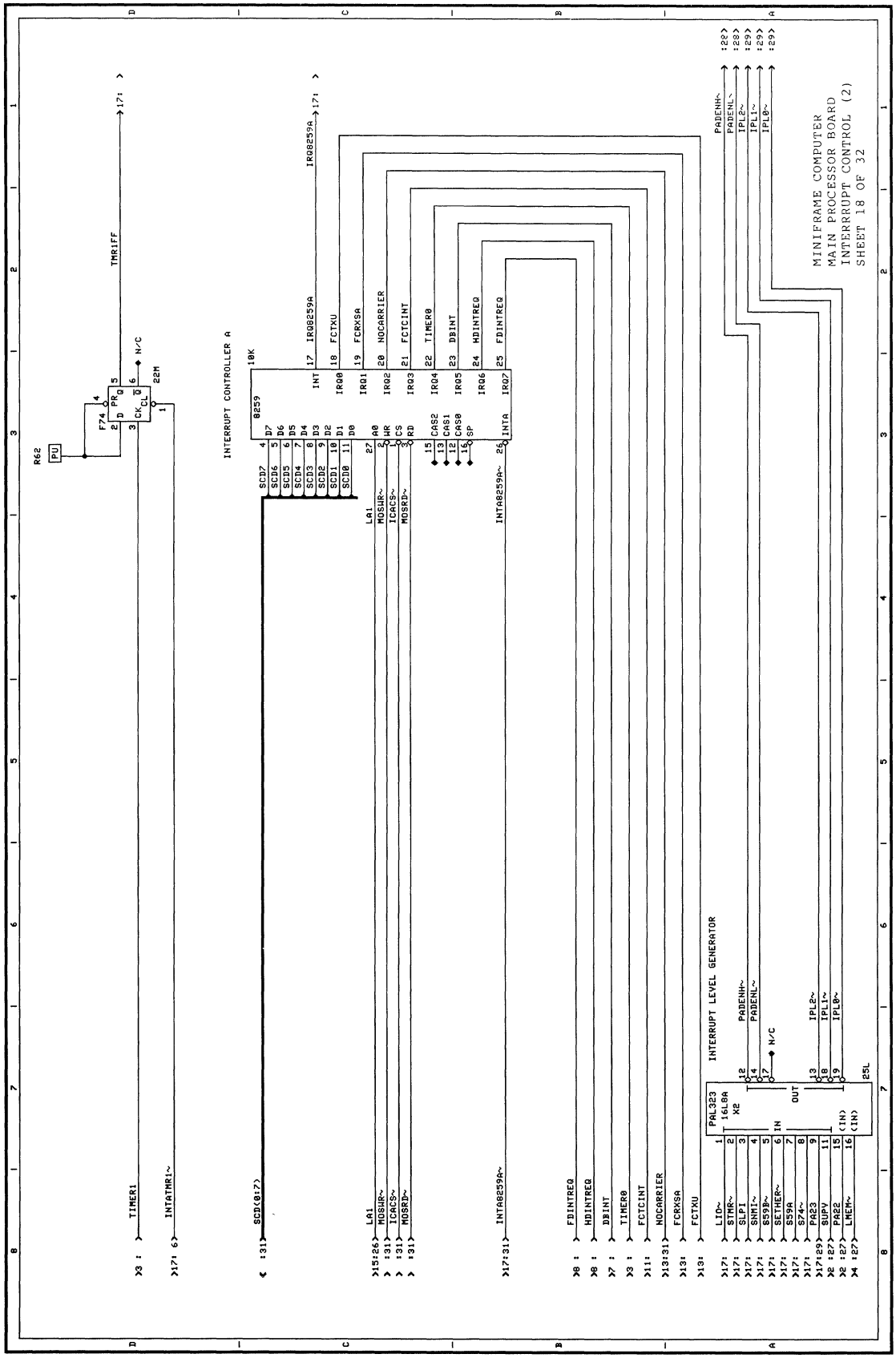
Sheet 15 of 32: Hard Disk Controller (1)



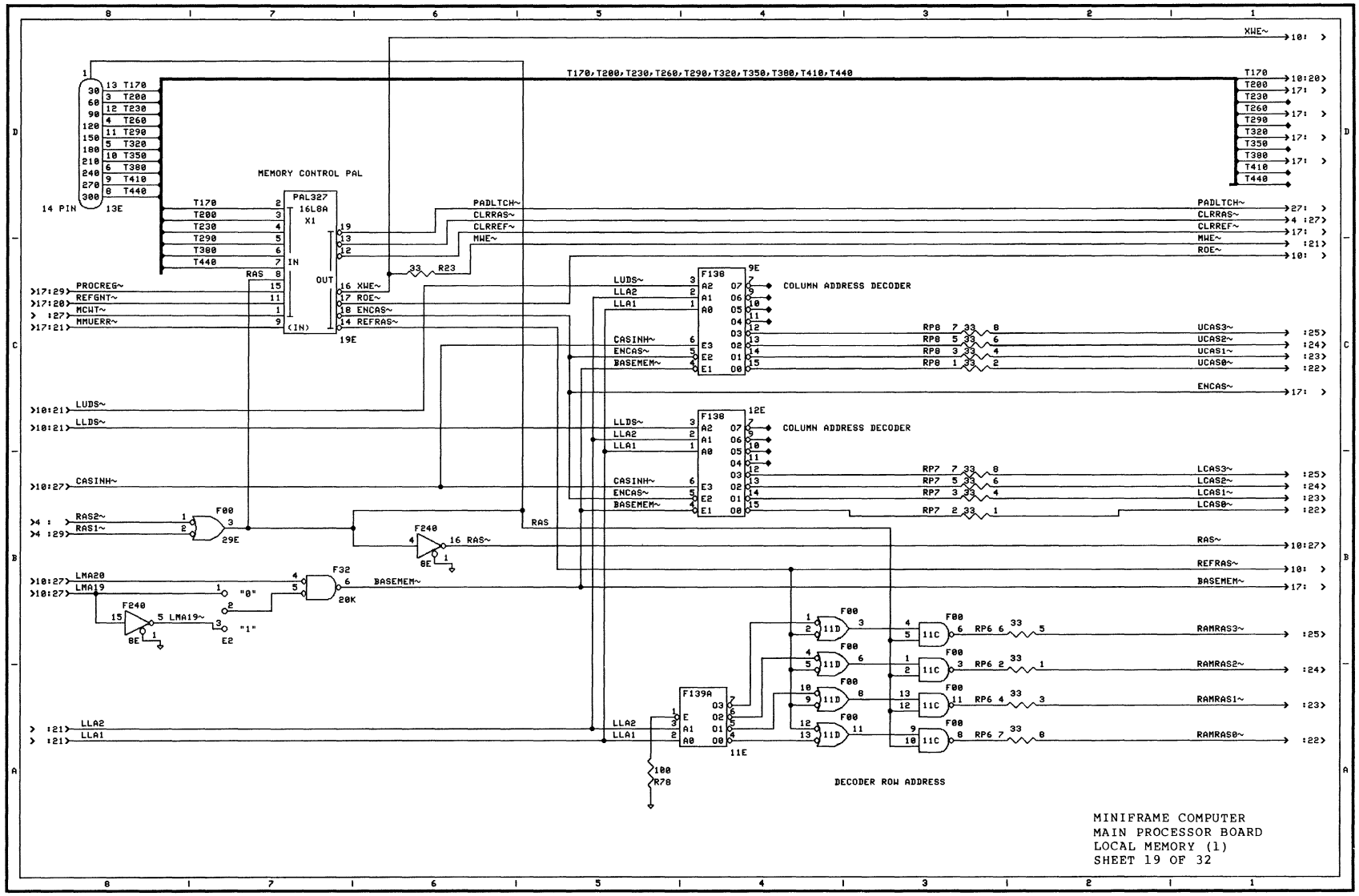
Sheet 16 of 32: Hard Disk Controller (2)



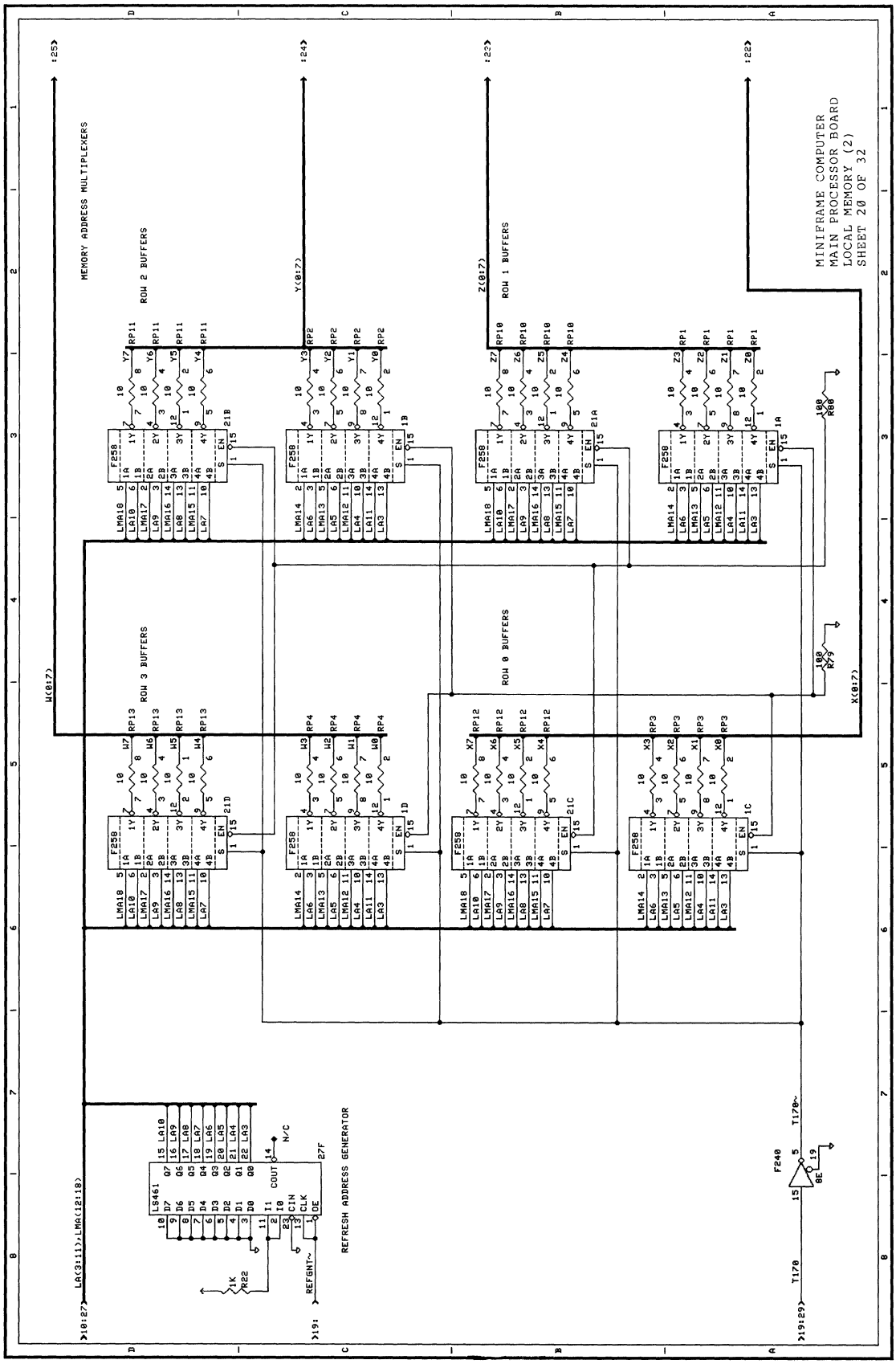
Sheet 17 of 32: Interrupt Control (1)



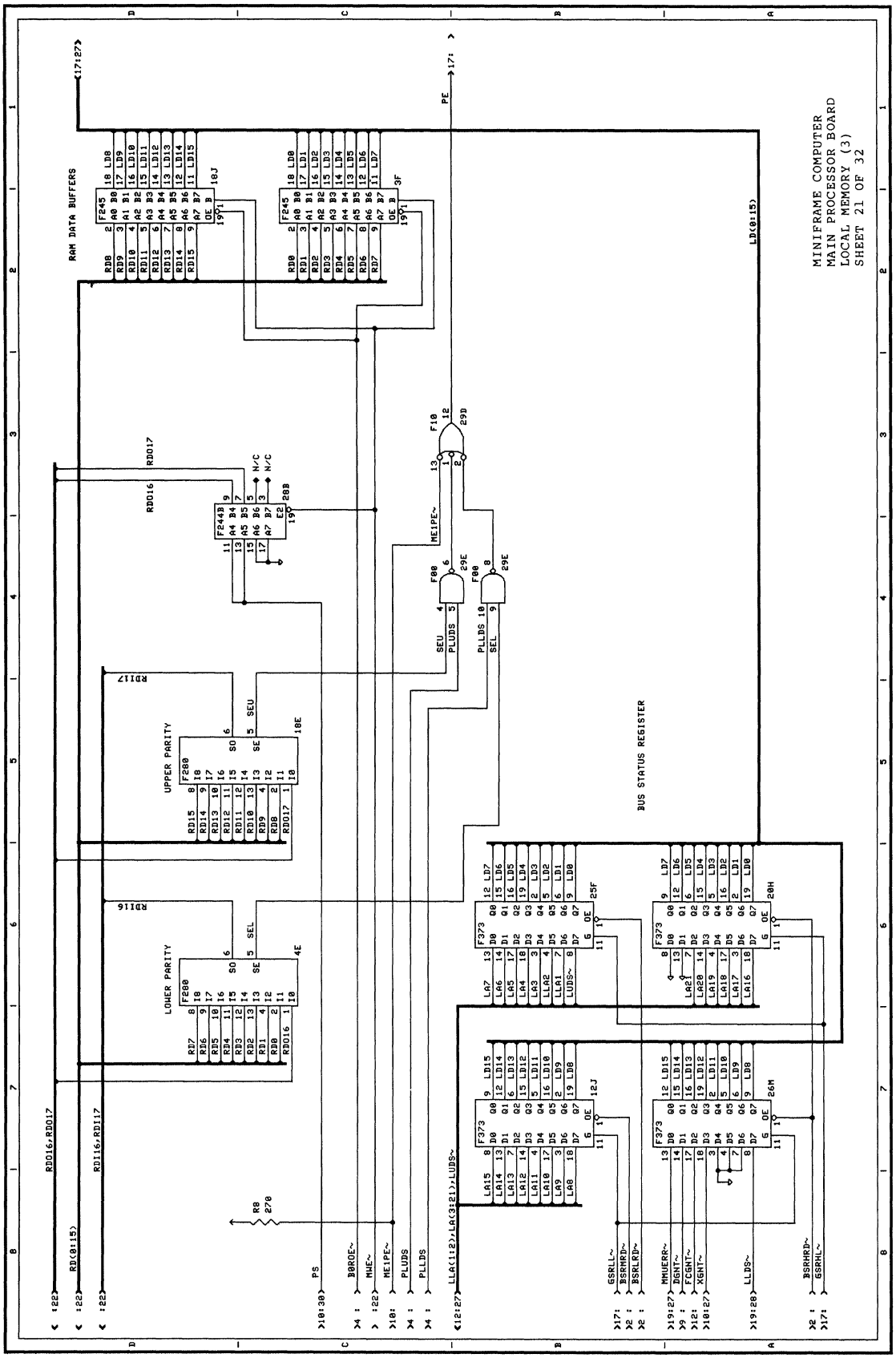
Sheet 18 of 32: Interrupt Control (2)



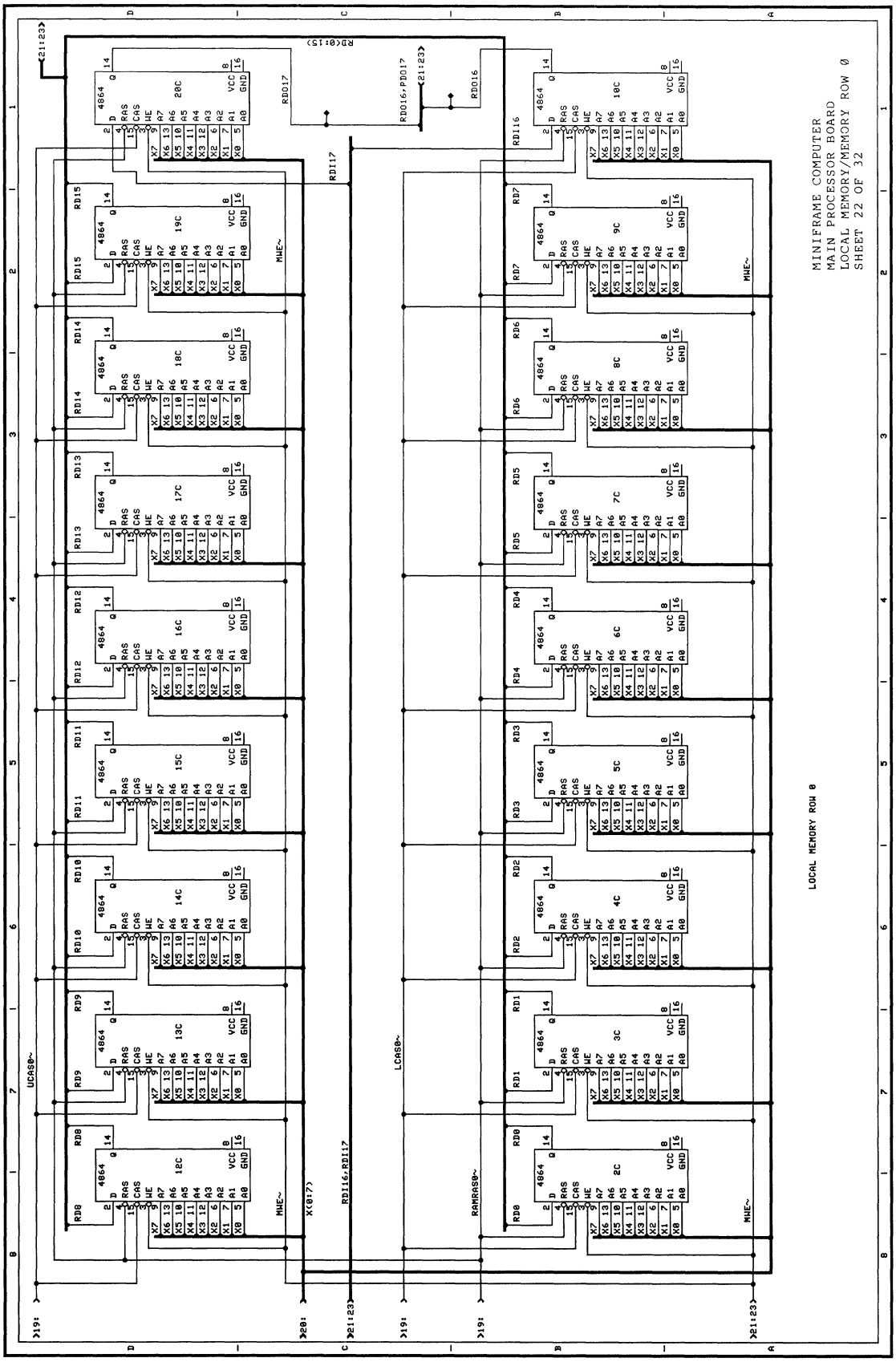
Sheet 19 of 32: Local Memory (1)



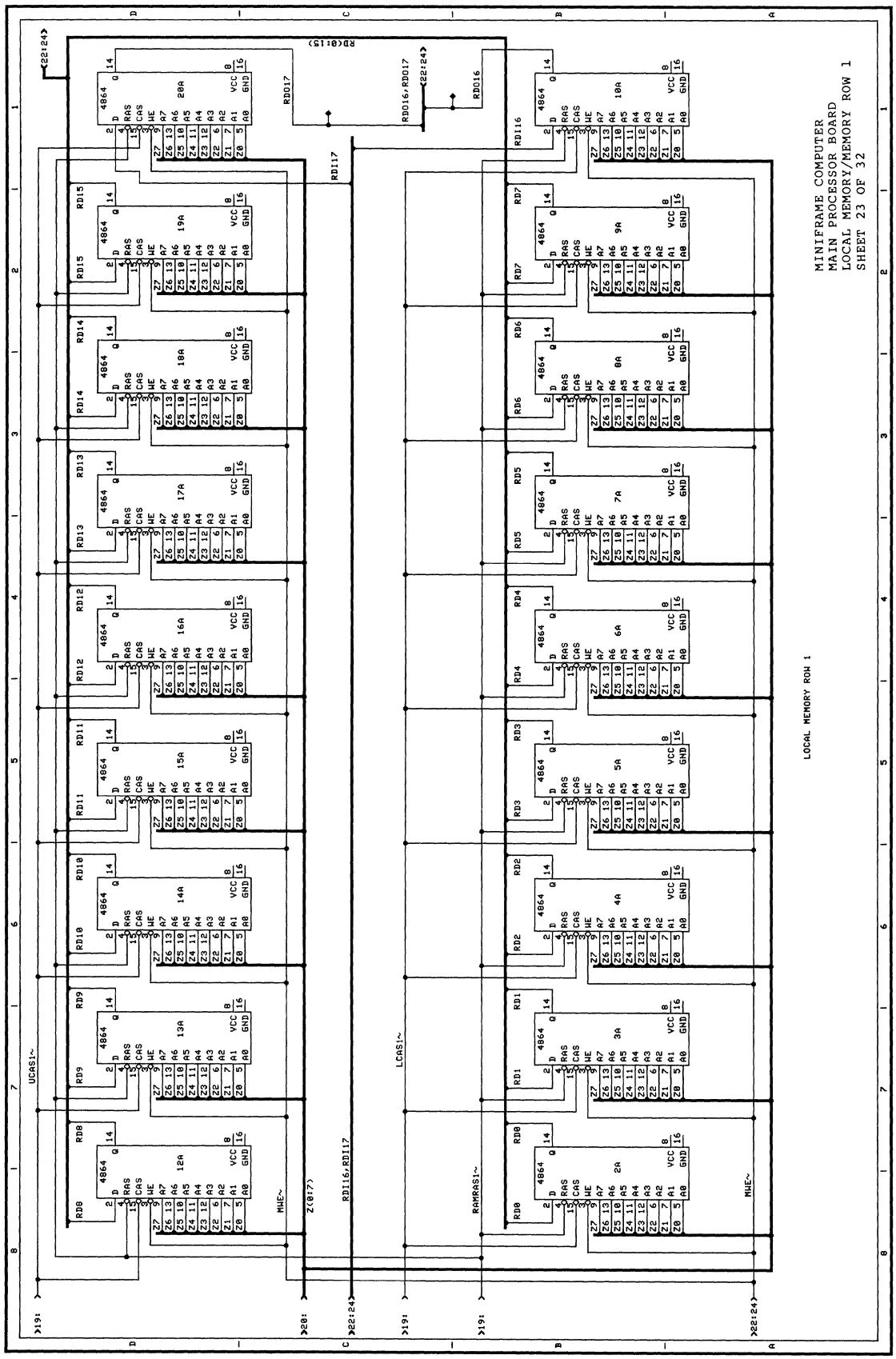
Sheet 20 of 32: Local Memory (2)



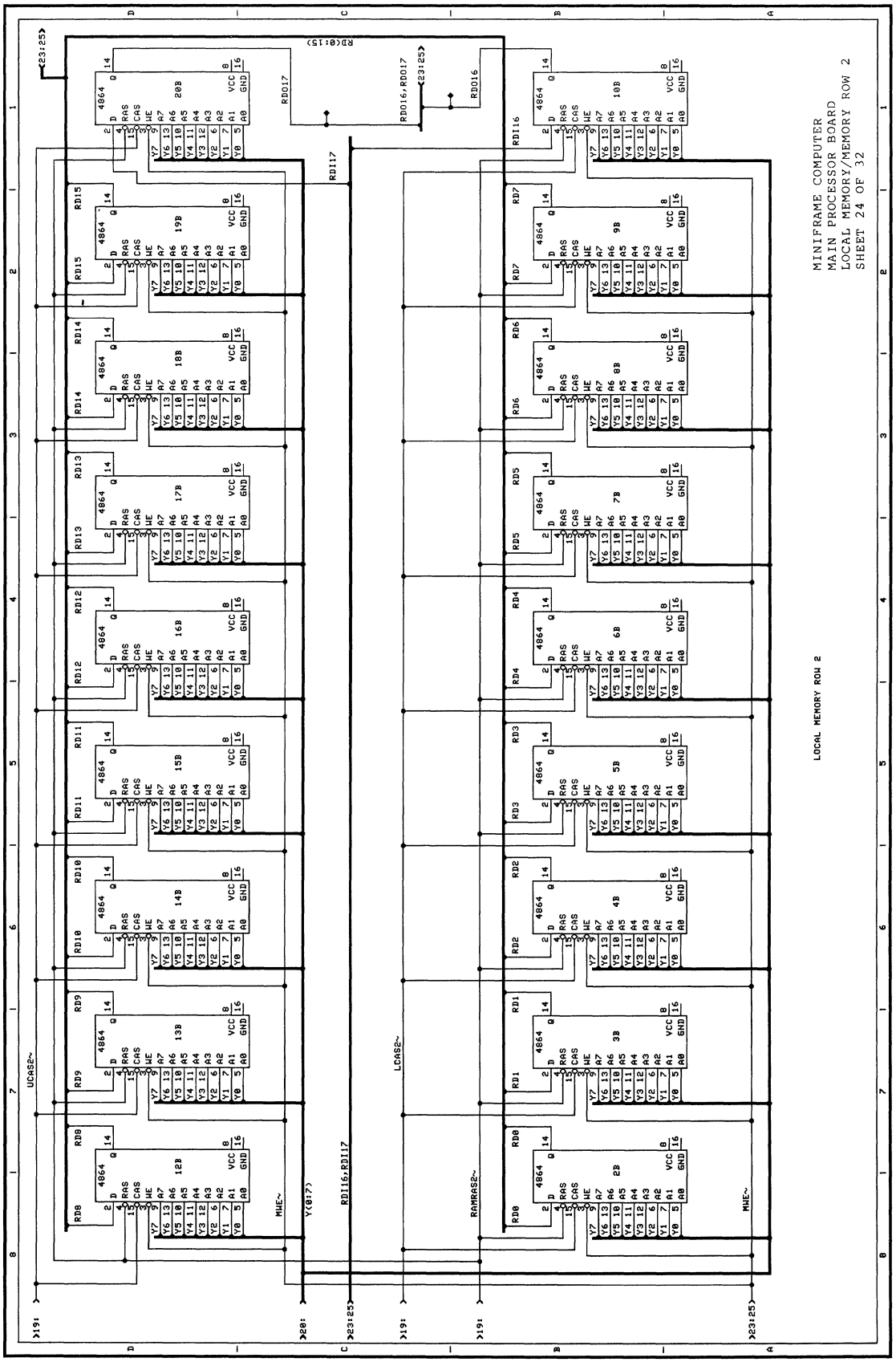
Sheet 21 of 32: Local Memory (3)



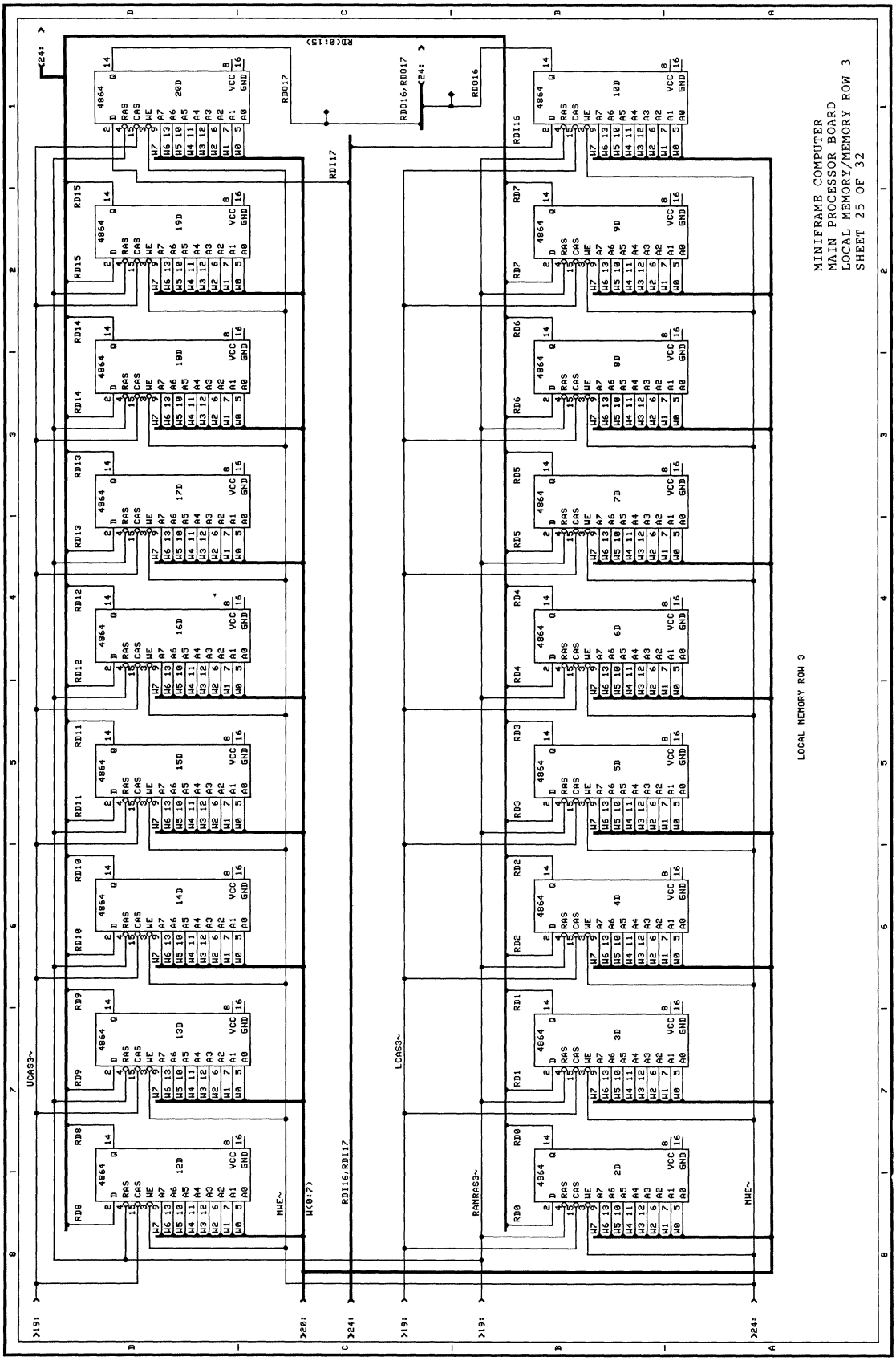
Sheet 22 of 32: Local Memory/Memory Row Ø



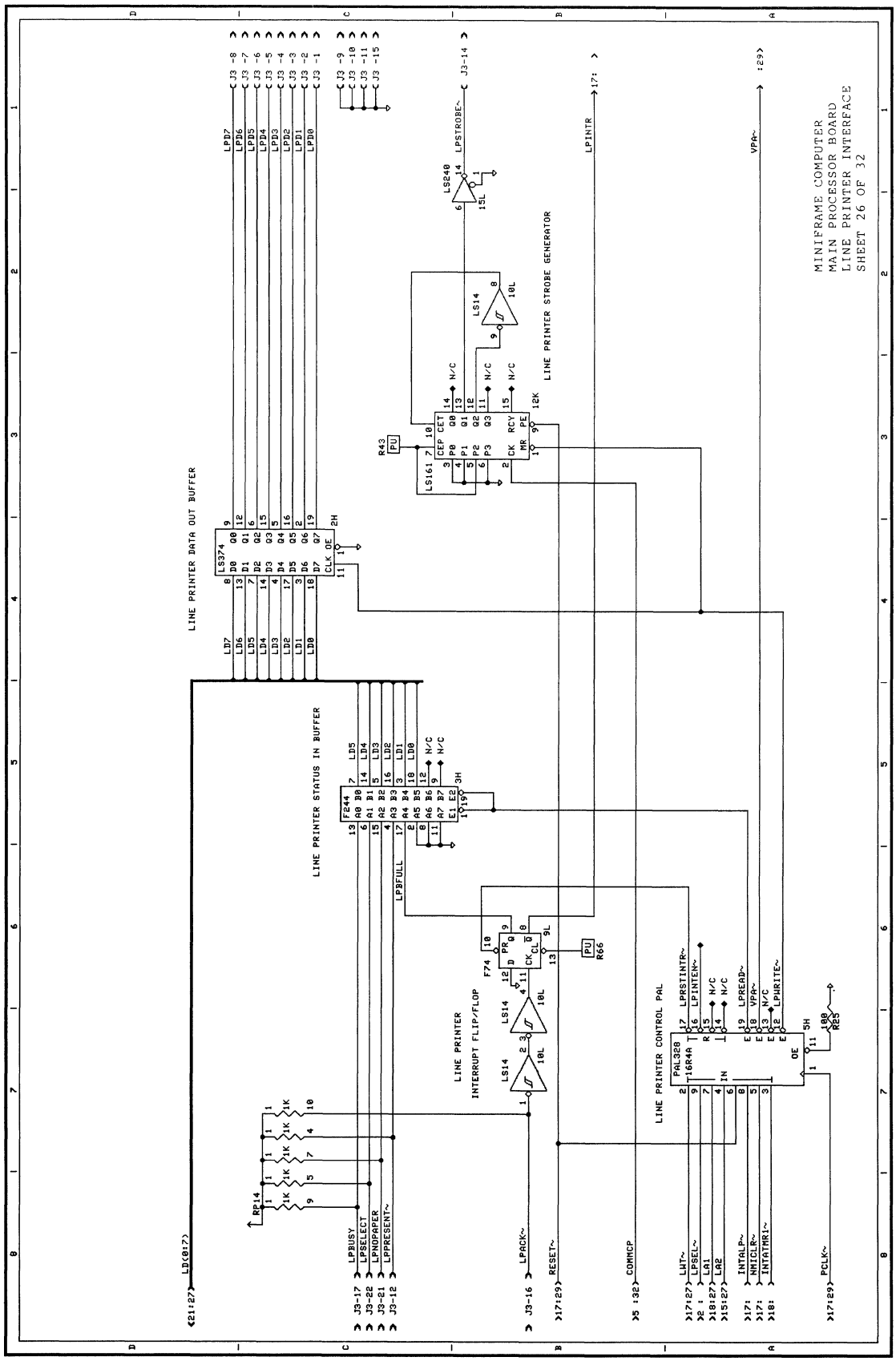
Sheet 23 of 32: Local Memory/Memory Row 1



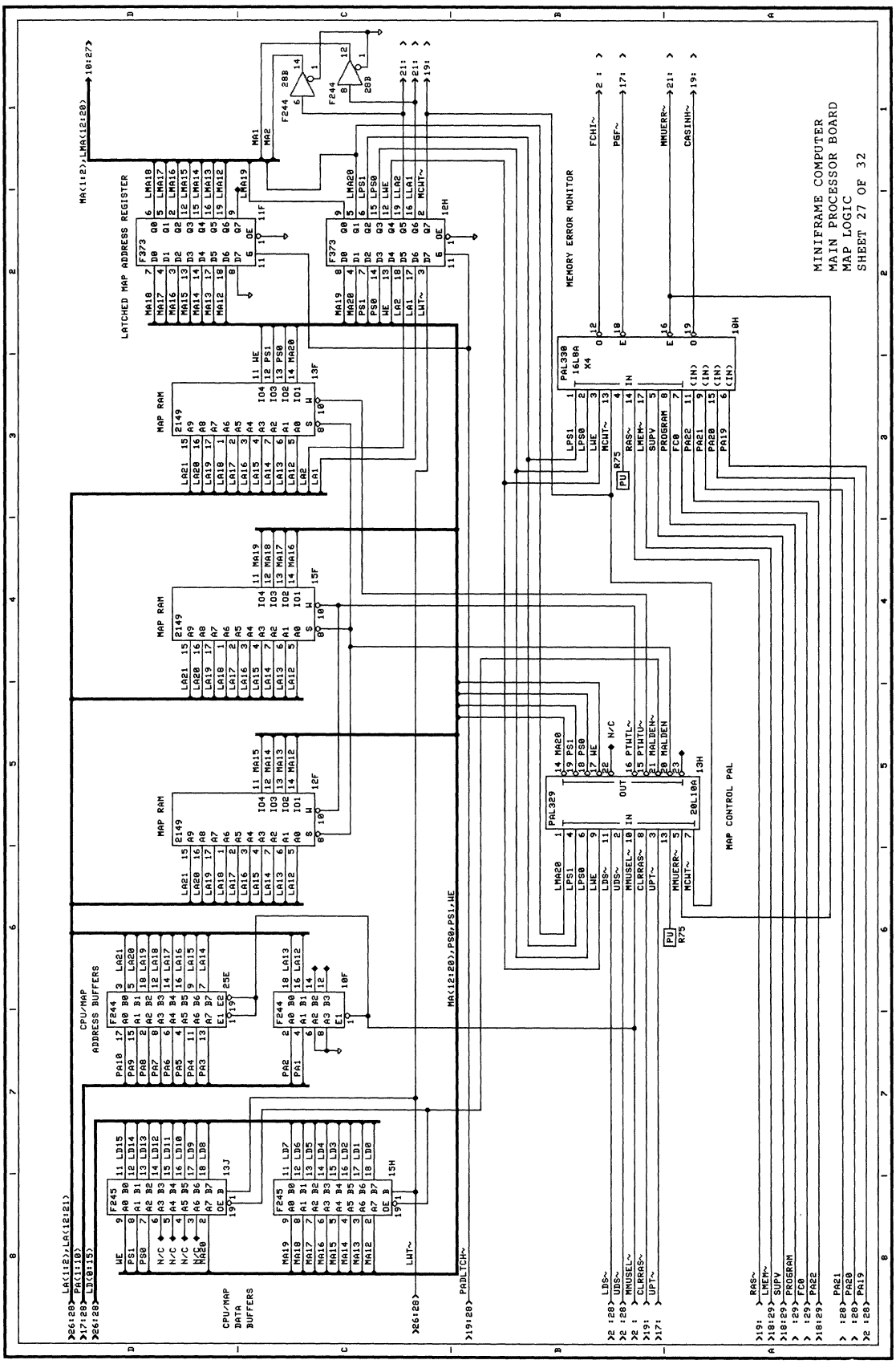
Sheet 24 of 32: Local Memory/Memory Row 2



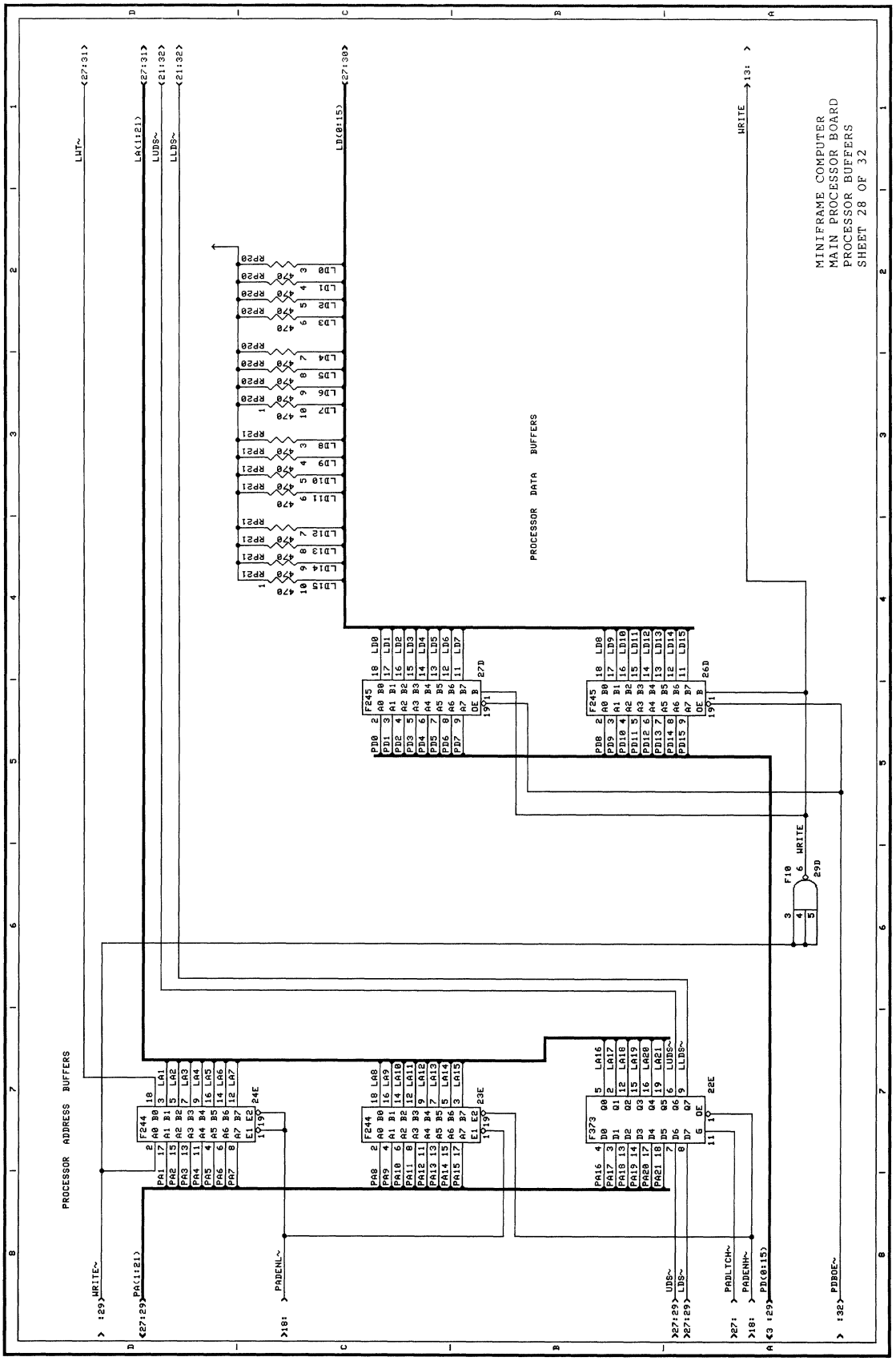
Sheet 25 of 32: Local Memory/Memory Row 3



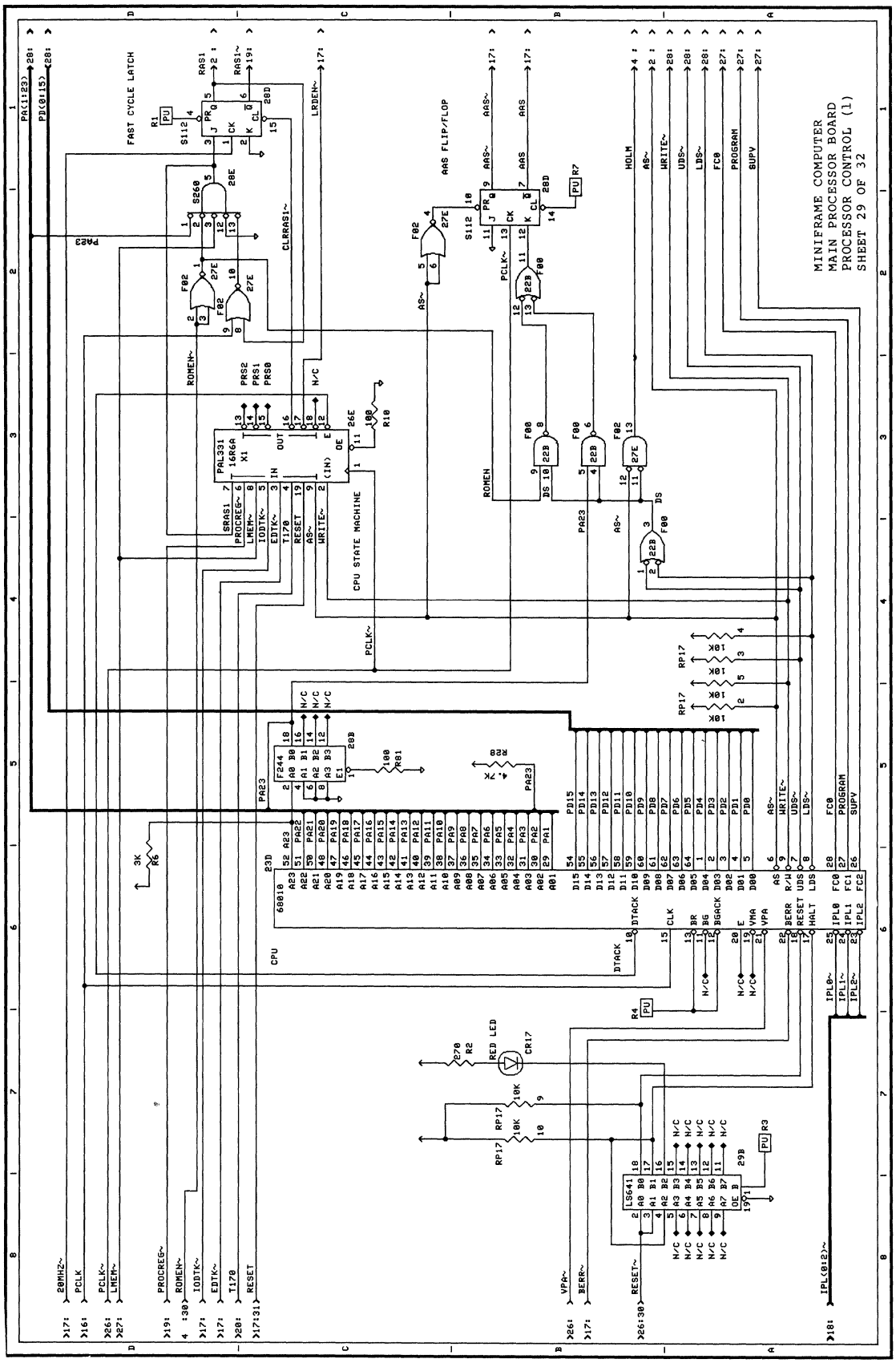
Sheet 26 of 32: Line Printer Interface



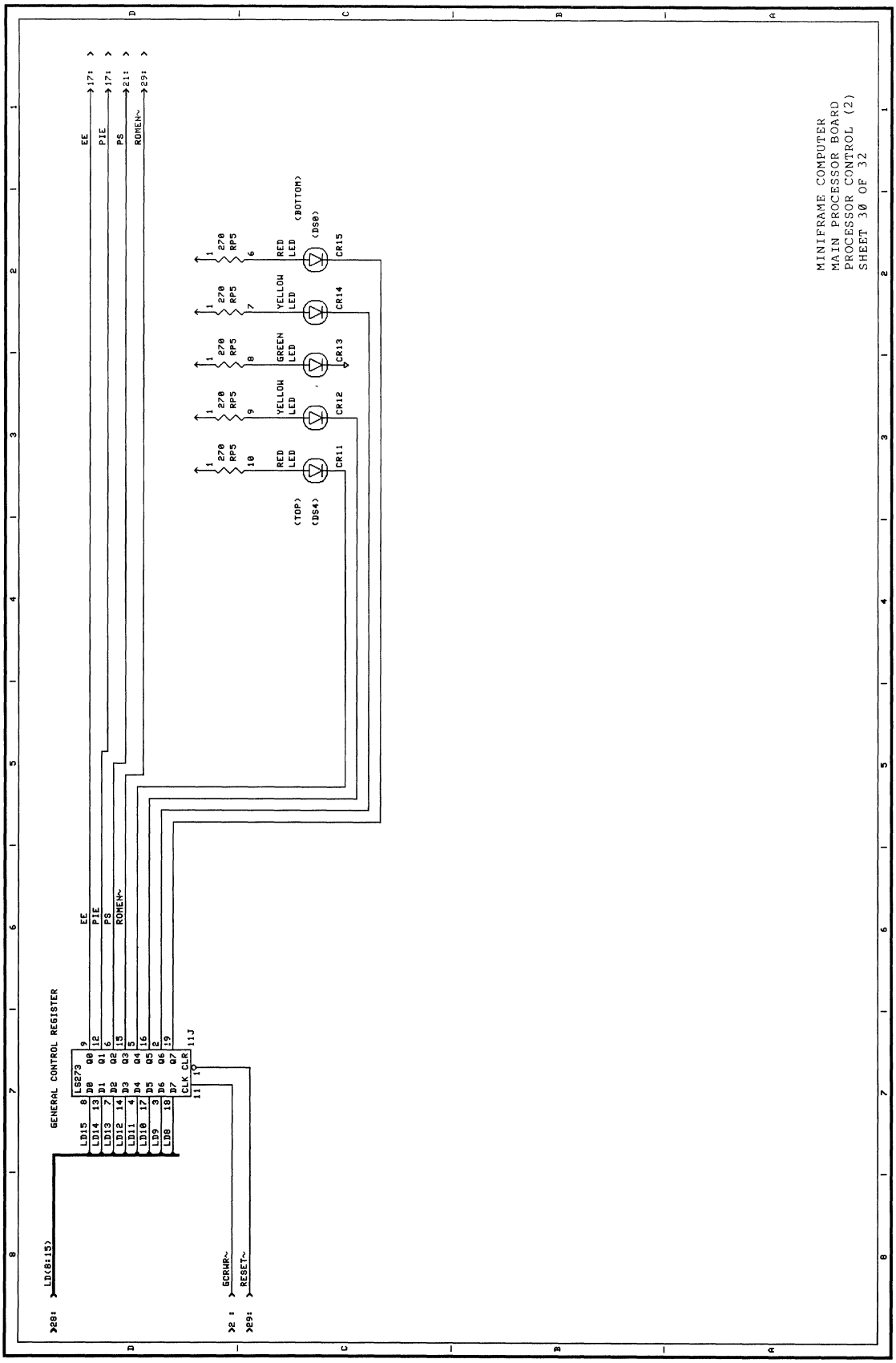
Sheet 27 of 32: Map Logic



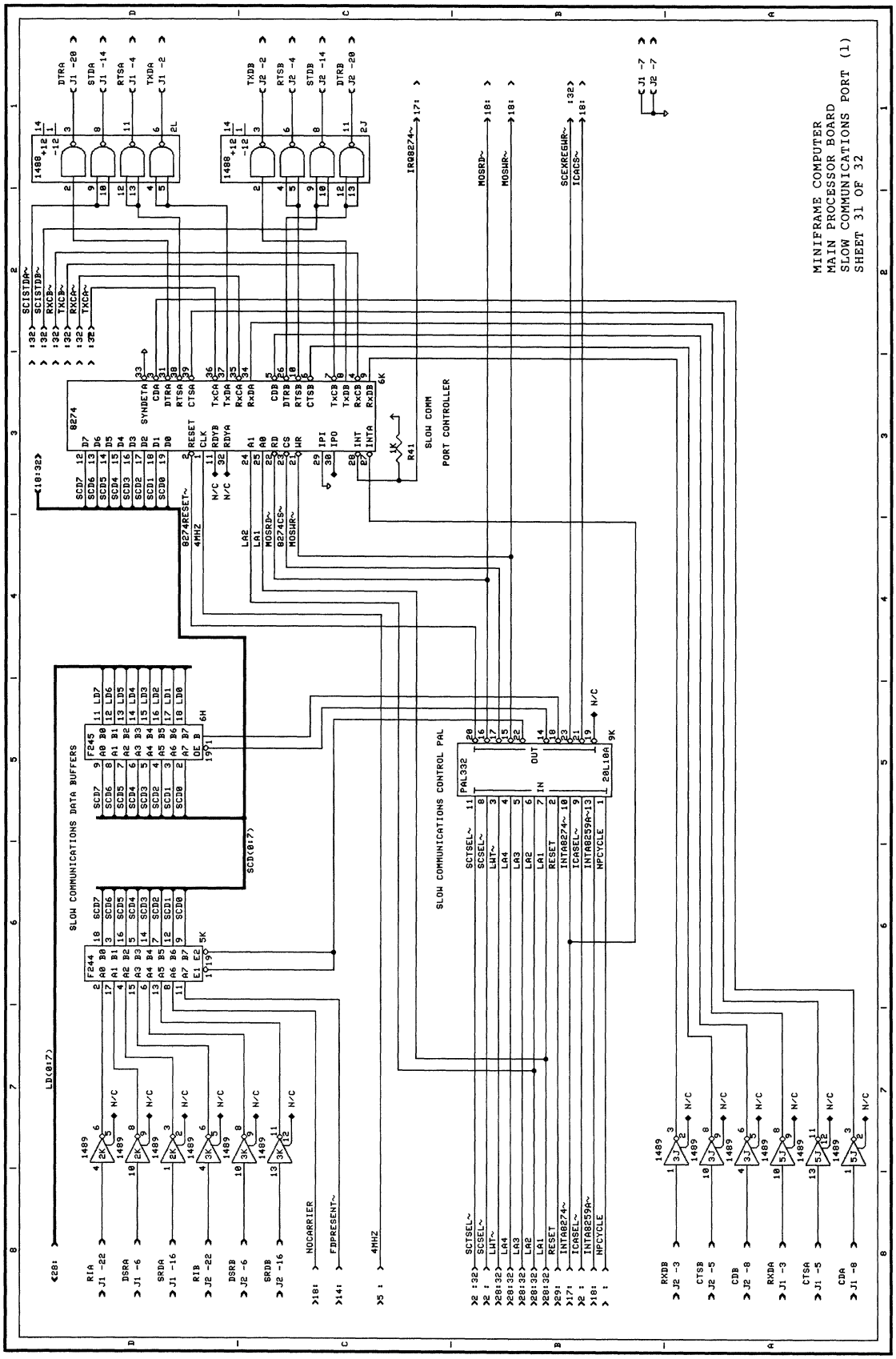
Sheet 28 of 32: Processor Buffers



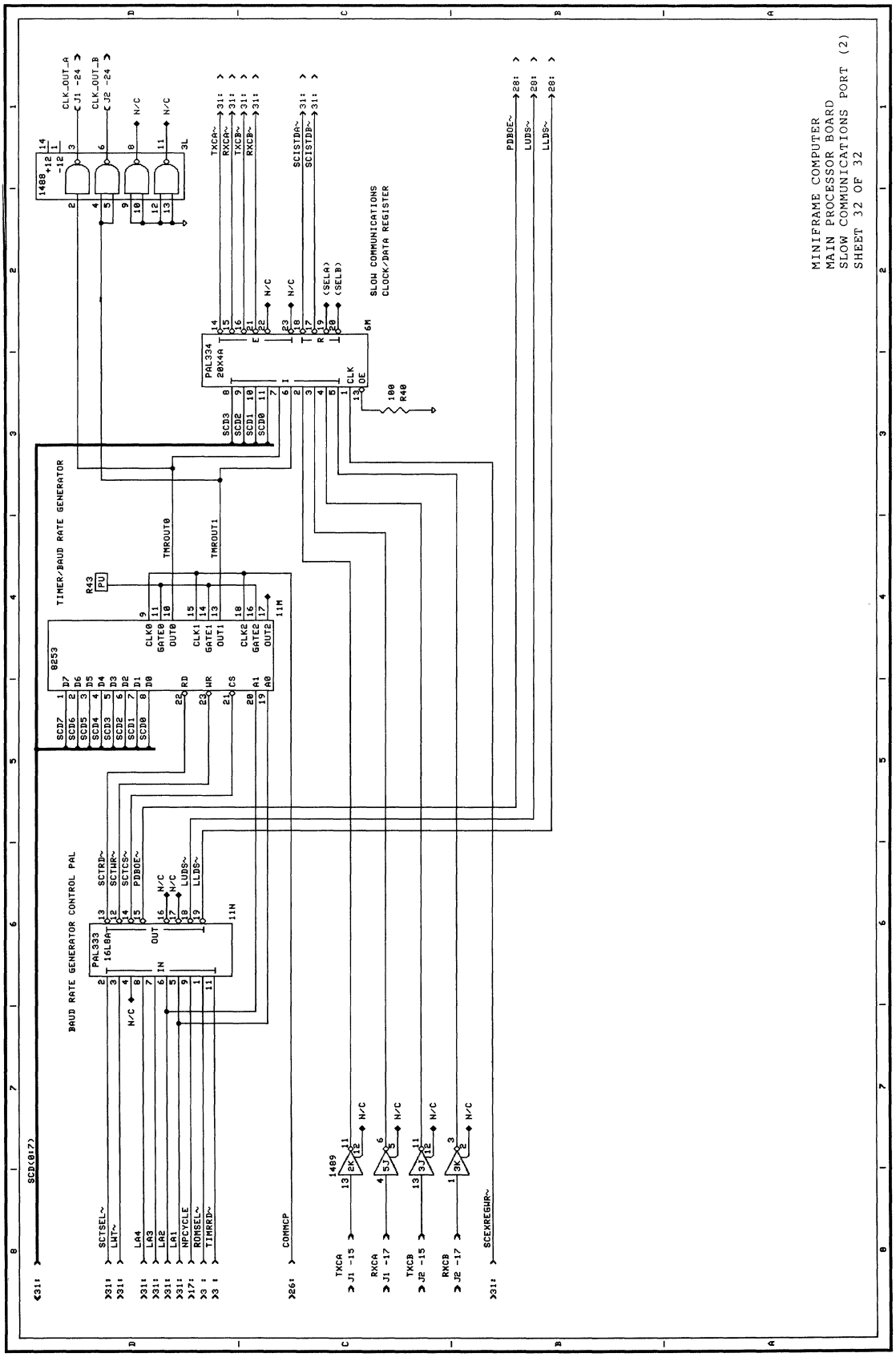
Sheet 29 of 32: Processor Control (1)



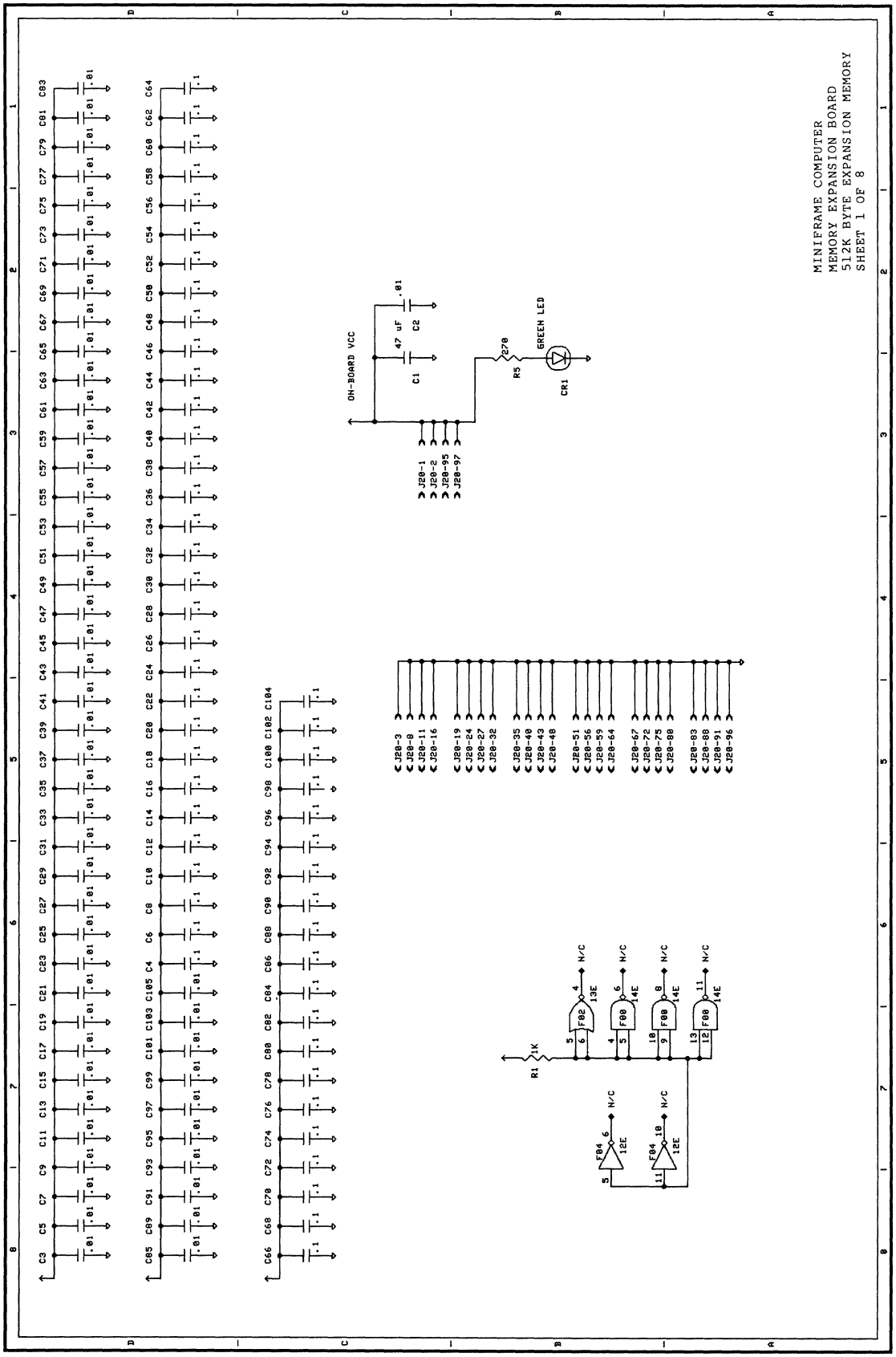
Sheet 30 of 32: Processor Control (2)



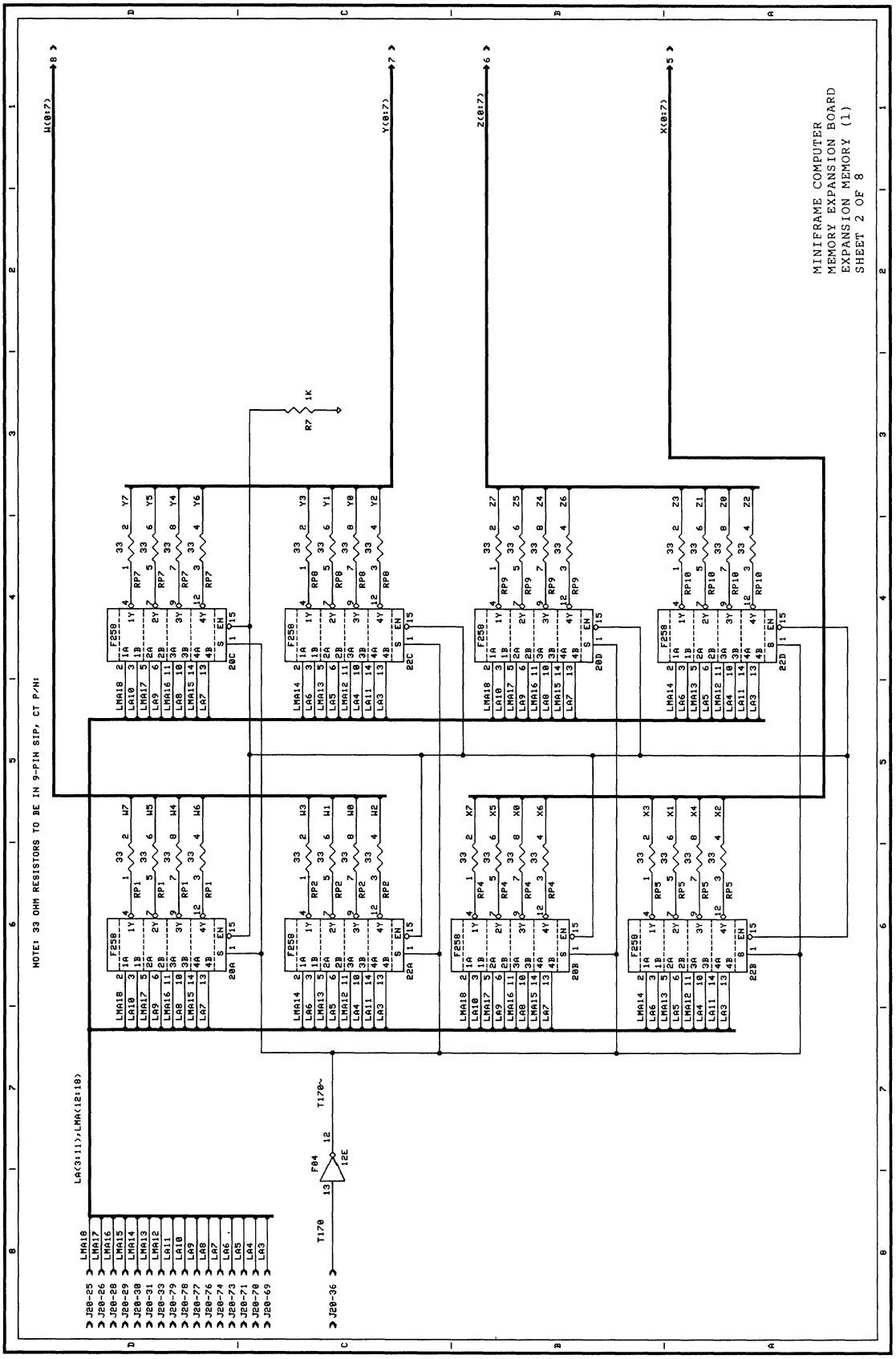
Sheet 31 of 32: Slow Communications Port (1)



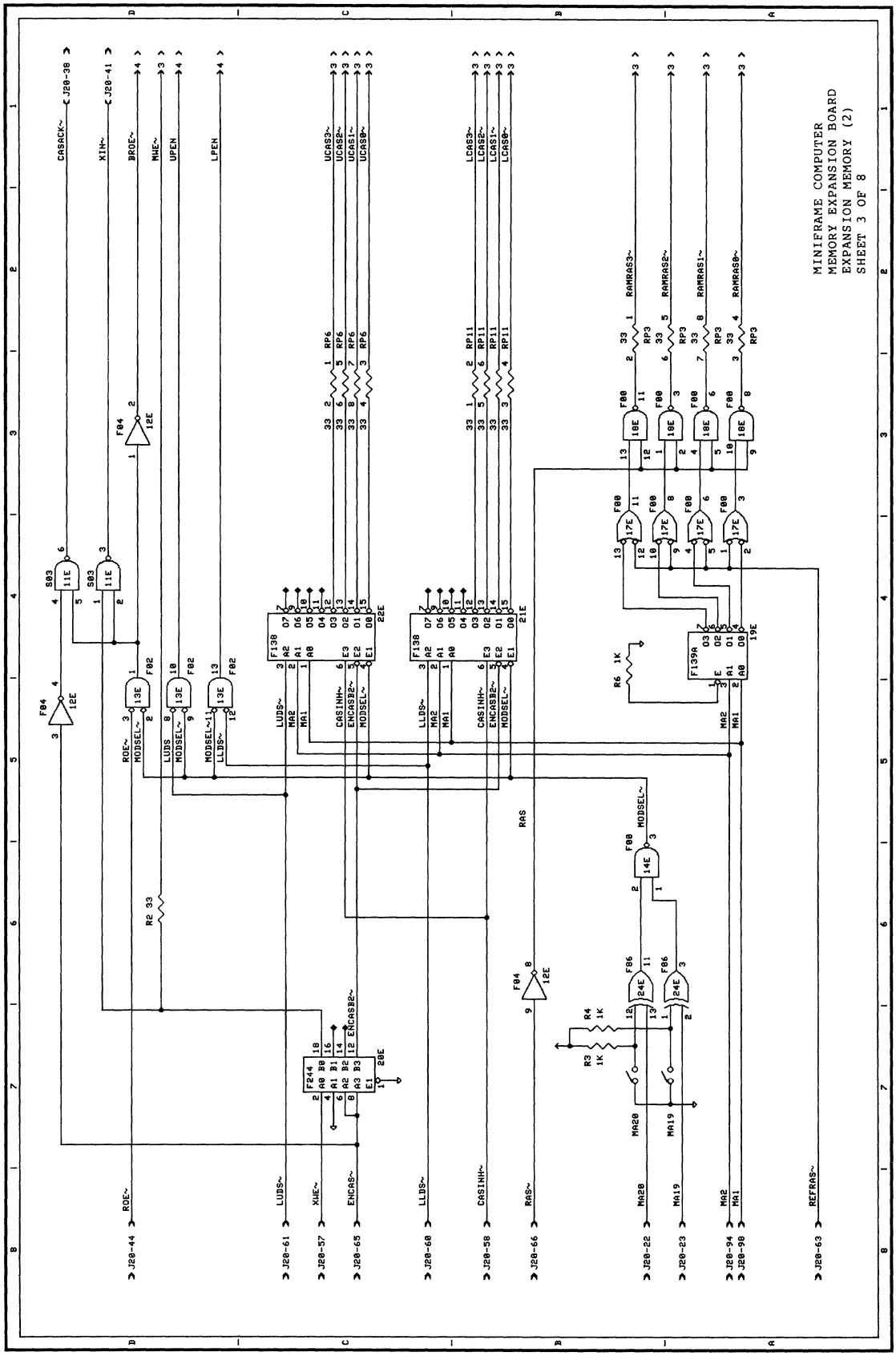
Sheet 32 of 32: Slow Communications Port (2)



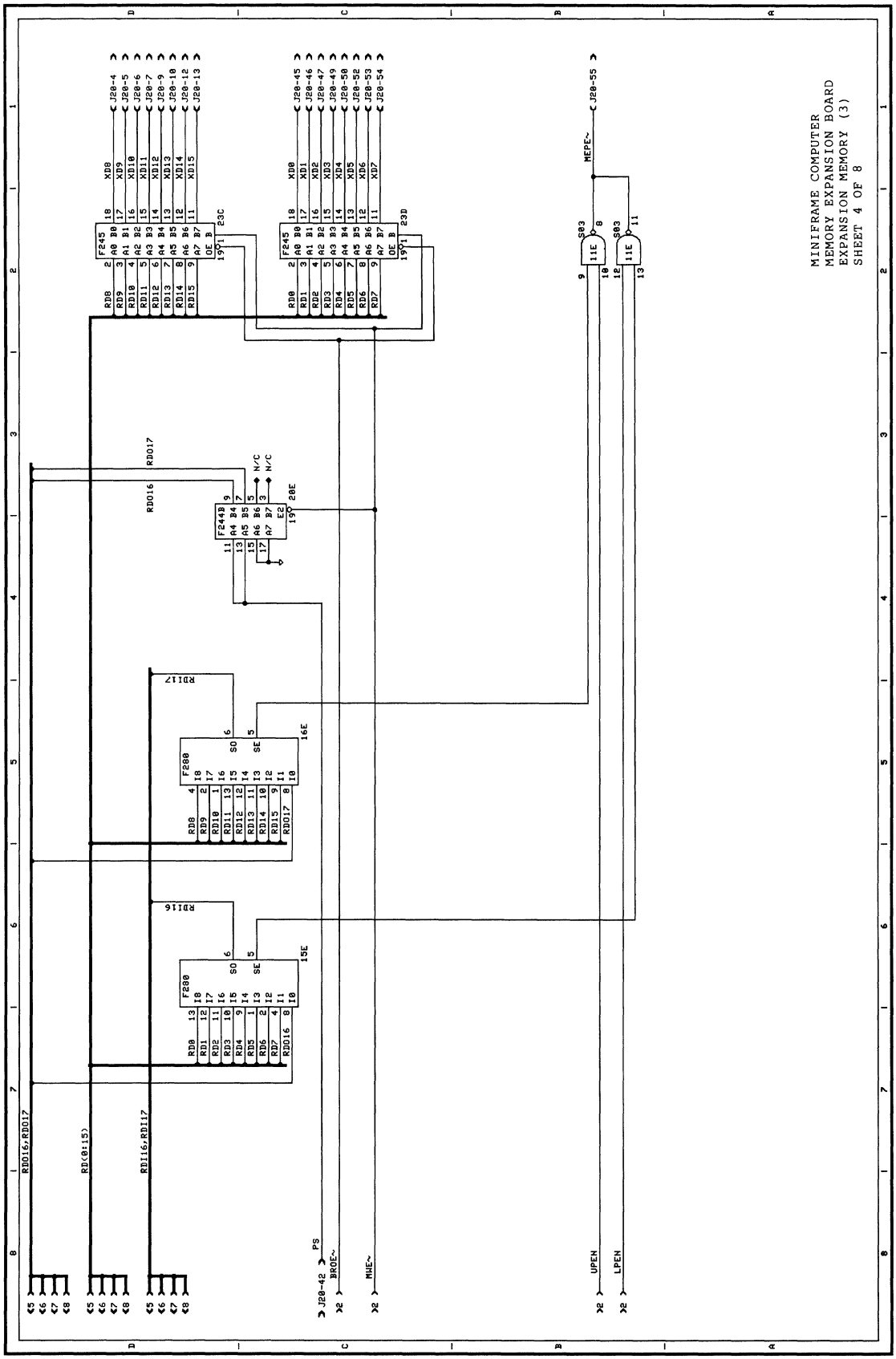
Sheet 1 of 8: 512 Kbyte Expansion Memory



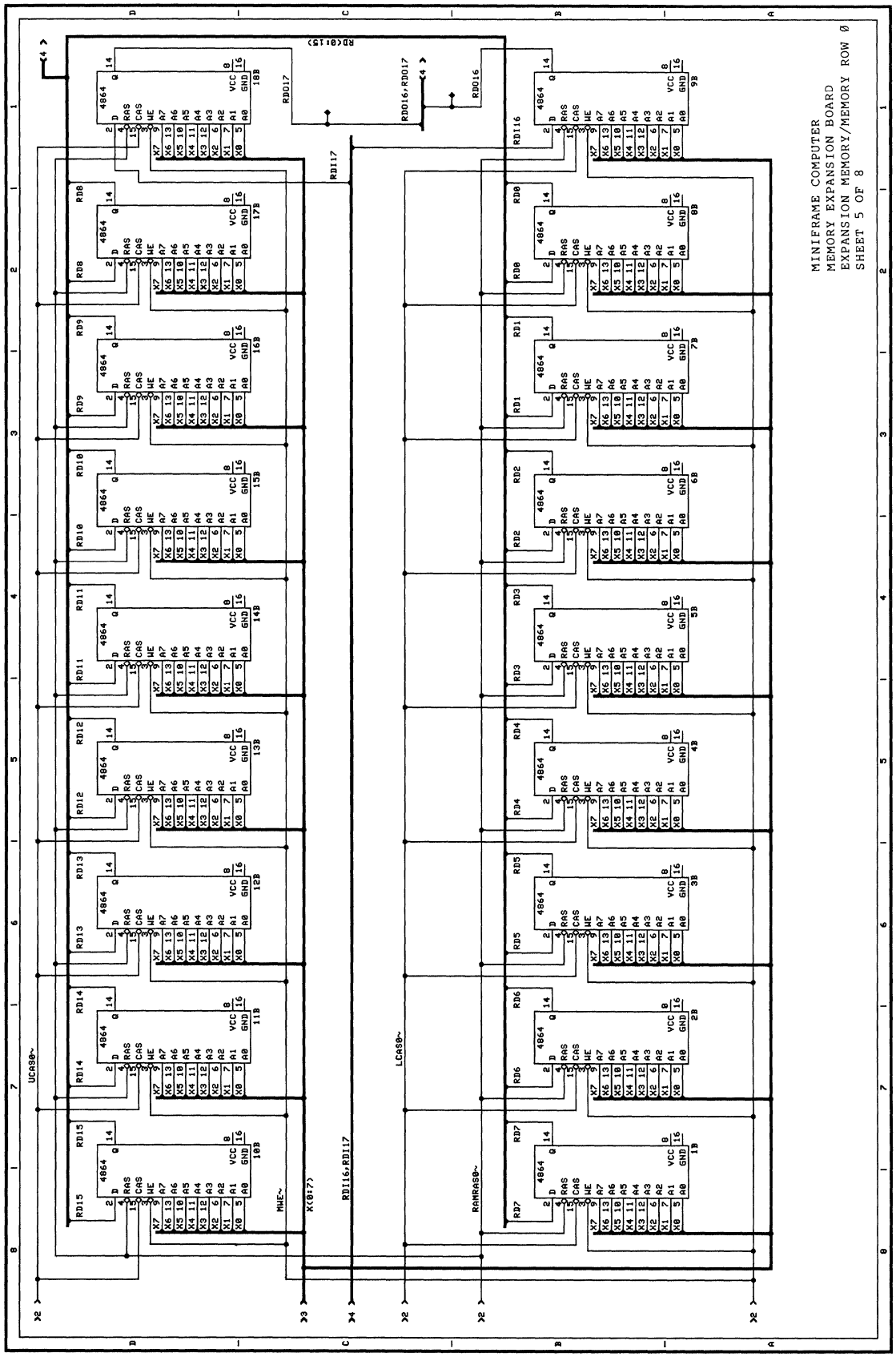
Sheet 2 of 8: Expansion Memory (1)



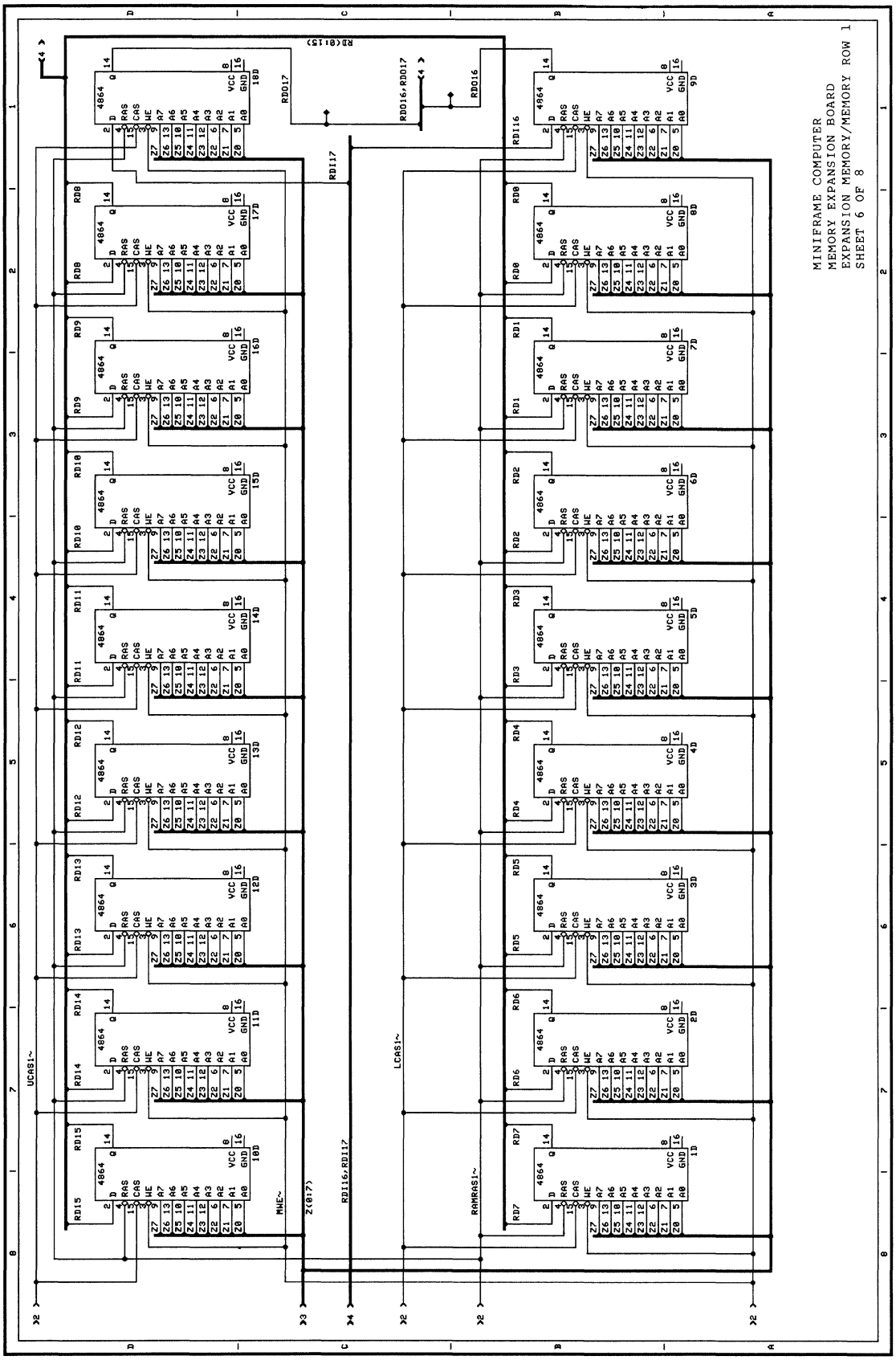
Sheet 3 of 8: Expansion Memory (2)



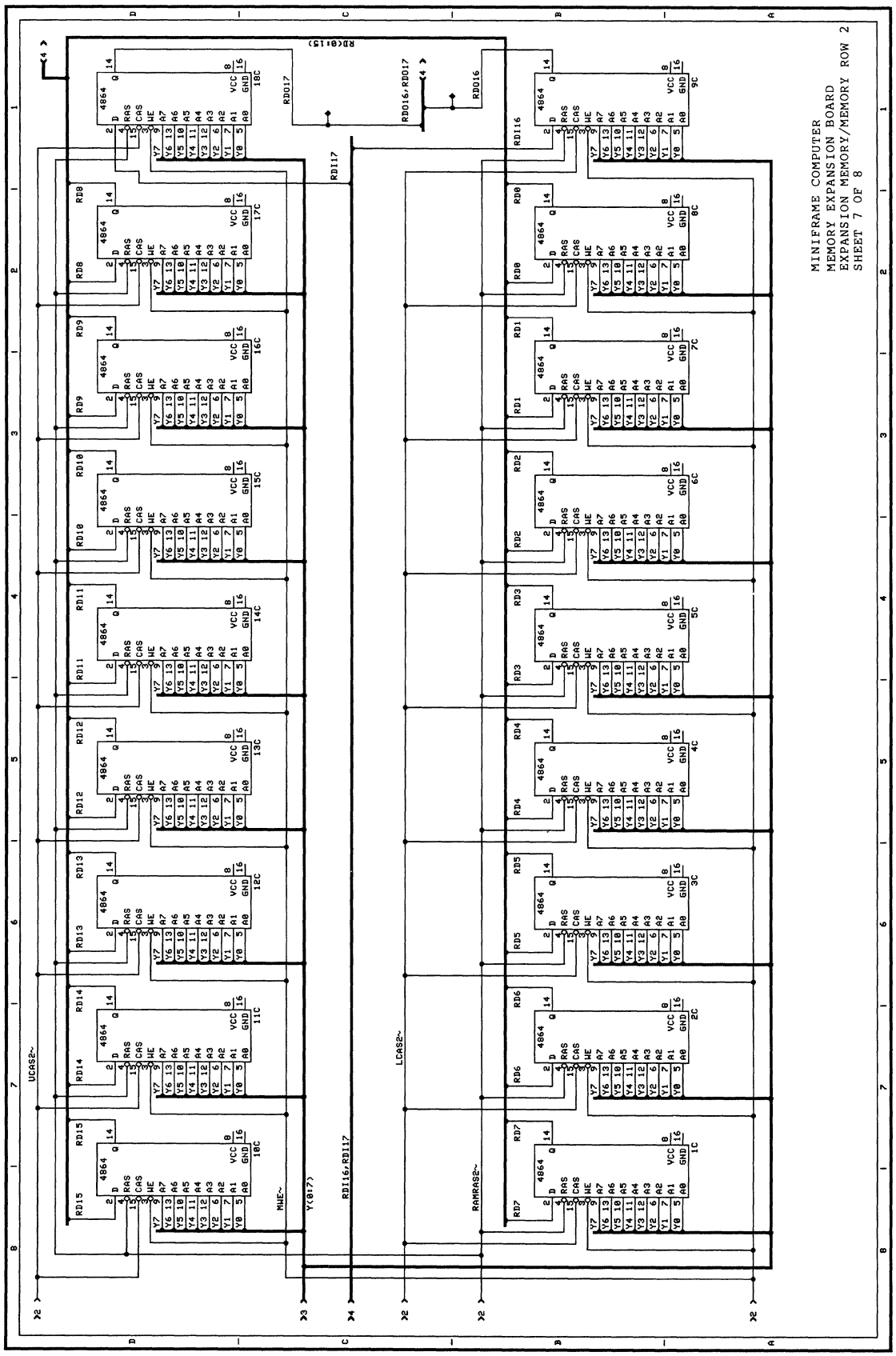
Sheet 4 of 8: Expansion Memory (3)



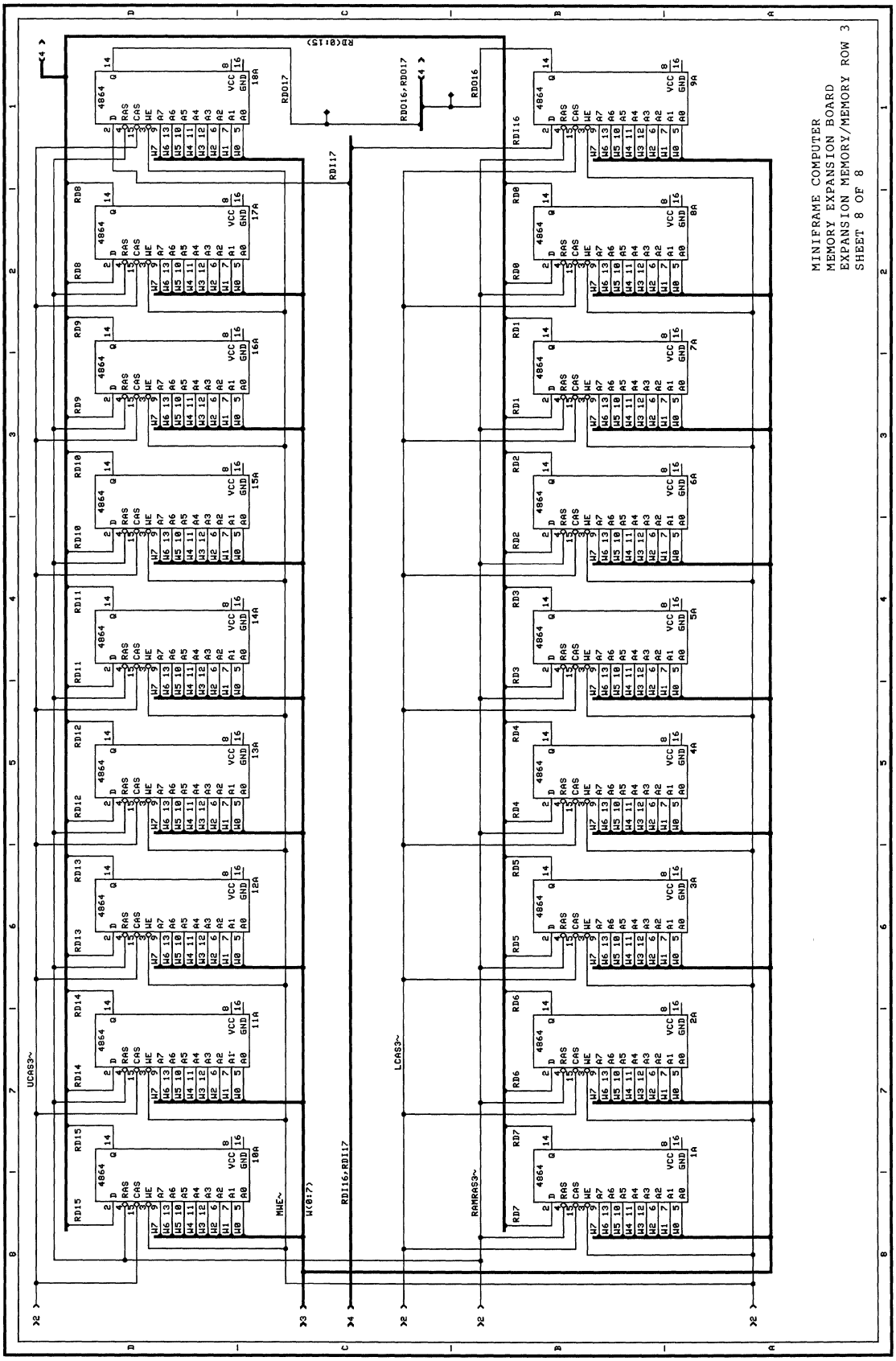
Sheet 5 of 8: Expansion Memory/Memory Row Ø



Sheet 6 of 8: Expansion Memory/Memory Row 1



Sheet 7 of 8: Expansion Memory/Memory Row



Sheet 8 of 8: Expansion Memory/Memory Row 3

APPENDIX C: POWER SUPPLY SPECIFICATIONS

This appendix provides information on the MiniFrame Computer power supply requirements, performance, and interface. The complete hardware for the power supply comprises a single unit, and the operating conditions apply to any part of the unit as well as to the unit as a whole.

AC INPUT REQUIREMENTS

Table C-1 lists the AC requirements of the power supply.

Table C-1. AC Power Requirements

Condition	Requirement
Input Voltage	115V Nominal (85 to 130V RMS) 230V Nominal (180 to 260V RMS)
Input Voltage Frequency	115/230VAC 47 to 63 Hz (continuous)
Surge Current	115VAC: 20 Amps (maximum for two cycles) 230VAC: 40 Amps (maximum for two cycles)
Operating Current	115VAC: 2.6 Amps maximum (230 Watts) 230VAC: 1.3 Amps maximum (230 Watts)
Heat Dissipation	784.99 BTU/hr (54.97 calories/second)

DC OUTPUT PARAMETERS

There are three DC supplies: +5 VDC, +12 VDC (nominal voltage), and -12 VDC. All three supplies have overvoltage and overcurrent protection. The +12 VDC and -12 VDC supplies have a feedback loop which provides linear post regulation. The feedback loop to the +5 VDC supply is closed, so the supply does not have linear post regulation. Table C-2 lists the DC output parameters.

Table C-2. DC Power Output Parameters

Output (VDC)	Overall Regulation	Power	Ripple MV	I Minimum ADC	I Maximum ADC
+5	+2%	90W	50	12.0	18.0
+12	+5%	48W	50	.1	*4.0
-12	+5%	6W	50	.1	0.5

*4.5 at turn-on surge (for 30 second duration).

POWER SUPPLY INTERFACE

There are three interface connectors, the 3-pin AC Input connector (J1, AMP part number 350429-1), 9-pin DC output connector (J3, AMP part number 1-48707-0 with pins 350552-3), and 4-pin DC output connector (J4, AMP part number 1-48703-0 with pins 350552-3). Tables C-3, C-4, and C-5 list the connectors' pin assignments.

Table C-3. AC Input Connector J1

Pin Number	Function
1	AC line
2	Chassis ground
3	AC neutral

Table C-4. DC Output Connector J3

Pin Number	Function
1, 4, and 7	+5VDC
8	+12VCD
2	-12VDC
3, 5, 6, and 9	Ground

Table C-5. DC Output Connector J4

Pin Number	Function
1	+12V
2	+12V Return
3	+5V Return
4	+5V

APPENDIX D: BOOTSTRAP ROM SEQUENCE

This appendix describes the bootstrap ROM program, called Boot. Any of the following activates Boot:

- Power on
- Software reset (supervisor write to the System Reset register; asserts HARIKIRI-)
- Reset button.

STATUS INDICATOR LIGHT ASSIGNMENTS

Figure D-1 shows the LEDs in the computer cabinet, with their corresponding color, LED number, and General Control register bit assignment. Boot uses LED4 as a program state indicator. When LED4 is lit, the other LEDs indicate the progress through the program. When LED4 is off, the other LEDs contain the Boot fail code. (All LEDs off indicates no failure.)

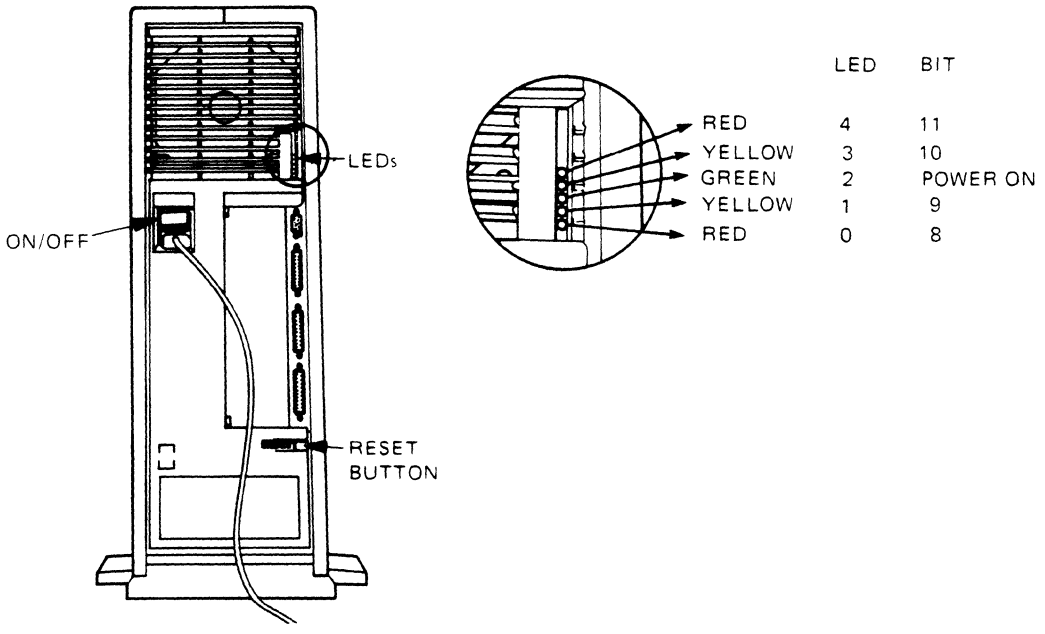


Figure D-1. Status Indicator Light (LED) Assignments

PROGRAM DOCUMENTATION

The ROM goes through the following seqeunce of events during Boot:

1. Boot turns off all the LEDs, to signal successful entry. (RESET turned all the LEDs on.) Simultaneously, Boot negates ROMEN- (General Control register bit 12), which disables the remapping of memory addresses to the ROM.

2. Boot programs timer 1 and counter 2 in the system timer chip to allow the Kernel to read the tick counter and thereby keep accurate account of time during a software generated Boot.

3. Boot initializes the first six memory map locations to a unity map (a one-to-one mapping of virtual addresses to physical addresses). To preserve the machine state at the reset (in case Boot later does a memory dump), Boot saves all the processor and map registers in page zero. Boot then writes the Clear Status register to reset the General Status register.

4. After copying the crash dump table into page zero, Boot checks it to see if the table has a valid signature; if it does not, Boot assumes the reset is a power on; if it does and also contains a panic string, Boot assumes the reset is by software; and if neither of these is true, Boot assumes the reset is by the reset button. Boot places the reason code in the crash dump table. **NOTE:** Only the Kernel sets a valid signature in the crash dump table; hence, only the Kernel will be dumped into the dump area.

5. The computer status is in page zero. Boot initializes the memory map to a unity map between zero and 2-Mbyte, the upper 2-Mbyte set as invalid.

6. The program turns all the LEDs on and initializes the 8259 interrupt controller A. (If there is an expansion 8259, Boot initializes it too.) If the computer aborts Boot during the 8259 initialization, the LEDs stay on; otherwise LED 3 turns off to indicate the program is recalibrating the disks.

7. Boot recalibrates the floppy disk, then recalibrates the hard disk. If the recalibration of a disk does not complete within two seconds, that recalibration is timed out. If Boot cannot recalibrate a disk, a FAIL1 occurs, indicating that the current status of the hardware is such that Boot cannot continue.

8. If Boot has been able to recalibrate one or more disks, it turns LED 4 on, LED 3 off, LED 1 on, and LED Ø off, to indicate it is now doing a dump. **NOTE:** Boot bypasses the dump if the reset was a power on. (Go to step 9.)

   Boot determines the size of valid physical memory, then searches the disk drives for suitable dump area. Upon finding a suitable dump area on a disk, Boot dumps the physical memory to this area. The size of the dump (disk write operation) is the lesser of either physical memory size or the dump area size. No error occurs if Boot does not find a suitable dump area; it just continues.

   LEDs 1-Ø show one of the following codes to indicate which disk the program has selected for the dump:

   LED 1 off and LED Ø on = floppy
   LED 1 on and LED Ø on = hard disk.

9. Next is the memory test. At the start of the memory test, Boot turns LED 4, LED 3, and LED 1 on and LED Ø off, then runs the tests on all memory addresses from $ØC9ØØ to $7FFFF, Kernel addresses, located in the lowest 512K bytes in memory. The write test consists of a write, then read and compare; the write data is the same as the write address. Boot enters the Memfail routine if the data comparison fails.

   When LED Ø goes on, Boot does the address test, which is simply a read-and-compare of the locations loaded during the write test. Boot enters the Memfail routine if the data comparison fails.

   Memfail generates FAIL6 for the write test and generates FAIL7 for the read test.

1Ø. If the memory test didn't generates an error, Boot turns LED 4 on, LED 3 off, LED 1 off, and LED Ø on, to indicate it is searching for the loader program, called Loader. If it cannot find Loader, the computer displays the FAIL3 code, but Boot continues to search for Loader. Inserting a floppy disk with Loader allows Boot to remove FAIL3 and continue.

11. Boot turns LED 4 and LED 3 on and LED 1 and LED Ø off to indicate it is loading Loader program, then reads Loader into memory, starting at address $7ØØØØ. If the Loader read fails, Boot returns to step 1Ø.

12. Once Boot reads Loader from the disk, Boot turns LED 3 on, then stops. Loader program takes over. The crash dump table is located at 128 bytes below $70000, and a unity map is established with the first 512K bytes of memory tested. The stack validity cannot be assumed.

LED CODES

Table D-1 lists the failure codes and progress codes for the LEDs as the Boot program proceeds.

Table D-1. Boot and Loader LED Status Codes

LED 4	LED 3	LED 1	LED Ø	Status
off	off	off	off	Successful completion of Boot.
off	off	off	on	FAIL1. Can't recalibrate any disk.
off	on	off	on	FAIL3. Can't find Loader. (Boot continues to search for Loader when FAIL3 is active.)
off	on	on	off	FAIL6. Write memory test fail.
off	on	on	on	FAIL7. Address memory test fail.
on	off	off	off	Recalibrating disks.
on	off	off	on	Searching for Loader. Loader searching for /UNIX.
on	on	off	off	Loading Loader. Loader loading /UNIX.
on	off	on	on	Entering Loader.
on	off	on	off	Searching for a dump area.
on	on	off	on	• Dumping to floppy. • Entering loaded program • Booting from floppy.
on	on	on	off	Write memory test.
on	on	on	on	• Dumping to hard disk. • Booting from hard disk. • Address test occurring. • Initial RESET before Boot.
off	off	off	off	Initializing timers and save status.
off	on	off	off	Debugger waiting for connection during down load.
on	off	off	off	Debugger doing down load.

APPENDIX E: DISK DRIVE DOCUMENTATION

This appendix contains information on the hard disk and floppy disk drives.

PRODUCT SPECIFICATION
FOR
FLEXIBLE DISK DRIVE
MODEL 9409

MAGNETIC PERIPHERALS INC.
a subsidiary of
CONTROL DATA CORPORATION

PRODUCT SPECIFICATION FOR THE 9409 FDD

TABLE OF CONTENTS

1.0 GENERAL DESCRIPTION

This specification describes the Model 9409 Flexible Disk Drive (FDD). The 5.25-inch FDD features simplicity of design and high reliability at low cost. Device applications include small computer systems, intelligent data teminals, and home computers. Refer to Figures 1 and 2.

The 9409 is a compact, random-access disk drive utilizing standard double-sided 5.25-inch flexible diskette media in either single density (FM) or double density (MFM) recording.

The 9409 contains a drive mechanism to rotate the media, a track accessing positioner, read/write control, and interface circuitry. The interface signal level, power requirements, physical size, and mechanical mounting are compatible with standard industrial requirements. Media interchange is achieved with standard diskette formats which accommodate tunnel erase head structures.

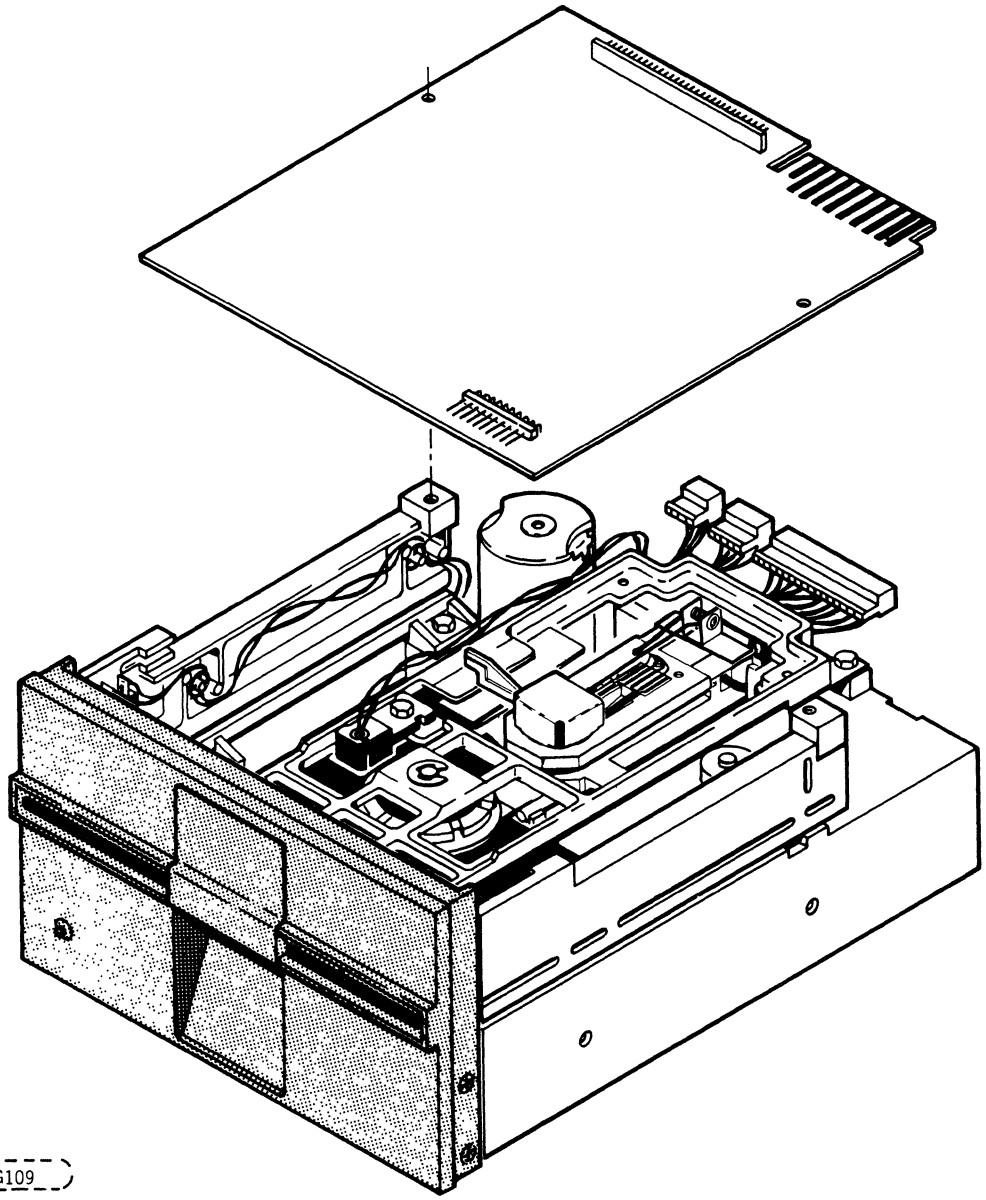


Figure 1. 9409 Flexible Disk Drive

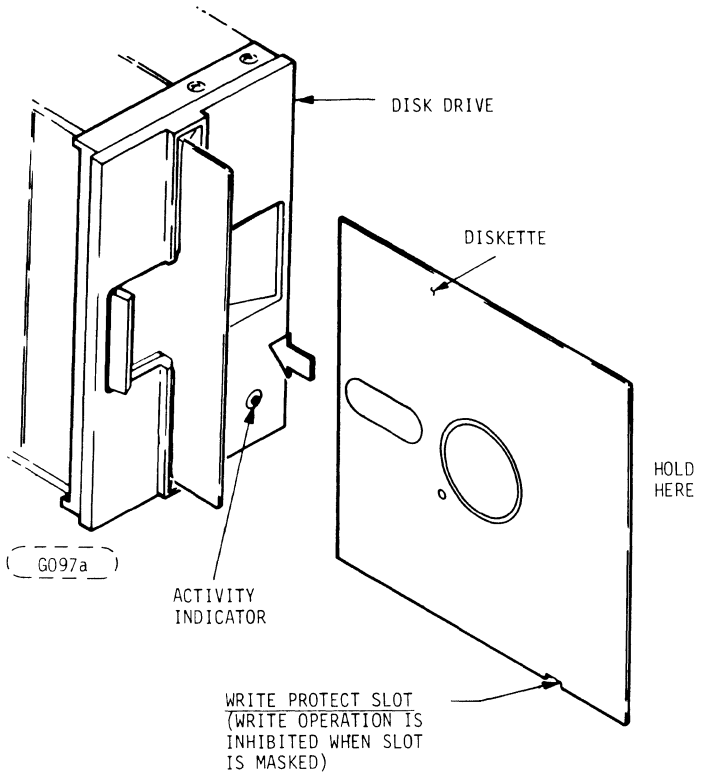


Figure 2. Diskette Insertion

1.0 -contd.

Index pulses are produced by a photoelectric sensing assembly mounted within the 9409. Track-to-track accessing and positioning are accomplished with a DC-powered, +12-volt band stepper motor.

2.0 APPLICABLE DOCUMENTS

2.1 STANDARDS

The Model 9409 FDD has been designed as a system peripheral to the best standards of design and construction. The drive, however, must depend upon its host equipment to receive adequate power and environment in order to provide optimum performance and compliance with applicable industry and governmental regulations. Special attention must be given in the areas of safety, power distribution, grounding, shielding, audible noise control, and temperature regulation of the device to ensure specified performance and compliance with all applicable regulations.

2.1 -contd.

The 9409 is a component and is not subject to standards imposed by FCC Docket 20780/FCC 80-148 Part 15 governing EMI of computing devices.

2.2 DOCUMENTATION

The following documents and specifications provide detailed theory of operation, maintenance procedures, and media format considerations. These documents are not included with each unit, but must be ordered separately.

- Model 9409 FDD Hardware Maintenance Manual, 77653408-3
- Alignment Diskette Specification (Double-Sided Operation), 76209000
- Double-Sided Diskette Specification (5.25-Inch Double Density), 76205900
- Application Note, PLO and Write Precompensation for 5.25-Inch Flexible Disk Drives, 77653447-1
- Application Note, 5.25-Inch FDD Format Considerations and Controller Compatibilities, 75897469

3.0 FEATURES

3.1 STANDARD FEATURES

The 9409 FDD has the following standard features:

- Double-Sided 5.25-Inch Media Capability
- Single-Density (FM) and Double-Density (MFM) Encoding
- Industry-Compatible Interface
- Industry-Compatible Mounting Configurations
- Only Two DC Power Voltages Required (+5 V and +12 V)
- Write Protect Indication
- Activity Indicator
- Long-Life Ceramic Read/Write Heads
- Band Stepper Motor
- Hard or Soft Sector Formatting
- Head Select
- Drive Select
- Track-Zero Detection
- Index Detection
- Step/Direction Control for Track-to-Track Access
- Black Finish

3.2 ACCESSORIES/SPECIAL TOOLS

The following accessories and special tools are recommended:

- Alignment Diskette, CDC Model 445-51
- DIP Switch, AMP 435626-4
- Cutting Tool, AMP 1-435830-4
- Shunt Module, AMP 435704-6
- Connector Kit, 75887774-0

PRODUCT SPECIFICATION FOR THE 9409 FDD

4.0 PERFORMANCE SPECIFICATIONS

	Single Density	Double Density
Capacity		
Unformatted		
Per Disk	250.0 kbytes	500.0 kbytes
Per Surface	125.0 kbytes	250.0 kbytes
Per Track	3.125 kbytes	6.250 kbytes
Formatted (16 Sectors, 128/256 Bytes)*		
Per Disk	163.84 kbytes	327.68 kbytes
Per Surface	81.92 kbytes	163.84 kbytes
Per Track	2048 bytes	4096 bytes
Code	FM	MFM
Transfer Rate	125 kbits/s	250 kbits/s
Latency (Average)	100 ms	100 ms
Seek Time		
Track-to-Track	5 ms	5 ms
Average	80 ms	80 ms
Settle Time	15 ms	15 ms
Head-Load Time	50 ms	50 ms
Side-Select Time	200 µs**	200 µs**
Write Precompensation	0	250 ns (trk 0-39)

4.1 DEFINITIONS

4.1.1 Latency

Latency is the time required for the read/write head to reach a particular area on a track after positioning is completed. For a rotational/spindle speed of 300 r/min +3.7%, -3.5%, the 9409 has an average latency of 100 ms.

4.1.2 Seek/Settle Time

The seek/settle time is the time required by the read/write head to move from track to track plus the time needed for the head to stabilize on track prior to data transfer. Seek/settle time for the 9409 is 20 milliseconds: 5 milliseconds for a single track access and 15 milliseconds for head settling time.

* Assumes tunnel erase heads.

** Assumes motor on, drive selected, head loaded, and erase turn-off delay completed from any previous write operation.

4.1.3 Head-Load Time

Head-load time is defined as the time from initial activation of the head-load function until valid data transfer between the drive and controller can be assured. The amount of head-load time required by the 9409 is 50 milliseconds. If the unit is configured with head loaded and motor on, valid read data occurs 2 microseconds after drive select.

4.1.4 Read Stabilization

Upon conclusion of a write operation, a minimum delay time of 1 millisecond is required before valid read data can be assured. This time is necessary to allow for the read circuit to stabilize after switching.

5.0 FUNCTIONAL SPECIFICATIONS

	Single Density	Double Density
Encoding Method	FM	MFM
Rotational Speed	300 r/min	300 r/min
Recording Density (Track 39, Side 1)	2938 bpi	5876 bpi
Flux Reversal Density (Track 39, Side 1)	5876 FRI	5876 FRI
Track Density	48 TPI	48 TPI
Tracks per Surface	40	40
Heads	2	2
Inside Recorded Radius (Side 0)	1.437 in (36.49 mm)	1.437 in (36.49 mm)
Outside Recorded Radius (Side 0)	2.250 in (57.15 mm)	2.250 in (57.15 mm)
Inside Recorded Radius (Side 1)	1.354 in (34.39 mm)	1.354 in (34.39 mm)
Outside Recorded Radius (Side 1)	2.167 in (55.04 mm)	2.167 in (34.39 mm)
Motor Start Time	0.5 second	0.5 second
Media	CDC Model 445 or equivalent	CDC Model 445 or equivalent

6.0 RELIABILITY SPECIFICATIONS

The following reliability specifications assume correct host/drive operational interface has been implemented, including all interface timings, power supply voltages, environmental conditions, use of recommended media (or equivalent), and application of recommended data-recovery techniques. (PLO and Write Precompensation Application Note is available on request.) The following MTBF assumes spindle drive motor duty cycle is 25% of power-on hours.

Error Rates	
Soft Read Errors	1 per 10^9 bits read
Hard Read Errors	1 per 10^{12} bits read
Seek Errors	1 per 10^6 seeks
MTBF	8000 Power-On Hours, Typical Usage
MTTR	0.5 Hour
Service Life	5 years
Media Life	
Wear Revolutions	Greater than 3 X 10^6
Insertions	Greater than 30,000
Preventive Maintenance	None Required

7.0 PHYSICAL/ELECTRICAL SPECIFICATIONS

7.1 POWER REQUIREMENTS

There are no AC power requirements for the 9409. DC power requirements are as shown in Table 1.

Table 1. DC Power Requirements

CONDITIONS	NOMINAL VOLTAGE	
	+5 V	+12 V
Tolerance	±0.25 V	±0.6 V
Ripple (Peak to Peak)	≤50 mV	≤100 mV
Seeking Current:		
Typical	0.5 A	0.9 A
Maximum	*0.7 A	*1.8 A

*Specified current requirements are on a per-drive basis. Current requirement increases by a factor equal to the number of drives per power supply. (For example, two drives per supply would require +12 V at 3.6 A and +5 V at 1.4 A.)

7.2 POWER DISSIPATION

26.4 watts (maximum)
13.3 watts (nominal)

7.3 INRUSH CURRENT

Inrush current is dependent upon the power source as well as the 9409. Primary considerations are host power supply source impedance(s), host-to-9409 power line resistance and inductance, and 9409 capacitance load presented by the filter capacitors. Schematically, this may be represented as shown in Figure 3.

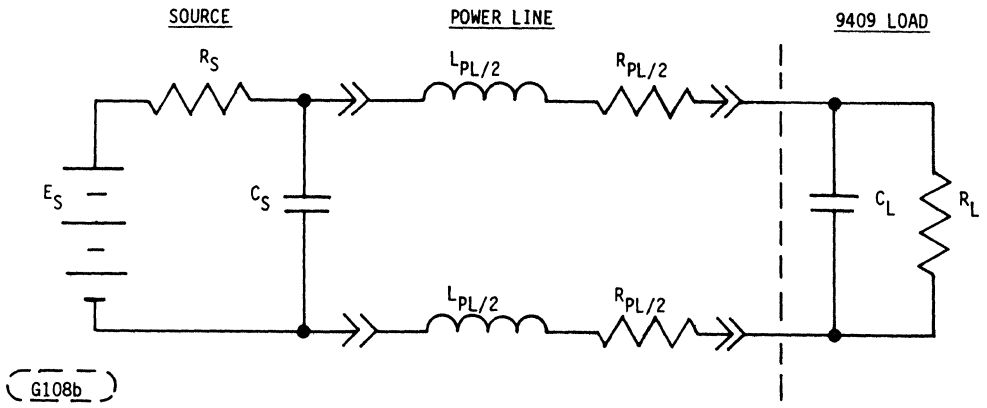


Figure 3. Inrush Current

Nominal equivalent loads for the 9409 are as follows:

	+12 V	+5 V
C_L	22 microfarads	11.8 microfards
R_L	13 ohms	10 ohms

7.4 ENVIRONMENTAL LIMITS

Temperature and humidity specifications preclude condensation on any drive part. The 9409 is intended for use with host systems which operate in computer room and office environments. Alititude and barometric pressure specifications are referenced to a standard day at 58.7°F (14.8°C).

	Operating	Shipping & Storage
Ambient Temperature	*40° to 115°F *(4.4° to 46.1°C)	-40° to 144°F (-40° to 62.2°C)
Temperature Gradient	18°F/hr (-7.7°C/hr)	36°F/hr (20°C/hr)
Relative Humidity	20% to 80%	0 to 95%
Maximum Wet Bulb Temperature	79°F (26.1°C)	No condensation
Altitude (Sea Level Reference)	-983 to +9850 feet (-300 to +3002 meters)	-983 to +9850 feet (-300 to +3002 meters)
Barometric Pressure		
in. of HG	30.9 to 22.2	30.9 to 18.8
mm of HG	784.9 to 563.6	784.9 to 476.9
psi	15.18 to 10.9	15.18 to 9.22

7.5 MECHANICAL SPECIFICATIONS

Following are mechanical specifications for the 9409. Refer to Figure 4 for detailed mounting dimensions.

Height:	3.38 inches	85.8 mm
Width:	5.88 inches	149.4 mm
Depth:	8.29 inches	210.6 mm
Weight:	3.2 pounds	1.45 kg

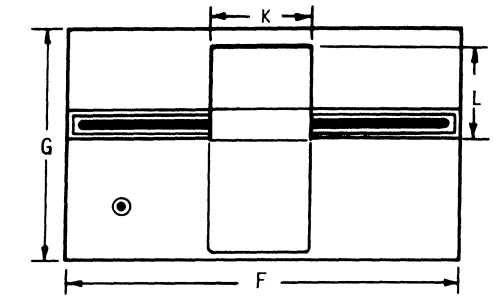
*Actual operating temperature is limited by media environmental specifications and performance. See applicable media specifications.

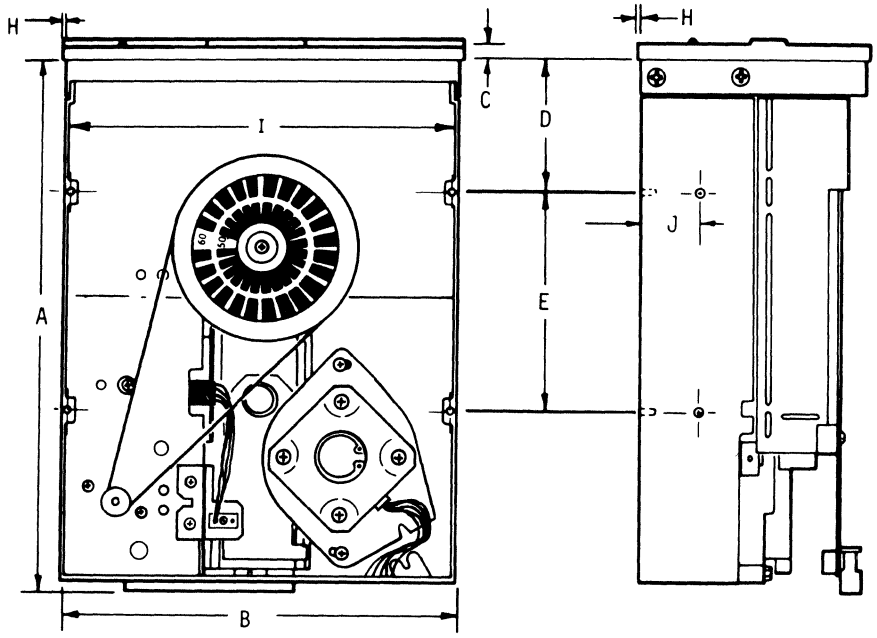
PRODUCT SPECIFICATION FOR THE 9409 FDD

WT: 3.2 lbs. (1.45 kg)

MOUNTING HOLES: FOUR ON BOTTOM, TWO EACH SIDE; #6-32 X 0.31 in. (7.9 mm) DEPTH

PACKAGE DIMENSION		
	INCHES	MILLIMETERS
A	8.00 MAX	203.2 MAX
B	5.75 +0.00 -0.02	146 +0.00 -0.5
C	0.29 ±0.02	7.4 ±0.5
D	1.87 ±0.02	47.5 ±0.5
E	3.12 ±0.02	79.2 ±0.5
F	5.88 ±0.01	149.4 ±0.3
G	3.38 ±0.01	85.9 ±0.3
H	0.06 ±0.01	1.5 ±0.3
I	5.50 ±0.02	140 ±0.5
J	0.86 ±0.02	21.8 ±0.5
K	1.50 ±0.01	38.1 ±0.5
L	1.41 ±0.01	35.8 ±0.5





G197a

Figure 4. Dimensions of 9409 FDD

7.5.1 Drive Orientation

The 9409 can be mounted in any of the following loading positions:

- horizontal with the access door opening upward and PCB facing upward
- vertical with the access door opening to the left or right
- upright with the front panel at the top.

The drive should not be mounted with the PCB facing downward.

8.0 RECORDING CHARACTERISTICS/FORMAT

Capacity is determined by customer selected data format and coding techniques.

8.1 ENCODING METHODS

Data can be recorded in either FM or MFM encoding. The format of the data recorded is a function of the host system.

8.1.1 Single-Density Recording (Figure 5)

Double-frequency modulation (FM) recording is recommended for use in the single-density mode. FM recording is self-clocking and provides synchronization for data separation utilizing simple one-shot techniques. However, use of a phase-lock oscillator (PLO)* instead of a simple one-shot system will increase operating margins and result in greater data reliability with more tolerance for media imperfections.

8.1.2 Double-Density Recording (Figure 5)

Modified frequency modulation (MFM) recording doubles the drive's capacity. Since the MFM coding technique is not self-clocking, it is necessary to use a PLO* to generate a synchronized clock from the data stream. If both single- and double-density systems are being used, the same PLO serves both applications. Use of MFM recording requires the use of precompensation during writing in order to limit the effective peak shift in the read-back signal.

*PLO and Write Precompensation Application Note is available on request.

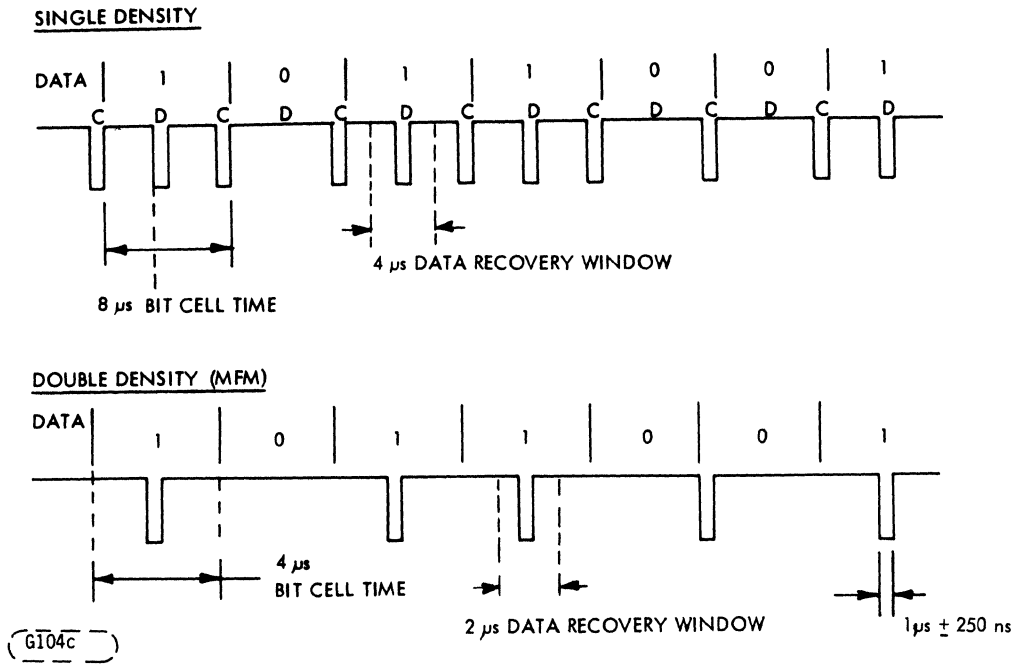


Figure 5. Nominal Data Timing

8.1.2.1 Write Precompensation

Write precompensation is required to decrease the effect of bit shift on the reduced bit cell and window of MFM data. The window is defined as being the total time allowed for the bit to appear and be recognized. The data recovery window of MFM is 2 microseconds as opposed to the FM window of 4 microseconds . The amount of write precompensation is 250 nanoseconds early or late in relation to nominal.

Write precompensation should be applied to all tracks, 0 through 39, on both sides of the disk.

Write precompensation is applied to data patterns that will result in a large amount of bit shift. The controller precompensation circuit looks at three bits on each side of a reference bit and determines whether or not to shift. (PLO and Write Precompensation Application Note is available on request.) The following patterns are compensated (bit-shifted) in the direction of the arrow.

8.1.2.1 -contd.

LATE	EARLY
X 0 1 1	X 1 1 0
1 0 0 0	0 0 0 1

X = Don't Care

The following is an explanation of the MFM write precompensation algorithm. When a flux transition pattern of 011 is written on the disk, the first 1 is shifted toward the 0 (data bit cell with or without flux transition). Write precompensation shifts this 1 in the opposite direction the amount of the expected shift. In the case of the 1000 pattern, the second 0 (clock bit) shifts toward the first 0 (data bit cell with no flux transition) and this clock bit is compensated late towards the third 0 (clock bit). With pattern 110 the second 1 shifts toward the 0 (bit cell with no transition) so this 1 data bit is compensated early towards the first 1 data bit. In the last pattern 0001, the third 0 (clock bit) is shifted towards the 1 (data bit) due to the nominal 6 microseconds between these two bits. Thus, the third 0 (clock bit) is compensated early towards the second 0 to counteract shifting towards the 1.

8.2 FORMAT CONSIDERATIONS

Design of the 9409 allows for data recording using either hard or soft sector formats. The following operational tolerances should be considered when selecting a particular format for the system. For additional information consult Application Note 75897469, 5.25-Inch FDD Format Considerations and Controller Compatibilities.

8.2.1 Index Tolerance

8.2.1.1 Post Index

Before initial recording of data upon a selected track, a minimum gap time of 1 millisecond is required to allow for drive-to-drive and adjustment tolerance.

8.2.1.2 Pre-Index

To compensate for drive-motor speed tolerances, a minimum gap spacing of 7.2 milliseconds is required.

8.2.2 Inter-Record Tolerance

8.2.2.1 Pre-Address

The Pre-address length is a minimum of 1.64 milliseconds, but varies with sector size for 128-byte sector lengths. The pre-address gap timing is Gap No. 3 and is based on the tunnel-erase structure of the read/write head and maximum erase-circuit turn-off delay.

8.2.2.2 Pre-Data

Pre-data gap is Gap No. 2 and has a minimum timing requirement of 489 microseconds. Gap timing is determined by the erase turn-on circuit tolerance and the tunnel-erase structure of the read/write head.

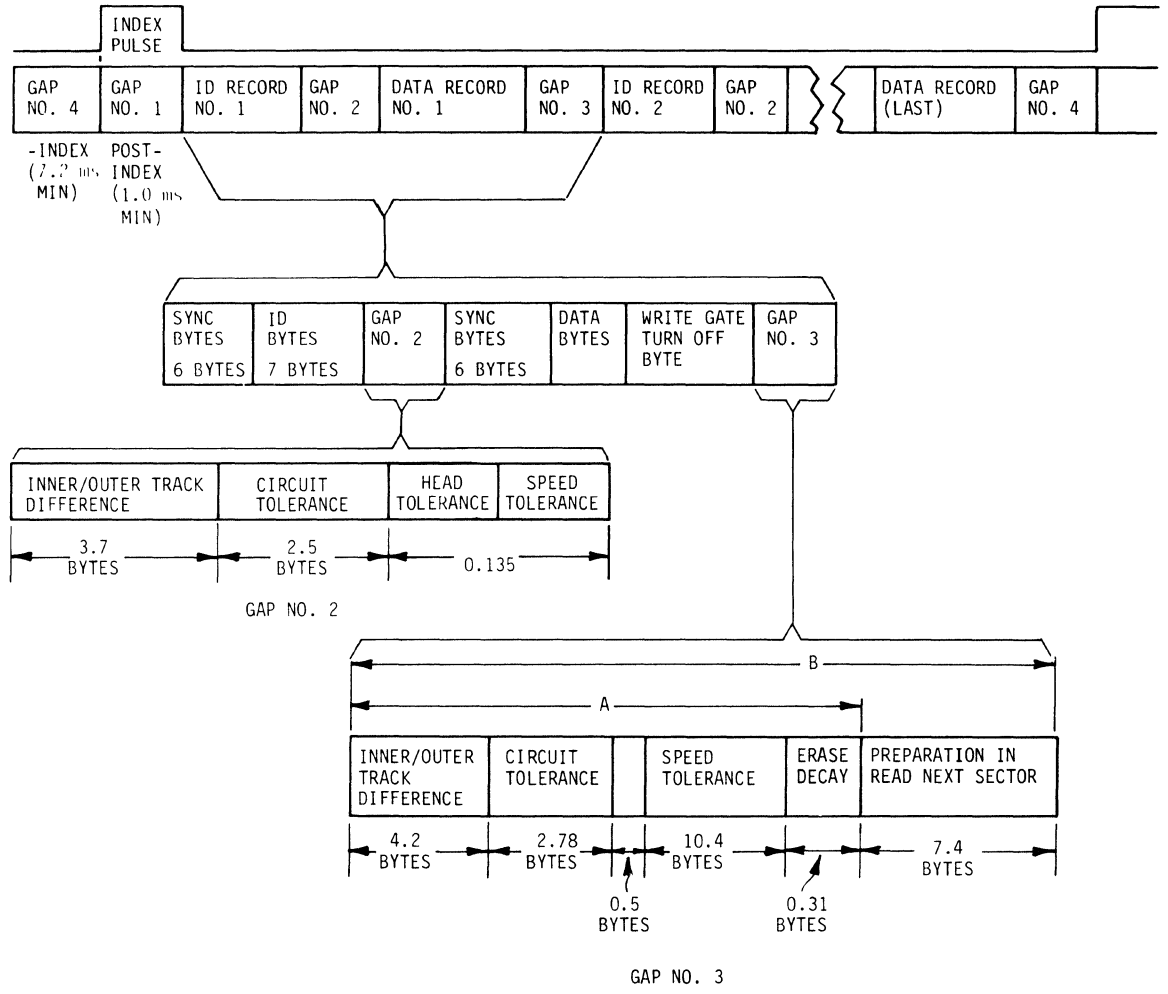
8.2.3 Soft-Sector Formatting

For system use of soft-sector formatting, Figure 6 provides several recommended timing tolerances for individual gaps positioned within the format. For soft-sector formats utilizing MFM recording, Figure 7 presents the minimum gap for timing tolerances required for this recording scheme.

8.2.4 Hard-Sector Formatting

Hard-sector formats use sector holes in the diskette to separate the track into sectors. The mechanical and electrical parameters of flexible disk drives and media require the format to consist of three parts--Preamble, User Data, and Postamble.

The Preamble consists of a defined number of gap bytes that are written starting at the beginning of the sector pulse. The Preamble length is defined so that under worst-case conditions User Data will not be lost due to sector pulse jitter or erase delays.



A - 18.2 BYTES - TO PREVENT DATA LOSS UNDER WORST-CASE CONDITIONS.

B - 25.6 BYTES - GUARANTEES CONSECUTIVE SECTOR OPERATION UNDER WORST CASE CONDITIONS.

ZZ102b

Figure 6. Minimum Requirements for Single-Density Soft-Sector Format (Part 1)

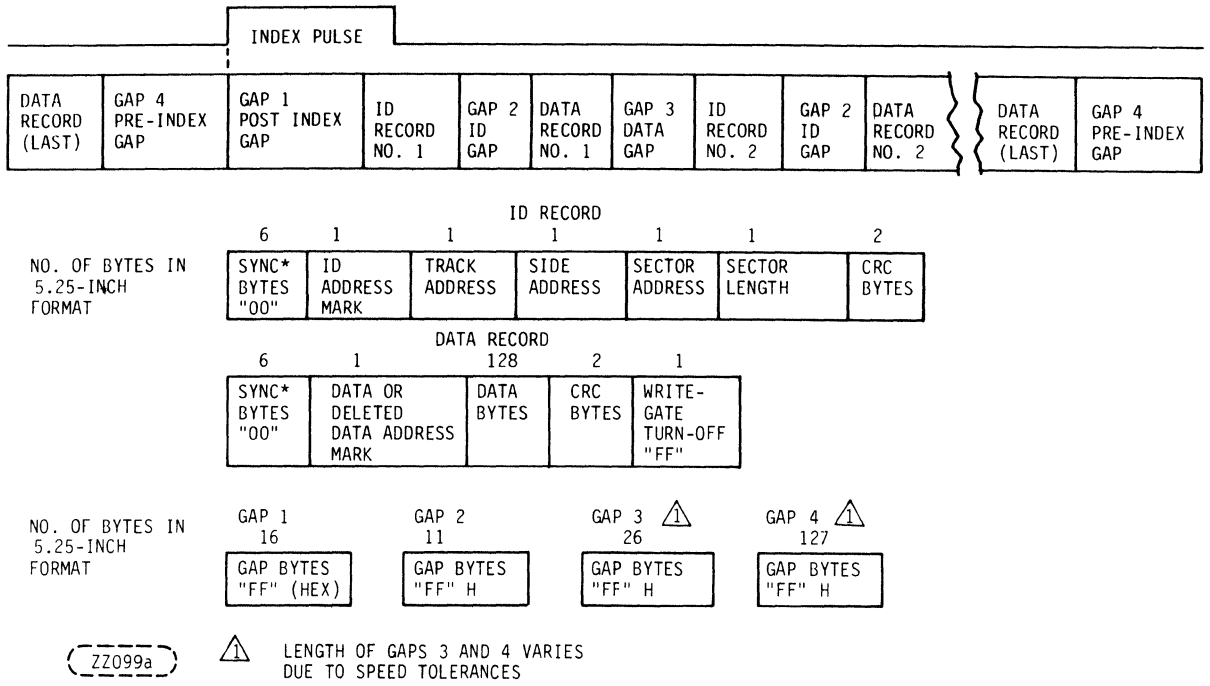
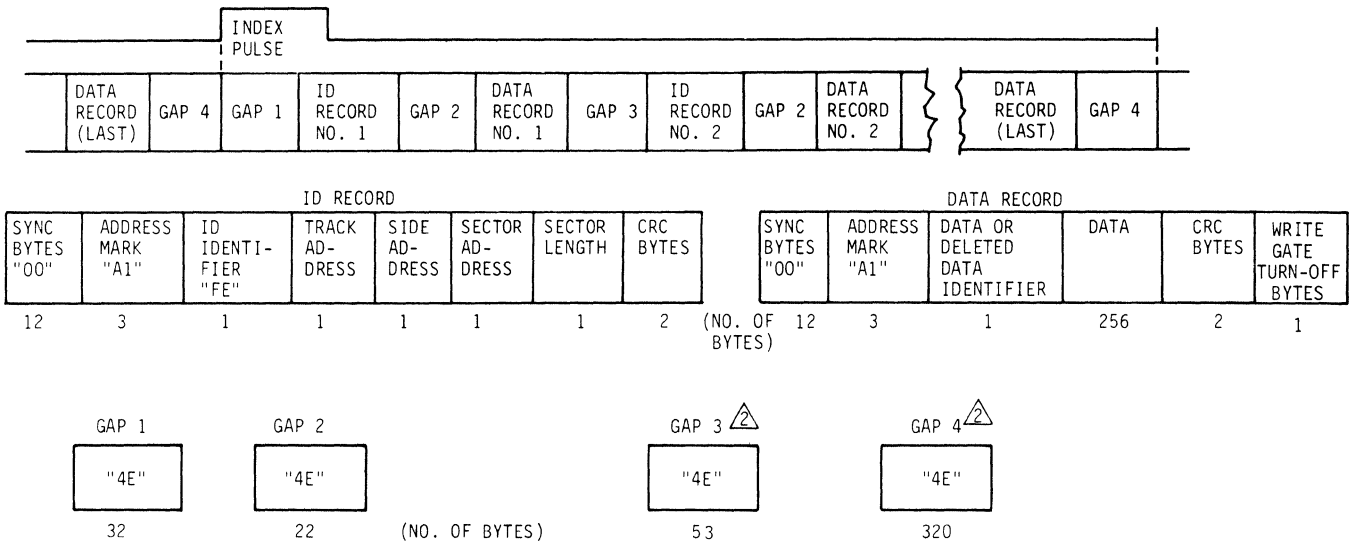


Figure 6. Minimum Requirements for Single-Density Soft-Sector Format (Part 2)



⚠1 SHOWN FOR 16-256 BYTE SECTORS.

⚠2 LENGTH OF GAPS 3 AND 4 VARIES DUE TO SPEED TOLERANCES.

ZZ099c

Figure 7. Recommended Double-Density Format

8.2.4 -contd.

User Data consists of data, sync bytes, ID bytes, and CRC bytes used by the particular format. User Data and the Preamble are both written each time a sector is updated.

A buffer zone, or Postamble, is required at the end of a sector between the User Data and the earliest sector pulse.

Recommended minimum Preamble and Postamble gaps are given in Table 2 and illustrated in Figure 8.

Table 2. Recommended Preamble/Postamble Gap

POSTAMBLE		BLANK (NO CHARACTER WRITTEN)	ZEROES (CHARACTER WRITTEN)
16 SECTORS PER TRACK	MINIMUM PREAMBLE	8 BYTES	24 BYTES
	MAXIMUM USER DATA	156 BYTES	148 BYTES
	MINIMUM POSTAMBLE	22 BYTES	15 BYTES
10 SECTORS PER TRACK	MINIMUM PREAMBLE	8 BYTES	23 BYTES
	MAXIMUM USER DATA	269 BYTES	261 BYTES
	MINIMUM POSTAMBLE	27 BYTES	19 BYTES

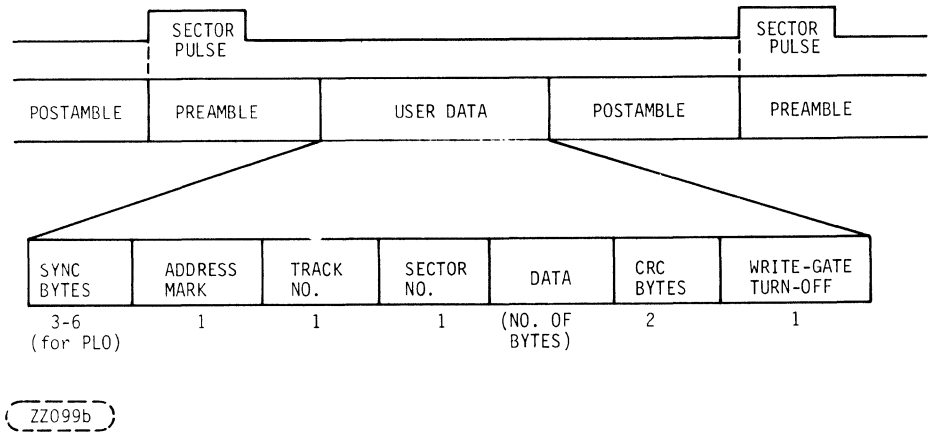


Figure 8. Recommended Hard-Sector Format

8.2.4 -contd.

As shown in Table 2, there are two ways to configure the Postamble Field, Blank and Zeroes.

1. Blank (No Character Written) - the controller can turn off write data at the end of the User Data field and leave the Postamble as an empty or blank field.
2. Zeroes (Character Written) - the write gate can turn off write data at the beginning of the next sector pulse. When doing this, the controller must write gap characters; e.g., all zeroes in the Postamble field.

The write gate turn off at the end of the User Data field is recommended because it allows the use of a larger User Data field. The values given for maximum User Data field guarantee maximum recovery for read-after-write operation under worst-case conditions.

9.0 INSTALLATION INFORMATION

9.1 ELECTRICAL INTERFACE

The electrical interface between the 9409 and host system consists of two primary connectors. The I/O signal connector J1 contains all interface signals transmitted to and from the drive. Connector J2 supplies DC power to the 9409. AC power is not applicable to this device. A separate frame ground connection is available at the rear of the unit.

See Figure 9 for interface connections and Figure 10 for connector locations.

9.2 POWER CONNECTOR

There is no AC power connector on the 9409.

DC power is provided to the 9409 by the host system through connector J2, which is mounted on the non-component side of the printed circuit board near the spindle drive motor. Refer to Figure 10 for connector location and Table 3 for pin assignments. Table 4 lists the recommended mating connector for J2.

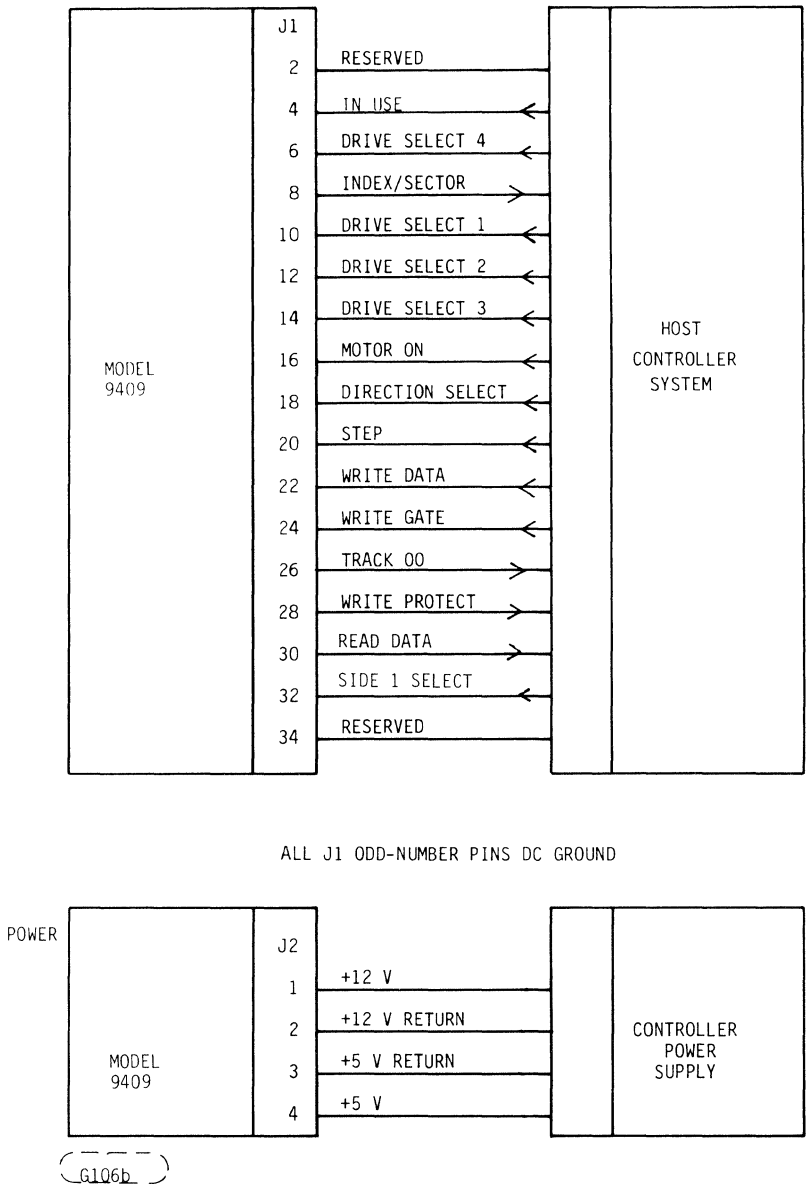


Figure 9. Interface Connections

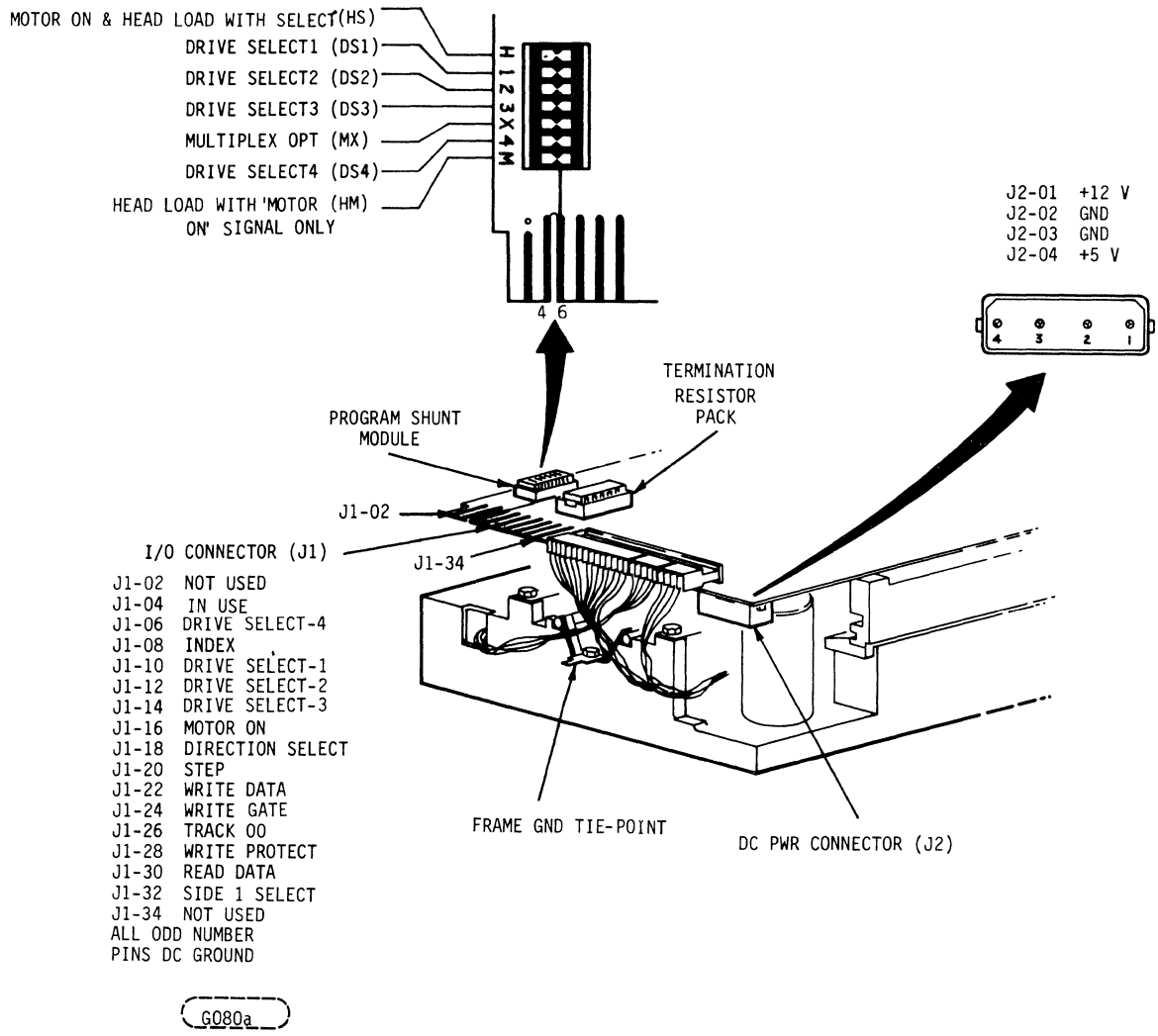


Figure 10. Input/Output (J1), DC Power (J2), Terminator, and Programmable Shunt Module Location/Description

Table 3. DC Interface

POWER LINE DESIGNATION	PIN NUMBER
+12 VOLTS	J2-01
+12 VOLTS RETURN	J2-02
+5 VOLTS RETURN	J2-03
+5 VOLTS	J2-04

Listed in Table 4 is connector information for J2/P2.

Table 4. J2/P2

TYPE OF CABLE	CONNECTOR	CONTACTS
	J2, 4-pin AMP Mate-N-Lok 350211-1	
18 AWG	P2, AMP 1-480424-0	AMP 61473-1

9.3 I/O CONNECTOR

The I/O connector J1 is a 34-pin PCB edge connector. Its location is shown in Figure 10. The dimensions for J1 are shown in Figure 11. The pins are numbered 1 through 34, with the even-numbered pins appearing on the component side of the PCB and odd-numbered pins on the non-component side. A key slot is provided between pins 4 and 6 for optional connector keying.

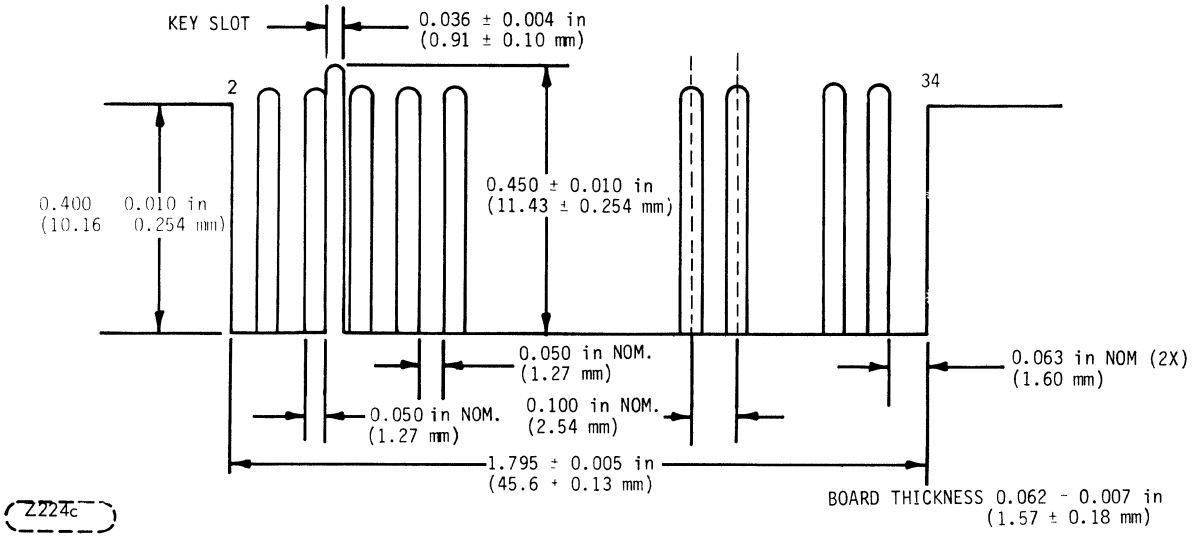


Figure 11. I/O Connector (J1) Dimensions

9.3 -contd.

The maximum I/O cable length between the host controller and the 9409 (or last FDD in daisy chain configuration) should be no greater than 10 feet.

The recommended mating connector for J1 and type of cable are shown in Table 5.

Table 5. J1/P1

TYPE OF CABLE	CONNECTOR	CONTACTS
TWISTED PAIR, 26 AWG	AMP 583717-5	AMP 1-583616-4
FLAT CABLE	3M SCOTCHFLEX 3463-0001	---
---	AMP KEYING PIN 583274-2	---

9.4 GROUNDING REQUIREMENTS

To ensure optimum performance and noise immunity, the 9409 must be frame grounded to the host equipment AC ground. A Faston tab has been provided on the rear of the casting where its mating Faston connector can be attached or soldered. The tab is AMP part number 61664-1 and its mating connector is AMP part number 60972-1.

9.5 CONFIGURATIONS

9.5.1 Daisy-Chain Configuration

A daisy-chain configuration incorporates interfacing of the disk drives on a common I/O cable. Only the FDD which is selected by the host system has its control and data signals enabled through this common interface. The program shunt module shown in Figure 10 allows creation of each individual drive-selection address for each drive in the daisy-chain. Four drive selection addresses are made available by the program shunt module for interfacing up to four drives in a daisy chain configuration (shown in Figure 12).

Refer to paragraph 12.0 for descriptions of drive selection and head load options.

Termination in a daisy chain configuration is described in paragraph 9.6.3.

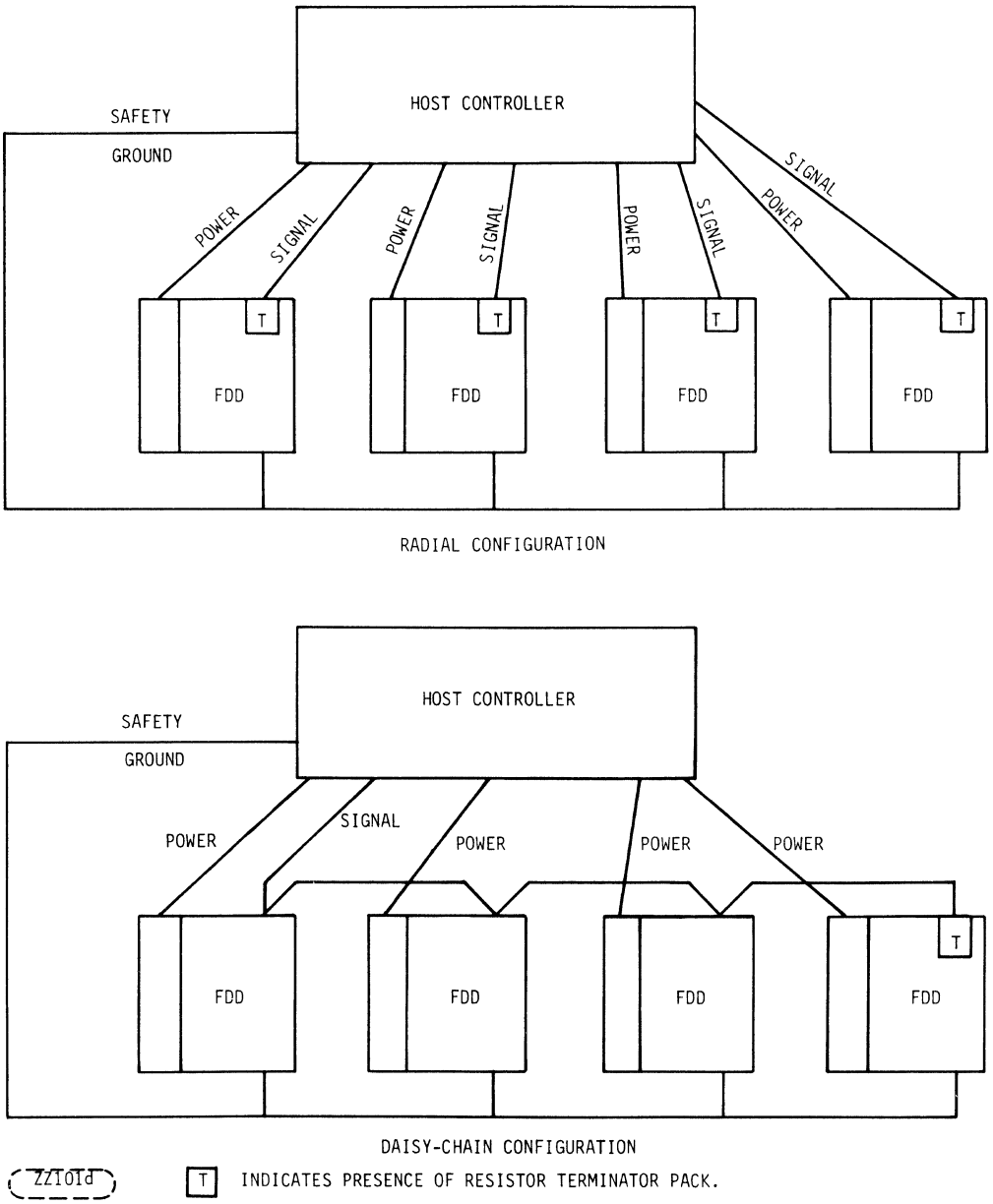


Figure 12. FDD System Connection Configurations

9.5.2 Radial Configuration

In a radial configuration the disk drives do not share the same I/O cable. Each drive is interfaced to its own I/O cable which, in turn, allows interfacing of more than four drives and a variety of system operational techniques. Drive-selection addressing must be accommodated at the program shunt module to allow activation of I/O signals.

Refer to paragraph 12.0 for descriptions of drive select and head load options.

A radial configuration is shown in Figure 12 and termination is described in paragraph 9.6.2.

9.6 INTERFACE TERMINATIONS

Figure 13 provides a logical representation of the interface line termination used for the 9409.

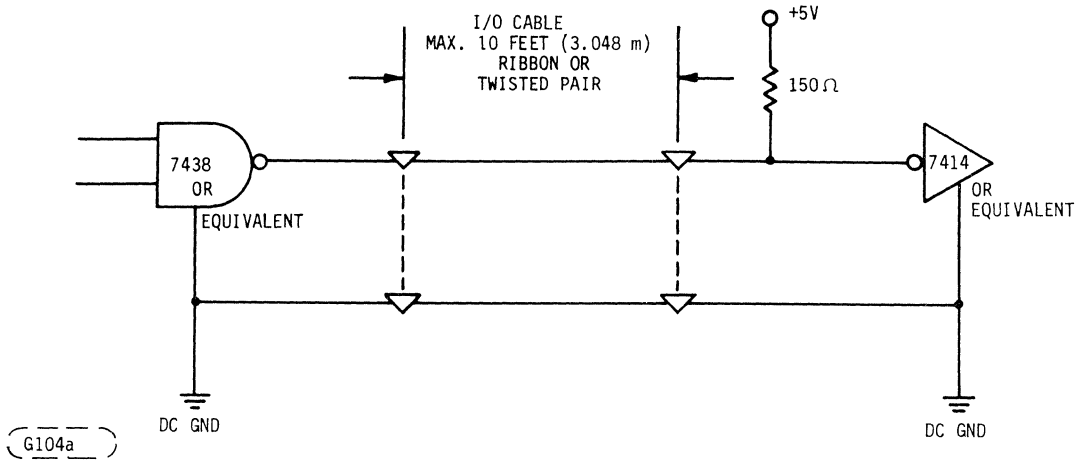


Figure 13. Interface Terminations

9.6.1 Areas of Termination

Termination for each input line is accommodated by a 150-ohm resistor pack installed in a DIP socket located on the PCB of the 9409 (Figure 10).

9.6.2 Termination of a Radial Configuration

In a single drive or radial configuration, the resistor pack must be kept in place on each PCB to provide the proper terminations.

9.6.3 Termination in a Daisy-Chain Configuration

In a daisy-chain configuration, only the last drive on the interface is to be terminated. All other drives on the interface must have the resistor pack removed.

9.6.4 External Terminations

For configurations requiring external termination, the user must terminate each input line to +5 V DC through a 150-ohm, 1/4-watt resistor.

10.0 ERROR RECOVERY

Seek errors will rarely occur unless the stepping rate is exceeded. In the event of a seek error, recalibration of track location can be achieved by repetitive step-out commands until a Track 00 signal is received.

To guard against degradation from imperfections in the media, no more than three attempts to write a record should be attempted when read-after-write errors are encountered. If a record cannot be successfully written within four attempts, it is recommended that the sector or track be labeled defective and an alternate sector or track assigned. If more than two defective tracks are encountered, it is recommended that the diskette be replaced.

In the event of a read error, up to 10 attempts should be made to recover with reread operations. If after 10 attempts the data has not been recovered, step the head several tracks away and then reposition to recover the data. Unloading the head when data transfers are not imminent will increase the data reliability and extend diskette life.

11.0 INTERFACE SIGNAL DESCRIPTION

11.1 ACTIVE/INACTIVE LOGIC SIGNALS

Signal State	Logic Level	Voltage
Inactive (False)	Logic 0/High	+2.5 V to +5.25 V
Active (True)	Logic 1/Low	0.0 V to +0.4 V

11.2 Input Signals

11.2.1 Drive Select

Drive Select (J1-6, -10, -12, -14) activates the internal circuitry of a selected drive. Activating this line to a logic low (active) level will condition the drive's input and output lines. All input and output lines are gated with drive select with the exception of Motor On (J1-16) and In Use (J1-02).

After the activaction of the Drive Select line, a minimum time delay is required before the start of any write or read operation (see Figures 14 and 17). See Table 6 for these minimum time delays which are dependent on drive shunt configuration.

Table 6. Minimum Time Delays

MODE OF OPERATION	TIME TO VALID READ DATA
1. Motor up to speed and head loaded.	2 microseconds
2. Motor up to speed, then load the heads.	50 milliseconds
3. Motor on when drive selected.	0.5 seconds

The operation of the Drive Select lines and selection of a designated drive is controlled through the programmable shunt position DS1, DS2, DS3, and DS4. Refer to section 12.0 for further information.

11.2.2 Motor On

The Motor On circuit can be configured to operate in the three different ways described below.

1. If shunt HM is shorted and HS is open, the drive motor will turn on and the head will load when a logic low is applied to Motor On line J1-16. There is a minimum 0.5 second delay from active Motor On until there is valid read data.

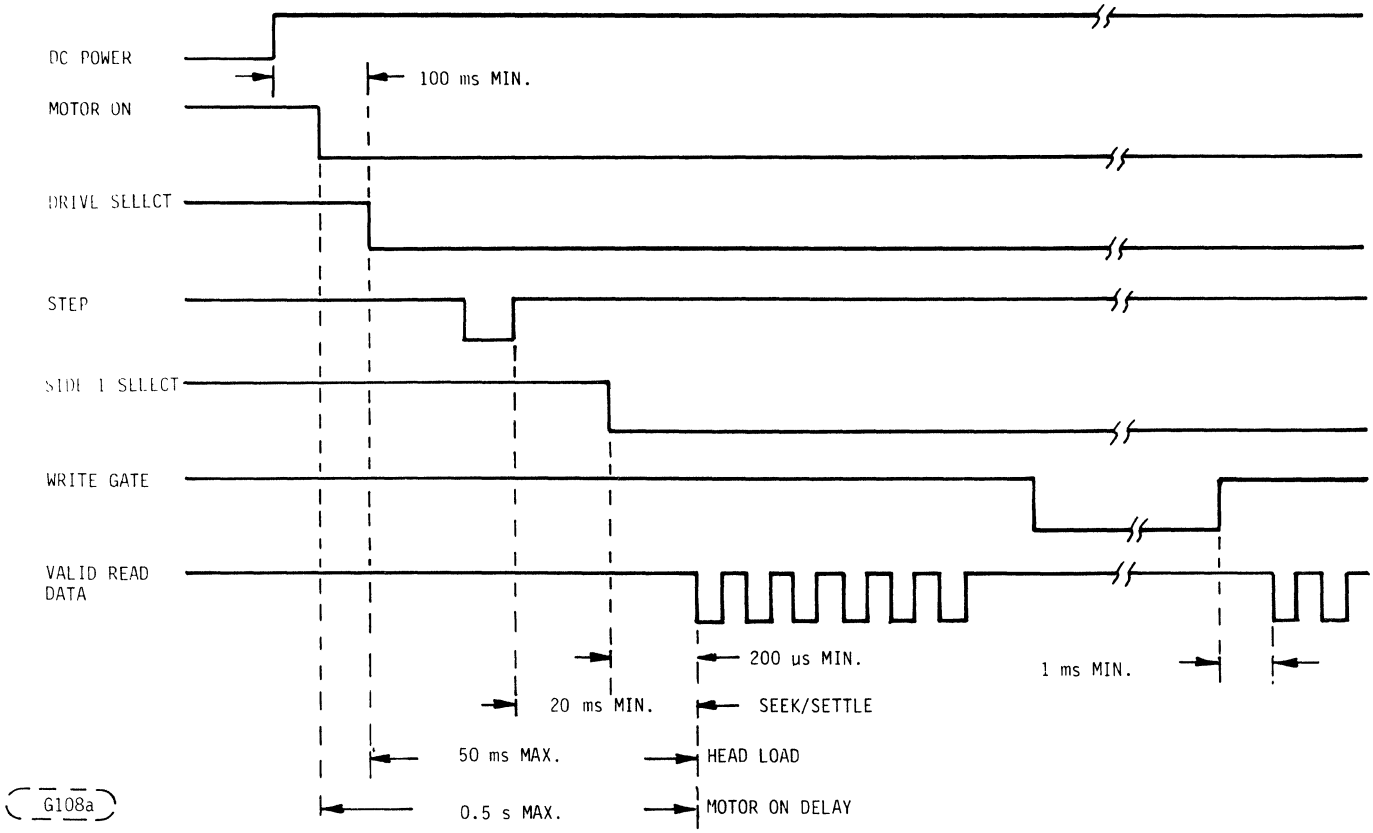


Figure 14. Read Initiate Timing

11.2.2 -contd.

a. The advantage of this configuration is that after initial motor up-to-speed delay, the drive can be selected (Drive Select line active) and there is a maximum of 2 microseconds to valid read data. (Motor must be on and head-loaded for this to be valid).

b. The disadvantage of this configuration is that with motor on and head loaded at all times, there is increased wear to drive motor, media, and heads.

2. If shunt HS is shorted and shunt HM is open, the drive motor will turn on when a logic low is applied to any of the drive select lines J1-6, -10, -12, or -14. This configuration uses the drive select line to start the motor and load the heads. The minimum delay from drive selection to valid read data is now 0.5 second due to the delay for motor start time.

11.2.2 -contd.

a. The advantage of this configuration is reduced motor, media, and head wear because motor, media, and heads are only used when needed (selected).
b. The disadvantage of this configuration is that the maximum time from drive selection to valid read data is 0.5 second.

3. Same as Configuration 2 (HS shorted and HM open), except that logic low on Motor On can start the motor. Then, at Drive Select and head is loaded and the drive is selected. If the drive is operated in this manner after an initial motor up-to-speed delay, there is a maximum 50 millisecond delay to allow for head load time until valid read data.

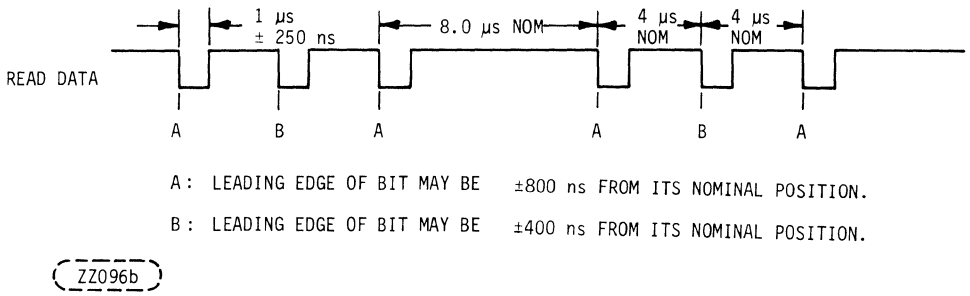


Figure 15. Read Signal Timing

11.2.3 Direction Select

Direction Select (J1-18) transfers a control signal to determine the direction the read/write head will move when the step line is pulsed.

A logic high defines the direction as "out," and if a pulse is applied to the Step line, the read/write head will move away from the center of the disk. Conversely, if this input is a logic low level, the direction of motion is defined as "in," and if a pulse is applied to the Step line, the read/write head will move toward the center of the disk. Refer to Figure 16 for timing details.

11.2.4 Step

Step (J1-20) pulses are control signals which cause the read/write head to move in a direction determined by the condition of the Direction Select line.

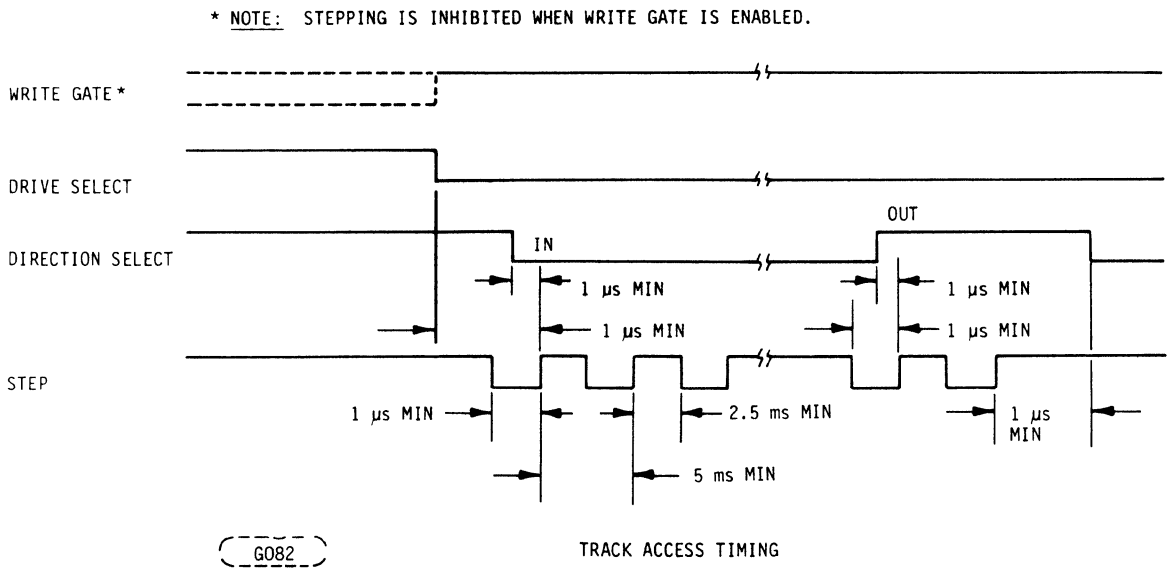


Figure 16. Track Access Timing

11.2.4 -contd.

Access motion is initiated on each logic low to logic high transition, or on the trailing edge of each signal pulse. Any change in the Direction Select line must be made at least 1 microsecond before the trailing edge of the Step pulse. A requirement exists for the Direction Select logic level to be maintained 1 microsecond after the trailing edge of the Step pulse.

Activation of the Write Gate line while using a write-protected diskette will not inhibit movement of the read/write head during transmission of Step pulses.

Refer to Figure 16 for timing details.

11.2.5 Write Gate

The active state of Write Gate (J1-24), or logic low, enables Write Data to be written on the diskette. The inactive state, or logic high, enables the Read Data logic and Step logic.

A maximum delay time of 8 microseconds is allowed between the switching sequence of the Write Gate line and the leading edge of the first data pulse to be recorded. Figure 17 illustrates this sequence of events.

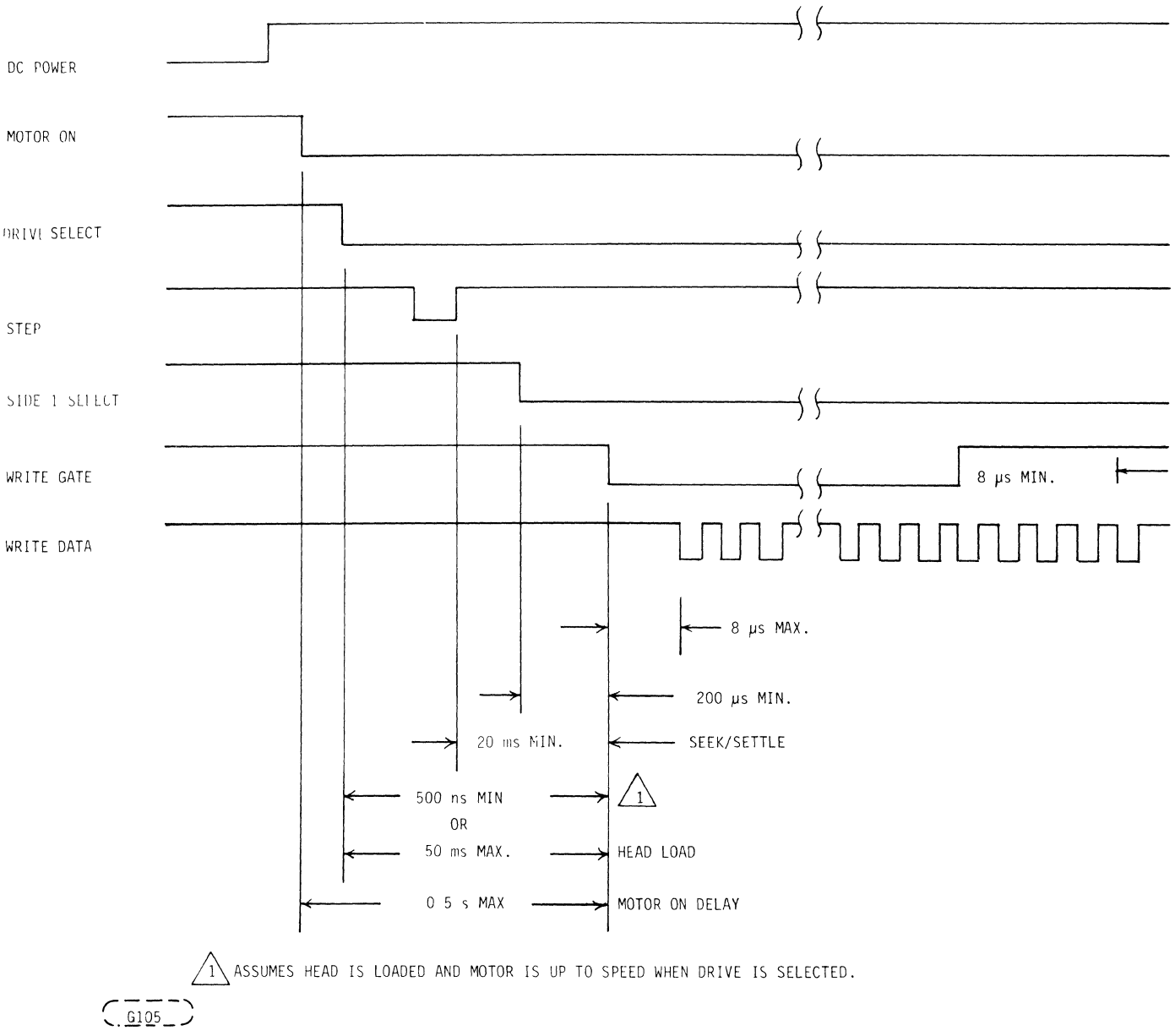


Figure 17. Write Initiate Timing

11.2.6 Write Data

Write Data (J1-22) provides data to be written on the diskette. Each pulse transition from a logic high to a logic low level, as clocked through a D flip-flop within the logic circuitry, will cause current through the read/write head to reverse, creating a data bit.

11.2.6 -contd.

Signals transmitted on the Write Data line are transmitted in the form of clock/data composite pulses. The recommended pulse width for both data bit and clock pulse is 200 nanoseconds (minimum) and 2.1 microseconds (maximum). Generation of data on the media is possible only during activation of the Write Gate to a logic low level. Figure 18 illustrates these timing sequences. Refer to paragraph 8.1.2.1 for recommended write precompensation.

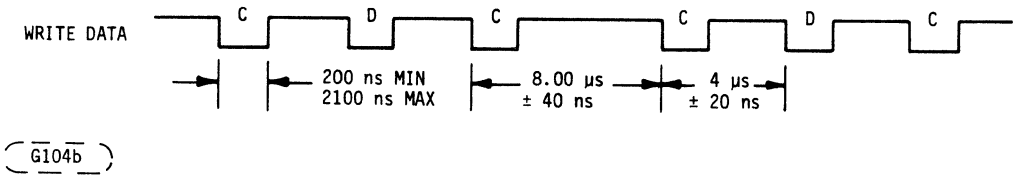


Figure 18. Write Data Timing (FM)

11.2.7 Read/Write Operations

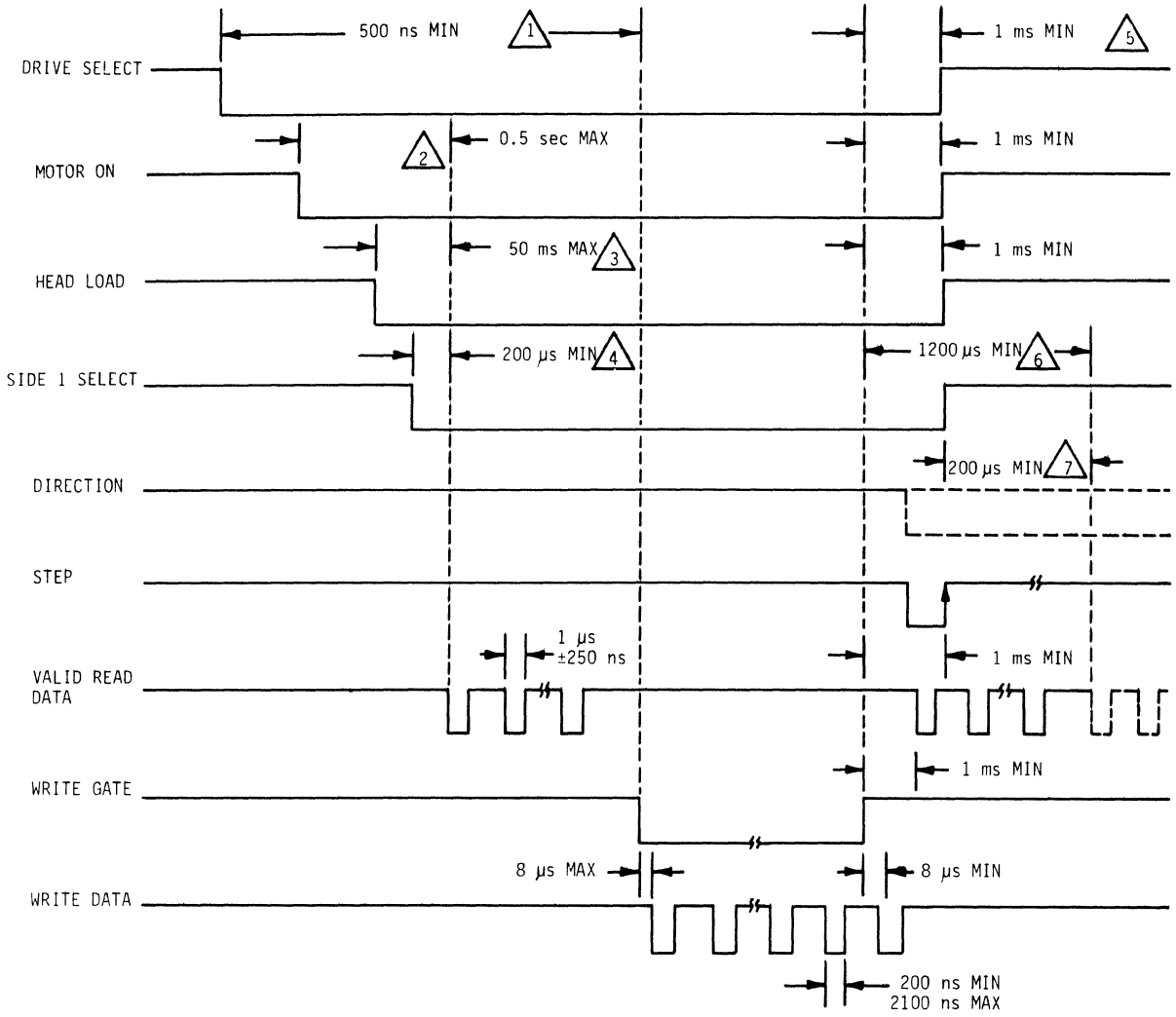
At the conclusion of a write or read operation, there are a number of timing considerations which must be observed.

1. After any write operation, there is a 1-millisecond minimum time delay to allow for the erase turn-off delay required with the tunnel erase head. Interface signals which must remain in a stable state during this delay are: a) Drive Select, b) Head Load, c) Motor on, d) Side 1 Select, and 3) Step. This can be considered as write-operation-to-read-operation time.

2. If a side change occurs at the termination of a write operation, the side change delay of 200 microseconds plus the 1-millisecond erase-turn-off delay is required; that is, a total of 1200 microseconds. This is write-to-read time including a side change.

3. In going from a read operation to a read-operation-including-a-side-change, only the side change delay of 200 microseconds must be observed.

Refer to Figure 19 for detailed timing.



1. ASSUMES MOTOR ON AND HEAD LOADED.
2. ASSUMES HEAD LOADED DURING MOTOR ON TIME DELAY.
3. ASSUMES MOTOR UP TO SPEED.
4. ASSUMES MOTOR UP TO SPEED AND HEAD LOADED.
5. THE 1 ms IS THE ERASE TURN-OFF DELAY REQUIRED FOR TUNNEL ERASE HEADS.
6. THE 1200 µs IS THE ERASE TURN-OFF DELAY PLUS SIDE CHANGE DELAY.
7. THE 200 µs IS REQUIRED FOR A CHANGE IN SIDE CHANGE.

G237

Figure 19. Read/Write Operations

11.2.8 Track 00

The active or logic low state of the Track 00 signal (J1-26) indicates the read/write head is positioned at track 00.

This signal is at a logic high level (inactive state) when the read/write head is not at track 00.

When the read/write head is at track 00, and an additional step-out pulse is issued to the drive, motion is inhibited and the Track 00 signal will remain active (logic low).

11.2.9 Index/Sector

The Index/Sector signal (J1-8) is detected and transmitted to the controller each time an index or sector hole is sensed by the index/sector photocell detector.

The signal generated is a 0.5 millisecond (min.) pulse which indicates the presence of an Index or Sector hole during its transition from a logical one to a logical zero level as shown in Figure 20.

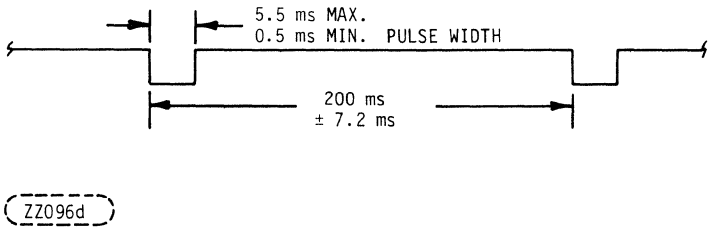


Figure 20. Index Timing (Soft-Sector Media)

11.2.9 -contd.

Using soft-sectored media, a single pulse will be generated per revolution of the diskette, every 200 ±7.2 milliseconds, indicating the physical beginning of a track.

Using hard-sectored media (16 or 10 sectors), 17 or 11 pulses, respectively, will be generated per revolution. Figures 21 and 22 provide an illustration of index/sector timings for 16 and 10 sector applications.

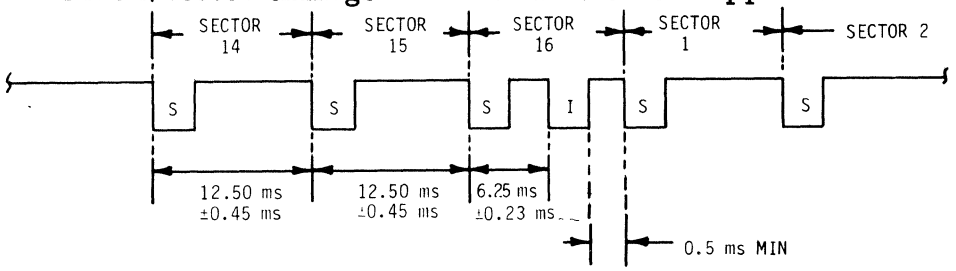


Figure 21. Index/Sector Timing (16 Hard-Sector Media)

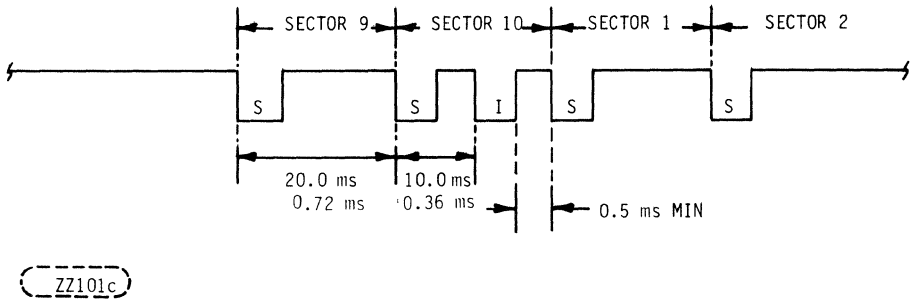


Figure 22. Index/Sector Timing (10 Hard-Sector Media)

11.2.10 Read Data

Read Data (J1-30) provides a transmission of composite clock and data pulses of 1 microsecond, ±250 nanoseconds, to indicate the presence of either a clock or data pulse by means of a logic high to a logic low transition. Figures 14 and 15 provide references for timing and bit shift tolerance within normal media variations.

11.2.11 Write Protect

Write Protect (J1-28) notifies the user that a write protected diskette (i.e., when the write protect slot on the diskette is masked) is installed in the drive. The normal configuration is that a logic low level will appear on the interface line J1-28 indicating that the diskette is write protected and writing will be inhibited.

11.2.11 -contd.

If it is necessary to invert this output signal so that an unmasked slot gives a write protect indication and inhibits writing, the user must move the black wire on the write protect switch to the unused switch pin (the one closest to the PWA). This gives write protect indication to the I/O and inhibits writing on a diskette with an unmasked slot.

11.2.12 Side 1 Select

Side 1 Select (J1-32) defines which side of a two-sided diskette is used for reading and writing.

A logic high selects side 0, the lower surface. A logic low selects side 1, the upper surface. When the selected diskette side is changed, a delay of 200 microseconds is required before any read or write operation is initiated (assuming that drive is selected, motor is on, and head is loaded).

11.2.13 In Use

In Use (J1-4) allows for the activation of the activity LED independent of Drive Select. When a logic low is applied to In Use the activity LED is turned on whether or not the drive is selected. In daisy-chain operation, all drives would have the activity LED illuminated if In Use is active.

12.0 CUSTOMER SELECTABLE OPTIONS

The 9409 contains a 14-pin program shunt module mounted on the PCB (Figure 23) which allows the drive to be selected to operate in a single drive system or a multiplexed drive system.

As shipped from the factory, each shunt position is shorted. Drive selection is accomplished by cutting selected shunt positions.

The program shunt module is AMP part number 435704-6. The shunt positions can be cut using an AMP cutting tool (part number 1-435830-1). The shunt is installed in a DIP socket and at the customer's option can be replaced by a DIP switch (AMP part number 435626-4).

Table 7 provides a listing of optional configurations which can be utilized by the customer.

12.1 MULTIPLEX OPTION

The 9409 is shipped from the factory configured to operate in a single drive system. To activate the multiplex option, cut the MX (5-10) position of the programmable shunt. This will allow the multiplexing of I/O lines. For single drive applications, shorting MX will cause a constant enable on the I/O signal lines whenever the drive is powered on.

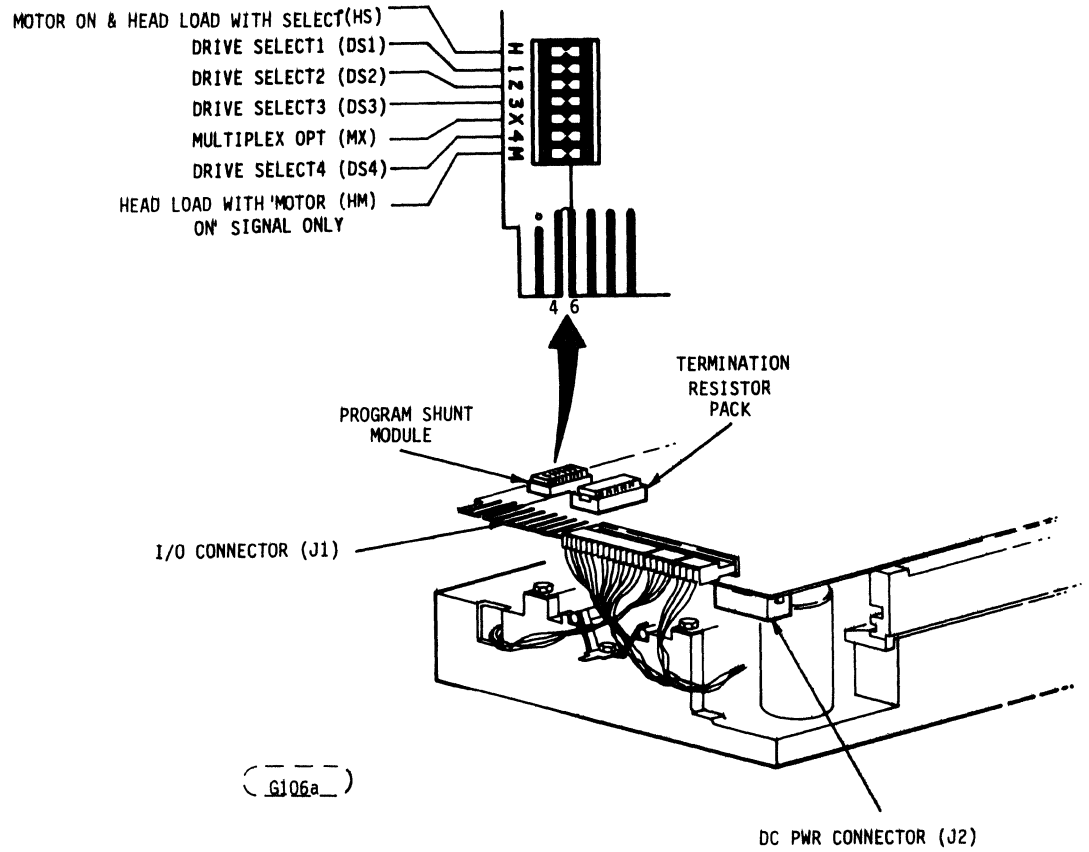


Figure 23. Program Shunt Module

12.2 DRIVE SELECT OPTIONS (1, 2, 3, 4)

The 9409 utilizes four input lines for designation of a particular drive. With MX (5-10) cut, positions DS1 through DS4 are used to select the Drive Select line which will activate the unique drive. Only the drive with its Drive Select line activated will respond to the input lines, gate the output lines, and turn on the indicator located on the front panel. (Exception: The In Use line will turn on the indicator independent of all other lines.) For example, in designating a drive as Drive 1, the user must cut Drive Select shunts DS2, DS3, DS4 (pins 3-12, 4-11, and 6-9, respectively) and leave DS1 (pins 2-13) shorted. Refer to Figure 24 for Drive Select shunt configurations.

12.3 HEAD LOAD WITH MOTOR ON

With HM (7-8) shorted and HS (1-14) open*, the read/write head will load against the media when the Motor On line is activated to a logic low.

12.4 HEAD LOAD WITH DRIVE SELECT

With HS (1-4) shorted and HM (7-8) open* the head load solenoid will energize with Drive Select activated and load the read/write head.

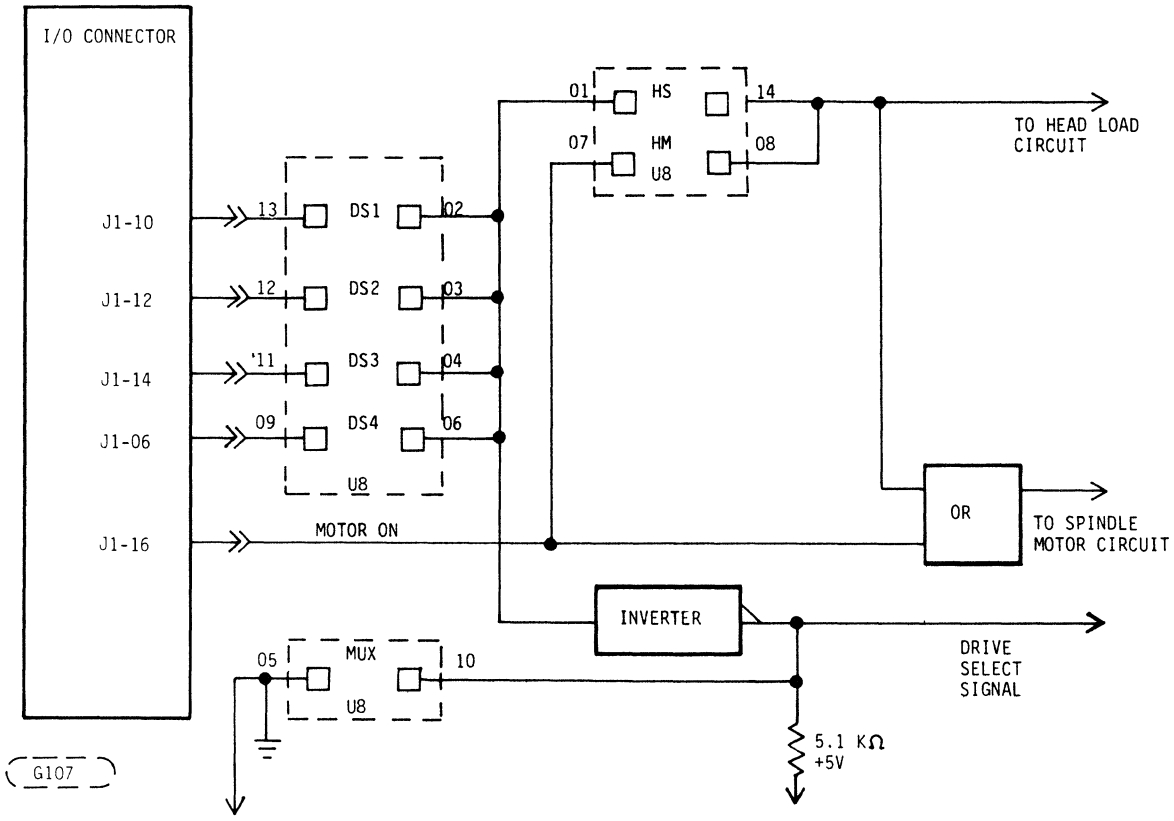
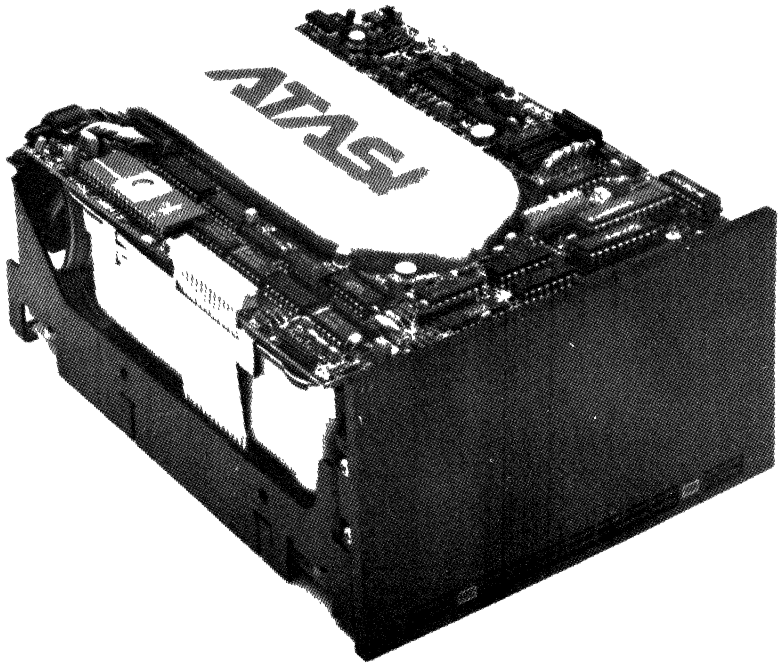


Figure 24. Drive Select Shunt Configuration

* It is an invalid configuration to have both HS (1-14) and HM (7-8) shorted. In such a configuration, input current to Motor On and/or Drive Select may damage drivers on the host.

Table 7. Customer Option Configurations

MX	HS	HM	DS1	DS2	DS3	DS4	NOTES
X	0	0	X	X	X	X	Invalid configuration. Head will not load.
0	0	S	S 0 0 0	0 S 0 0	0 0 S 0	0 0 0 S	• Read/write head is loaded and motor is turned on with a low level on Motor On. • Activity Indicator activated (if In Use remains at a high level) and drive is selected by a low level on Drive Select.
0	S	0	S 0 0 0	0 S 0 0	0 0 S 0	0 0 0 S	• Read/write head is loaded, motor is turned on (if Motor On remains at a high level). • Activity Indicator is activated (if In Use remains at a high level), and drive is selected by a low level on Drive Select.
S	0	S	S 0 0 0	0 S 0 0	0 0 S 0	0 0 0 S	• Drive is always selected. • Activity Indicator is activated (if In Use remains at a high level) by a low level on Drive Select. • Heads load and motor is turned on with a low level on Motor On.
S	S	0	S 0 0 0	0 S 0 0	0 0 S 0	0 0 0 S	• Drive is always selected. • Read/write head is loaded, motor is turned on (if Motor On remains at a high level), and Activity Indicator is activated by a low level on Drive Select (if In Use line is held high).
X	S	S	X	X	X	X	Invalid Configuration. Input current to Motor On and/or Drive Select may damage drivers on host.
S = SHORT					O = OPEN		X = DON'T CARE



SERIES 3000
OEM MANUAL

E-43

TABLE OF CONTENTS

FIGURES:

SECTION 1 — INTRODUCTION

1.1 GENERAL DESCRIPTION

The ATASI Series 3000 DISK DRIVES are random access storage devices with two, three or four non-removable 5¼ inch disks as storage media. Each disk surface employs one moveable head to service its data tracks and one disk surface is dedicated to continuous servo positioning data.

High performance and high unit reliability are achieved through the use of a linear voice coil and a closed loop servo positioning system. The inherent simplicity of mechanical construction and electronic controls allows for maintenance free operation throughout the life of the drive. The electronic PWA's are mounted outside the head disk area for quick and simple field serviceability.

Mechanical and contamination protection for the heads, disk and actuator are provided by a Disk Drive Module, (DDM) which is a sealed mechanical enclosure with a self contained recirculating air filtration system that supplies clean air and temperature equalization throughout the DDM.

Shock and thermal isolation are provided by the combination of a heavy duty steel chassis to which the DDM is shock-mounted, and internal isolation within the DDM.

The bottom surface of the lowest disk contains the continuous servo data, utilized to ensure precise positioning of the read/write heads.

The ATASI Series 3000 uses a ST506/4XX compatible interface.

The unit size and mountings are identical to the industry standard mini-floppy disk drives and uses the same dc voltages and connector.

KEY FEATURES:

- 30 Msec average seek time. (+3.0 Msec settling)
- Storage capacities of 19.84, 33.07 and 46.30 megabytes unformatted.
- Spare Tracks. (10 per surface)
- Dedicated disk servo surface with servo guard band.
- Dedicated landing zone.
- Patented linear motor/closed loop servo head positioning.
- Double shock isolation.
- Thermal isolation.
- Industry Standard 5¼-inch physical size and mountings.
- Same DC voltages as Standard 5¼-inch drive.
- ST506/4XX compatible interface.
- 5.0 Mbit/sec data transfer rate.

1.2 DEVICE SPECIFICATION

1.2.1 ENVIROMENTAL SPECIFICATIONS

Environmental Limits:

Operating Temperature	10° to 50° C
Non Operating Temperature	-40° to 60° C
Operating Humidity	10% to 80%
Non Operating Humidity	5% to 95%
Maximum Wet Bulb	25° C (Non-condensing)
Thermal Gradient	10° C per hour
Operating Altitude	0 to 10,000 feet
Operating Vibration	.5G at 10-500 Hz
Non Operating Shock	30Gs

Voltage Requirements:

+5VDC± 5% 1.0 Amp typical

+12VDC±5% 4.5 Amps max (starting for 15 sec) 2.5 Amps typical

(For more information see Figure 12)

1.2.2 RELIABILITY SPECIFICATIONS

MTBF=12,000 POH

PM=NONE REQUIRED

MTTR=30 Minutes

Component Design life=5 Years

ERROR RATES:

Soft read errors=1 per 10/10th bits read

Hard read errors=1 per 10/12th bits read*

Seek errors=1 per 10/6th seeks

*Not recoverable within 16 re-tries

1.2.3 PERFORMANCE SUMMARY

Model	3020	3033	3046
Capacity Unformatted (+ 10 spare Cylinders)			
Per Drive	19.84MB	33.07MB	46.30MB
Per Surface	6.61MB	6.61MB	6.61MB
Per Track	10.416KB	10.416KB	10.416KB
Capacity Formatted (+ 10 Spare Cylinders)			
Per Drive	15.60MB	26.00MB	36.40MB
Per Surface	5.20MB	5.20MB	5.20MB
Per Track	8.192KB	8.192KB	8.192KB
Per Sector	256Bytes	256Bytes	256Bytes
Sectors/Track	32	32	32
Transfer Rate	5Mbit/sec	5Mbit/sec	5Mbit/sec
Seek Time			
Track to Track	3.0ms	3.0ms	3.0ms
Average	30.0ms	30.0ms	30.0ms
Maximum	60.0ms	60.0ms	60.0ms
Settling	3.0ms	3.0ms	3.0ms
Average Latency	8.33ms	8.33ms	8.33ms
Start Time	15 Sec	15 Sec	15 Sec

NOTE: The access times specified above are typical over a large number of positionings. Due to the nature of a track following servo and some customer unique parameters they should not be used as absolute maximum values.

1.2.4 FUNCTIONAL SUMMARY

Rotation +/-1%	3600rpm	3600rpm	3600rpm
Recording max	8780bpi	8780bpi	8780bpi
Flux Density	8780fci	8780fci	8780fci
Track Density	800tpi	800tpi	800tpi
Data Cylinders	645	645	645
Tracks	1905	3175	4445
R/W Heads	3	5	7
Disks	2	3	4
Index	1	1	1

SECTION 2 — GENERAL OPERATION

2.1 **Organization**

There are three basic functions that are required by a Disk Drive. They are, to position the Read/Write heads over the desired track, to Write Data, and to Read Data. In the ATASI 3000 drives this is done with the following electro-mechanical assemblies:

- Read/Write/Interface PWA
- Spindle/EMA Drive PWA
- Servo Control PWA
- Spindle Drive Mechanism
- Air Filtration System
- Positioning Mechanism
- Read/Write Heads and Media

2.2 **READ/WRITE/INTERFACE PWA**

The READ/WRITE/INTERFACE PWA, to which all power, control and data signals are connected, provides the following functions:

- POWER reception and internal voltage regulation.
- CONTROL INPUT SIGNAL reception and Internal distribution.
- CONTROL OUTPUT SIGNAL accumulation, sequencing and transmission.
- READ/WRITE SIGNAL, bi-directional reception, conditioning and transmission.
- FAULT detection and FAULT SIGNAL generation.

2.3 **SPINDLE/EMA DRIVE PWA**

The SPINDLE/EMA DRIVE PWA contains a dedicated MICROCOMPUTER and is mounted directly to the bottom of the DDM mechanical enclosure. It derives its power from the R/W/INTERFACE PWA and provides the following functions:

- Power and speed control to the spindle drive motor.
- Power drive to the voice coil actuator, Electro Magnetic Actuator (EMA).

2.4 **SERVO CONTROL PWA**

The SERVO CONTROL PWA contains a dedicated MICROCOMPUTER and is mounted to the top cover of the DDM mechanical enclosure, and provides the following functions:

- Signal sequence control and monitoring during the power up operation.
- Receives the SERVO DATA that is read from the dedicated SERVO DISK surface by the SERVO HEAD.
- Conditions the SERVO DATA and generates POSITION SIGNALS.
- Distinguishes between STEP and BUFFERED MODE seeks. In the case of BUFFERED MODE, generate, detect and control the carriage velocity to ensure the optimum arrival at the desired cylinder.
- Continuous position control while on track.

2.5 **SPINDLE DRIVE MECHANISM**

A brushless DC drive motor rotates the spindle at 3600 RPM ± 1%. The motor is thermally isolated from the baseplate to minimize the temperature transfer. The motor, spindle and disk stack are dynamically balanced to eliminate vibration. A dedicated MICROCOMPUTER provides complete digital control of the spindle rotation and permits algorithm control of motor start and stop.

2.6 **AIR FILTRATION SYSTEM**

The disks and read/write heads are fully enclosed in the DDM module using an integral recirculating air system with an absolute filter to maintain a clean environment. Integral to the filter is a port which permits pressure equalization with the ambient air.

2.7 **POSITIONING MECHANISM**

The read/write heads are mounted on a ball bearing supported linear carriage which is positioned by a linear voice coil motor, driven by the closed loop servo system.

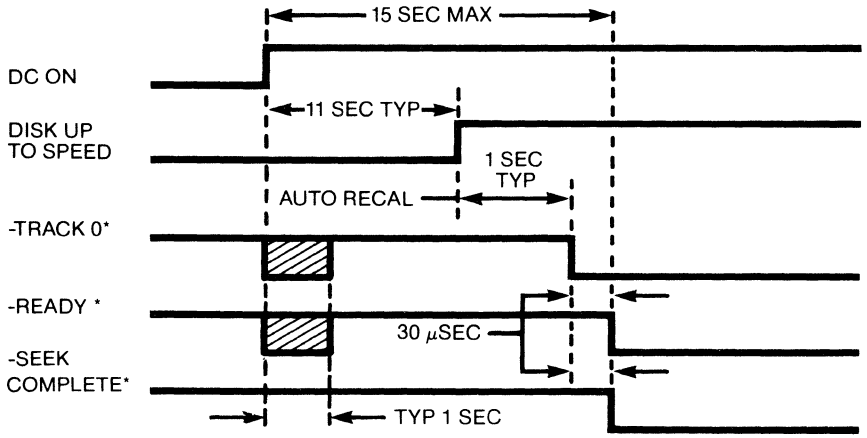
2.8 **READ/WRITE HEADS AND MEDIA**

The recording media consists of a lubricated thin magnetic oxide coating on a 130 mm diameter aluminum substrate. This coating formulation, together with the low load force/low mass Winchester type heads, permit reliable contact start/stop operations. Data on each of the data surfaces is read or written by one read/write head. Each head accesses 645 data cylinders.

SECTION 3 — FUNCTIONAL OPERATION

3.1 POWER SEQUENCING

The +5 and +12 volts dc may be applied in any order. +12 volts must be applied to start the spindle drive motor. A MICROCOMPUTER monitors the disk rotation. At 3600 +/–1%, the heads will automatically recalibrate to track 00. Under normal operation, the signal -TRACK 0 will precede the signals -READY and -SEEK COMPLETE by typically less than 30 μsec. The -READY signal is inhibited or disabled by any fault condition. The disk drive can only perform read/write or seek functions following the setting of the -READY signal. (see Figure 1 for the signal sequence).



* -TRACK 0, -READY and -SEEK COMPLETE will not be present at the interface unless the drive is selected

Figure 1 POWER UP SEQUENCE

3.2 DRIVE SELECTION

Drive selection occurs when one of the -DRIVE SELECT lines is activated. Only the respective drive selected will respond to the signals from the controller interface. There is also a radial select line which permanently selects the drive (See Section 4.6).

3.3 CYLINDER ACCESSING

Read/Write head positioning is accomplished by:

- Activating the respective -DRIVE SELECT line.
- -READY condition with -SEEK COMPLETE.
- No FAULT conditions exist.
- Selecting the correct -DIRECTION.
- Pulsing the -STEP line.

Each -STEP pulse will move the heads 1 cylinder. Motion is inward or outward depending on the state of the -DIRECTION IN signal. A high state indicates inward toward the spindle and a low indicates outward from the spindle. Seeks can be done in 3ms steps, or in the buffered mode (see 4.1.4).

3.4 HEAD SELECTION

Read/write heads may be selected by placing the respective binary address on the -HEAD SELECT lines.

3.5 READ OPERATION

Reading data is accomplished by:

- -DRIVE SELECT.
- Inactive -WRITE GATE.
- No FAULT conditions.
- Disk drive is -READY and -SEEK COMPLETE.
- -HEAD SELECT for the appropriate head.
- Present MFM READ DATA to the host controller.

3.6 WRITE OPERATION

Writing data is accomplished by:

- -DRIVE SELECT.
- Drive is -READY and -SEEK COMPLETE.
- -HEAD SELECT for the appropriate head.
- No FAULT conditions.
- Activate -WRITE GATE and receive MFM WRITE DATA from the host controller.

SECTION 4 — ELECTRICAL INTERFACE

The disk drive interface is divided into three catagories, each of which is physically separate.

- CONTROL SIGNALS
- DATA SIGNALS
- DC POWER

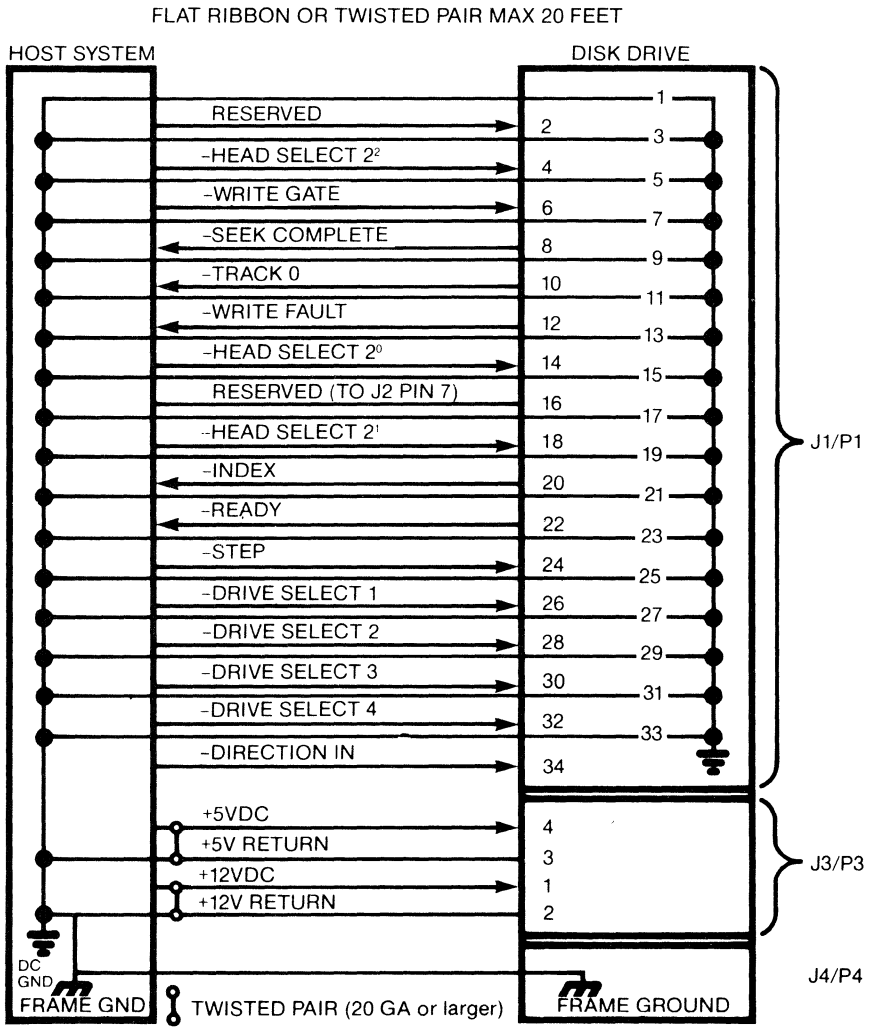


Figure 2 CONTROL SIGNALS

All control signals are digital (open collector TTL) and either provide signals to the drive from the host controller or to the host controller from the drive via connector J1/P1. The data transfer signals are differential and are connected via connector J2/P2. DC power utilizes connector J3/P3. The one exception to the above is -DRIVE SELECTED which is a digital signal, but is transmitted via J2/P2. Figures 2 through 4 show connector pin assignments and interconnection of cabling between the disk drives and the host controller.

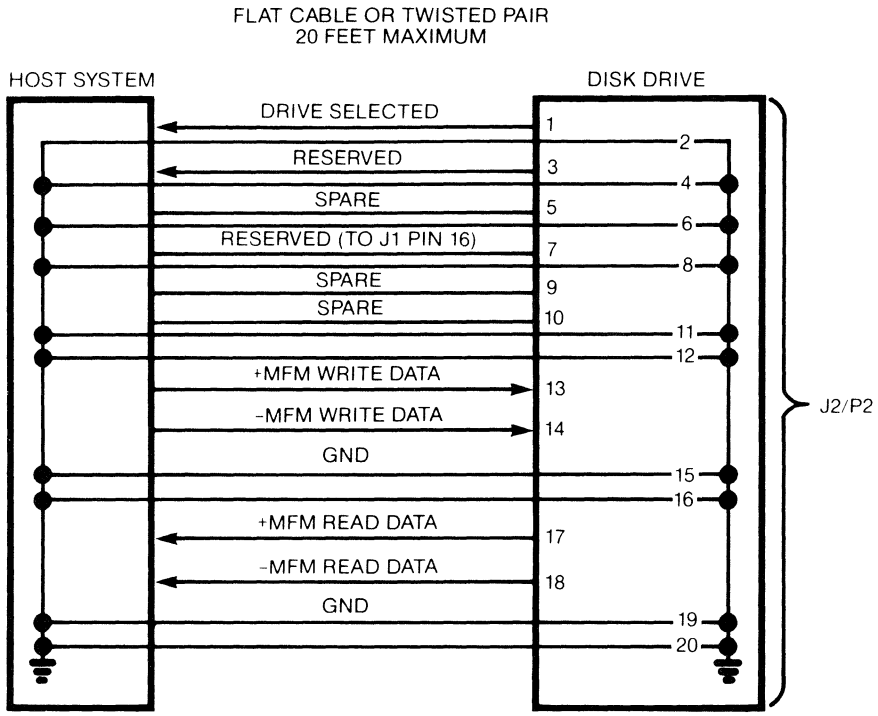
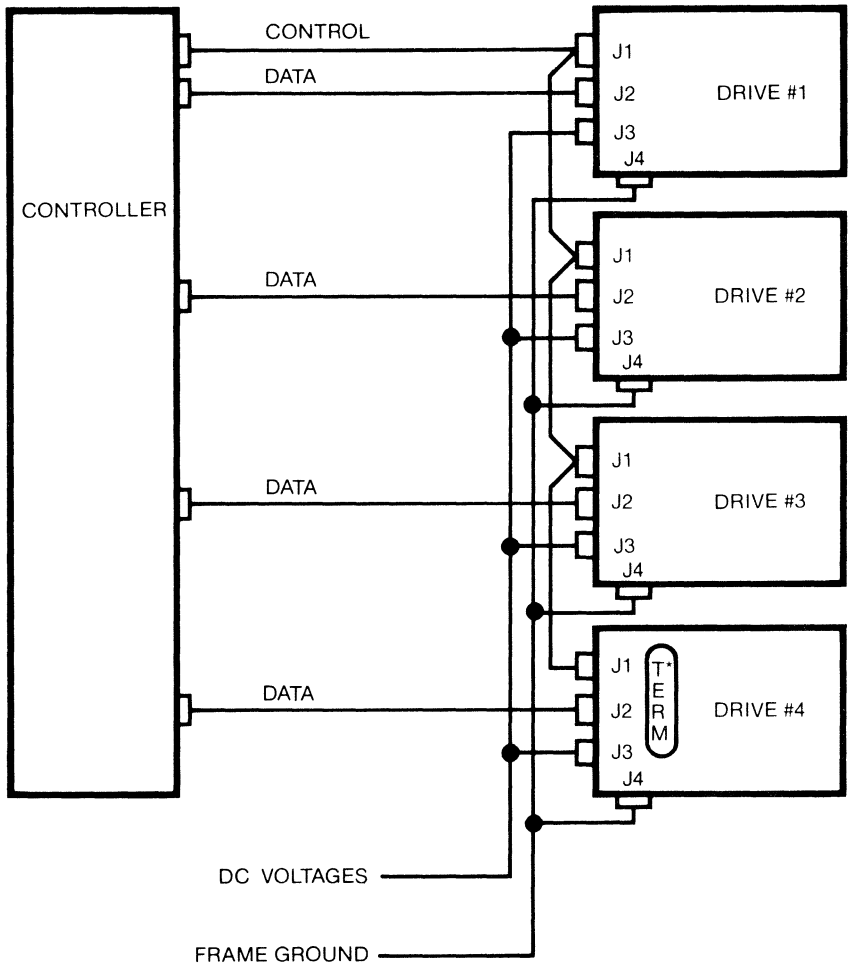


Figure 3 DATA SIGNALS



* The last or only drive in the control cable string must have the terminator resistor pack installed All other drives must have their terminators removed

Figure 4 TYPICAL CONNECTION, 4 DRIVES

E-49

4.1 **CONTROL INPUT LINES**

The control input signals are of two types: Those to be multiplexed in a multiple drive system and those intended to do the multiplexing. The multiplexed lines are -WRITE GATE, -HEAD SELECT, -STEP and -DIRECTION IN. The multiplexer is -DRIVE SELECT. The active state for all of these lines is low or 0–+.7VDC. The inactive state is high OR +2–+5VDC. The control input lines require the following specifications. (see Figure 5 for recommended circuitry).

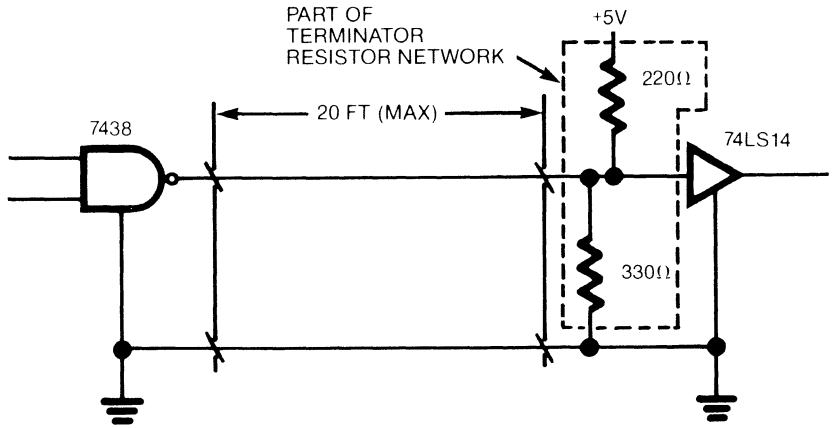


Figure 5 CONTROL SIGNAL RECEIVER/DRIVER COMBINATION

4.1.1 **-WRITE GATE**

This line, when active, enables write data to be written on the disk. The inactive state enables data to be read from the disk. The inactive state also enables seek operations.

4.1.2 **-HEAD SELECT 2^0, 2^1, 2^2**

These lines provide a means to select R/W heads in a binary coded sequence. When all -HEAD SELECT lines are inactive, HEAD 0 is selected. When the drive is not selected, R/W heads are deselected. An illegal head address will deselect all heads.

4.1.3 **-DIRECTION IN**

This signal defines the direction of the carriage and R/W head movement when the -STEP line is pulsed. An active -DIRECTION IN defines a seek toward the spindle. An inactive -DIRECTION IN defines a seek away from the spindle. (see Figures 6A and 6B for signal timing).

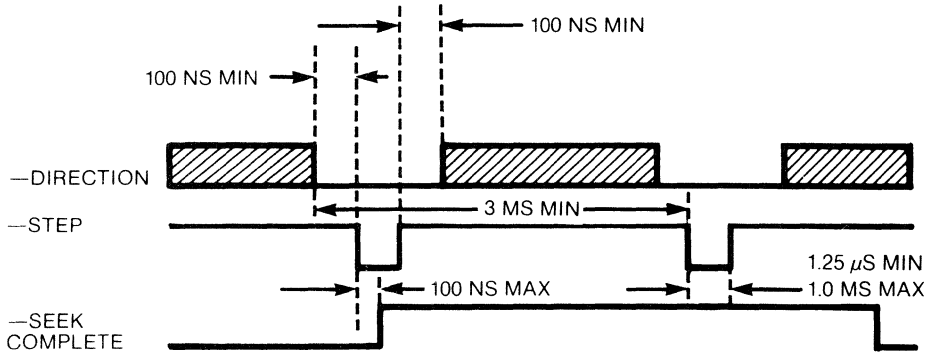


FIGURE 6A STEP MODE TIMING (NORMAL)

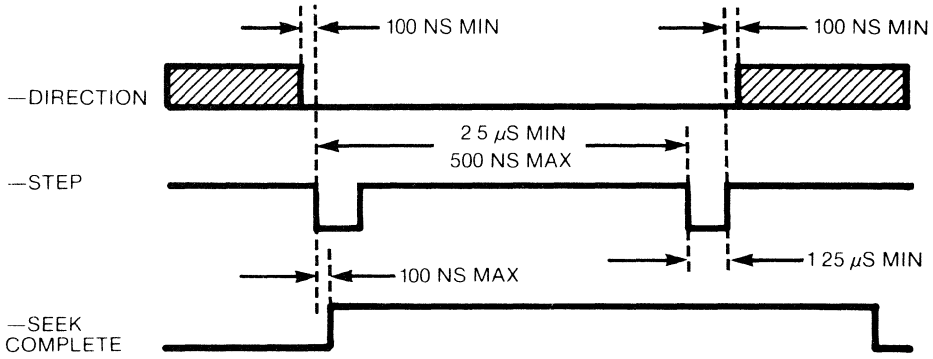


FIGURE 6B STEP MODE TIMING (BUFFERED)

4.1.4 **-STEP**

This signal causes the R/W heads to move in the direction indicated by DIRECTION IN. There are two modes of STEP operation, NORMAL and BUFFERED. In the NORMAL MODE the -STEP pulses may occur at intervals not less than 3 msec. The width of the pulses may range from 1.25 μsec to 1.0 msec. In the BUFFERED MODE the -STEP pulses can occur at intervals from 2.5 μsec to 500 μsec. The minimum pulse width in BUFFERED MODE is 1.25 μsec (See Figures 6A and 6B for timing).

4.1.5 **-DRIVE SELECT 1-4**

This signal, when active, enables the respective drive interface signals to communicate with the host controller. Addresses are customer selectable internal to the drive via S1-4 on the READ/WRITE/ INTERFACE PWA. (see Figure 7 for jumper locations).

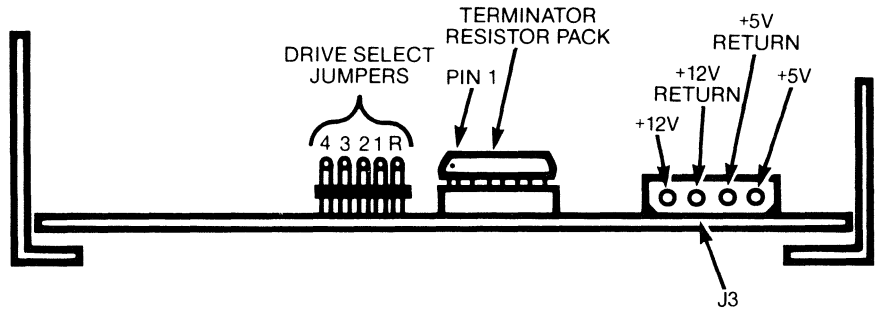


Figure 7 SELECT JUMPER, TERMINATOR AND J3 LOCATIONS

4.2 **CONTROL OUTPUT LINES**

The control output lines indicate the operational status, timing and functional response of the disk drive. These signals are enabled to the host controller, only while the drive is selected. (see 4.6 for the exception).

4.2.1 **-SEEK COMPLETE**

This signal becomes active when the R/W heads have settled on the desired cylinder at the end of a SEEK. READ or WRITE operations may not be initiated until -SEEK COMPLETE is true. -SEEK COMPLETE will go inactive within 100 nsec after the leading edge of a -STEP pulse, or the first in a series of -STEP pulses, or if +5 VDC or +12 VDC are lost momentarily.

4.2.2 **TRACK 0**

This signal becomes active when the disk drive's R/W heads are positioned at TRACK 0, the outermost data track.

4.2.3 **-WRITE FAULT**

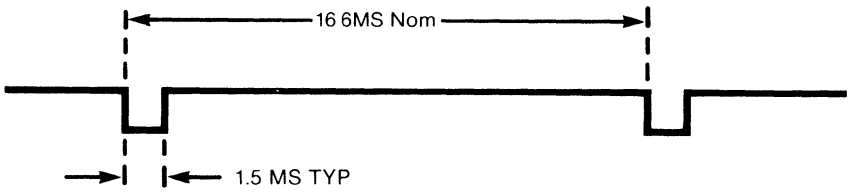
This signal becomes active when one of the following conditions exists in a selected disk drive:

(a) -WRITE GATE true and NO WRITE CURRENT.
(b) WRITE CURRENT present without -WRITE GATE.
(c) Multiple heads are selected.
(d) When a valid selected head is shorted or open
(e) When-WRITE GATE is active with NO WRITE DATA

-WRITE FAULT is latched in the drive and can be cleared by a power down or deselection of the drive

4.2.4 **-INDEX**

This signal is presented to the host controller from a selected disk drive once for each revolution of the disk, and represents the beginning of a track. (see Figure 8 for timing).



NOTE The only valid way to capture index is on the leading edge

Figure 8 INDEX TIMING

4.2.5 **-READY**

This signal in combination with -SEEK COMPLETE indicates that the selected disk drive is READY to READ, WRITE or SEEK.

4.3 **DATA TRANSFER LINES**

These DATA TRANSFER LINES are DIFFERENTIAL in design, consisting of two pairs of balanced signals used to transfer MFM DATA to and from the selected disk drive. (see Figure 9 for recommended circuitry).

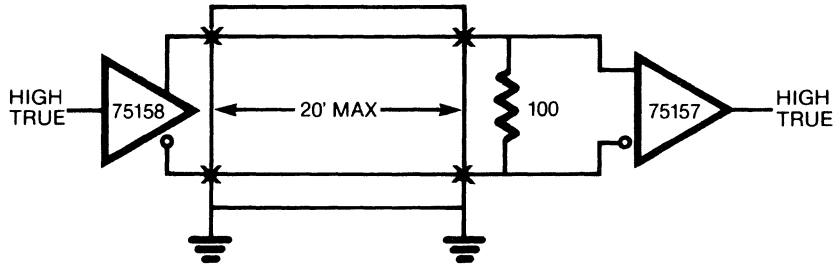


Figure 9 DATA LINE DRIVE/RECEIVER COMBINATION

4.3.1 **+/-MFM WRITE DATA**

These two differential MFM (Modified Frequency Modulation) signal lines define the code to be written on the track. The transition of +MFM WRITE DATA going more POSITIVE than -MFM WRITE DATA causes a transition, or flux reversal on the disk by the selected head. To ensure data integrity at the error rate specified, the WRITE DATA transmitted by the host controller must be PRE-COMPENSATED from cylinders 320 through 645. Optimum pre-comp of both early and late data is 12 nsec. (see Figure 10A for MFM ENCODING and Figure 10B for WRITE PRE-COMP patterns. All patterns not represented in Figure 10B are written on-time).

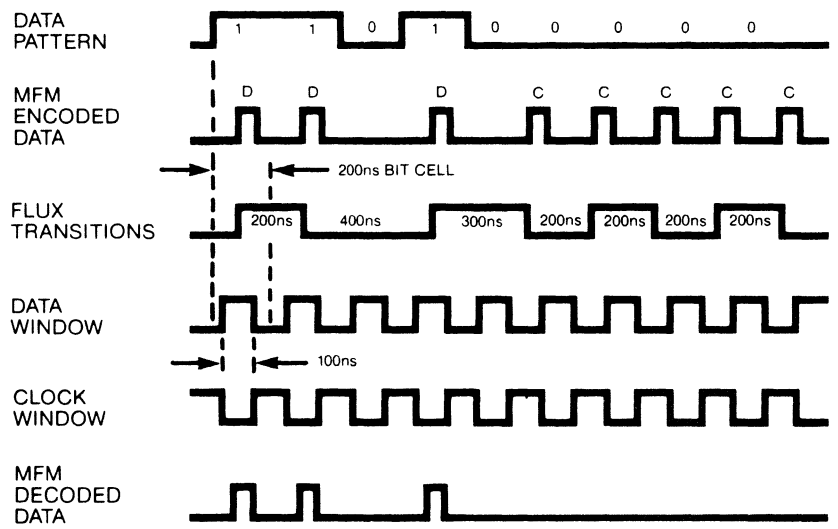


Figure 10A MFM ENCODING

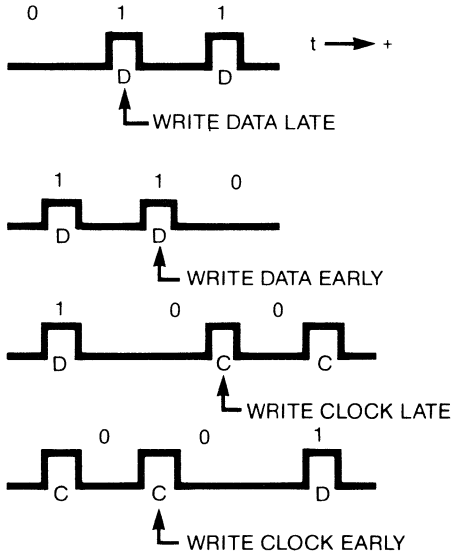


Figure 10B WRITE PRE-COMPENSATION PATTERNS

4.3.2 **+/- MFM READ DATA**

The transitions, "DATA" sensed by the selected head from a pre-recorded track are transmitted to the host controller as MFM READ DATA. The transition of +MFM READ DATA going more positive than -MFM READ DATA indicates the sensing of a flux reversal on the disk by the selected head.

4.3.3 **READ/WRITE TIMING**

READ and WRITE functions require proper signal sequencing to ensure DATA INTEGRITY. (See Figure 11 for proper signal timing).

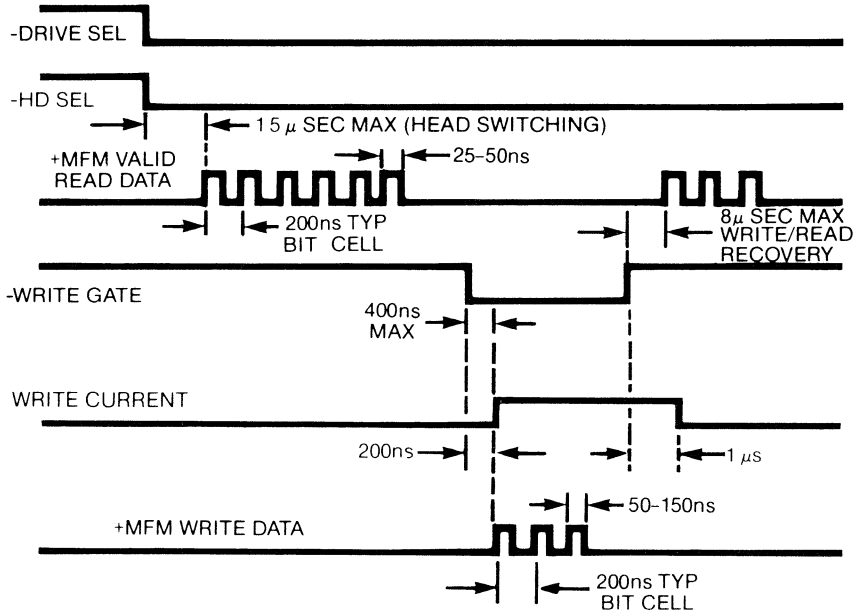


Figure 11 READ/WRITE DATA TIMING

4.4 **-DRIVE SELECTED**

The -DRIVE SELECTED line becomes true when the associated -DRIVE SELECT matches the PHYSICAL ADDRESS of the disk drive, or the radial option is selected.

4.5 POWER INTERFACE

The voltages required to operate the drive are +5VDC and +12 VDC (See Figures 12 and 13 for more detail on the current requirements.)

VOLTAGE	MAX START	TYP START	MAX SEEK-ING	TYP SEEK-ING	MAX STEADY STATE	TYP STEADY STATE	MAX RIPPLE P-P
+5	1.5 AMP	1 AMP	1.5 AMP	1 AMP	1 5 AMP	1 AMP	50mV
+12	4.5 AMP	3 8 AMP	3 AMP	2.5 AMP	2 AMP	1.5 AMP	50mV

Figure 12 CURRENT REQUIRMENTS

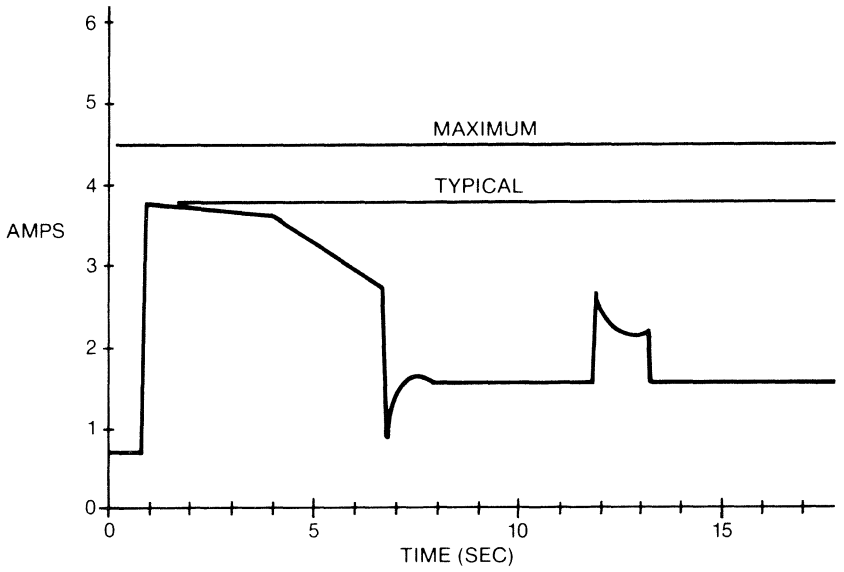


Figure 13 +12V STARTING CURRENT

4.6 CUSTOMER OPTION

A RADIAL interface option is available and is CUSTOMER SELECTABLE by placing a jumper in LOCATION "R" on the READ/WRITE/INTERFACE PWA. With this jumper installed, the interface output signals are enabled at all times. The DRIVE SELECTED "LED" will illuminate only when the drive receives a valid -DRIVE SELECT, ie, in this case, there must be two jumpers installed, one on "R" and one on 1, 2, 3, or 4.
(See Figure 7 for jumper location)

SECTION 5 — PHYSICAL INTERFACE

The interface between the disk drive and the host controller consists of four connections:

J1 — CONTROL SIGNALS

J2 — READ/WRITE SIGNALS

J3 — DC POWER

J4 — FRAME GROUND

(see Figure 14 for connector locations).

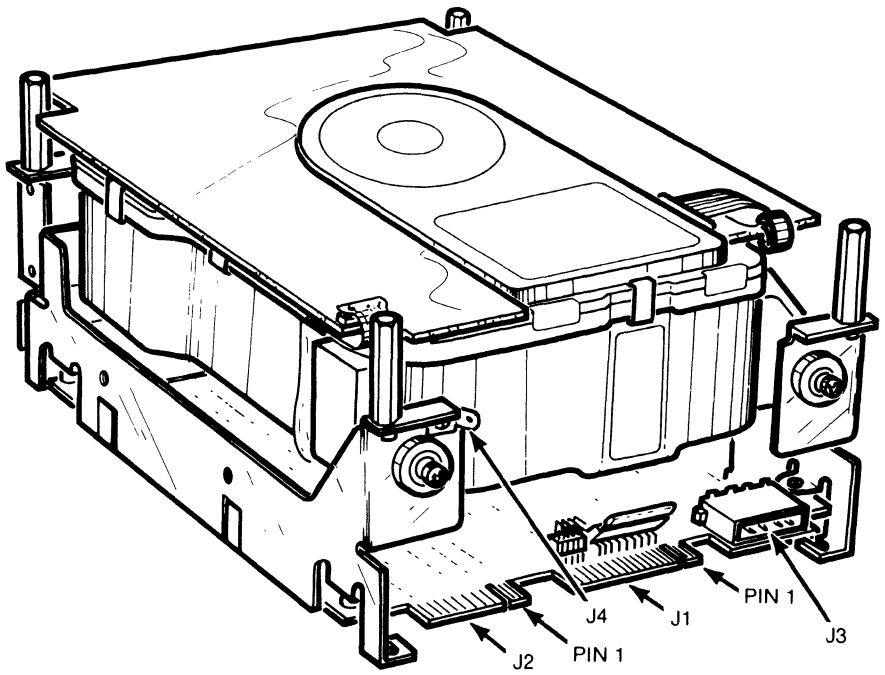


Figure 14 CONNECTOR LOCATIONS

5.1 J1/P1 CONNECTOR—CONTROL SIGNALS

Connection to J1 is through a 34 pin PWA edge connector. The dimensions for this connector are shown in Figure 15. The pins are numbered 1 through 34 with the even pins on the solder side of the PWA. A key slot is provided between pins 4 and 6. The recommended mating connector for J1 is AMP Ribbon connector p/n 88383-3. All odd pins are ground.

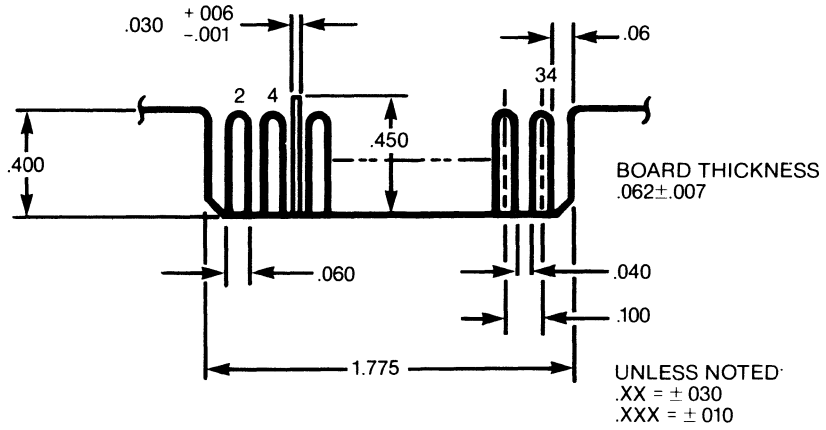


Figure 15 J1 CONNECTOR DIMENSIONS

5.2 J2/P2 CONNECTOR—DATA SIGNALS

Connection to J2 is through a 20 pin PWA edge connector. The dimensions for the connector are shown in Figure 16. The pins are numbered 1 through 20, with the even pins located on the solder side of the PWA. The recommended mating connector for J2 is AMP p/n 88737-6. A key slot is provided between pins 4 and 6.

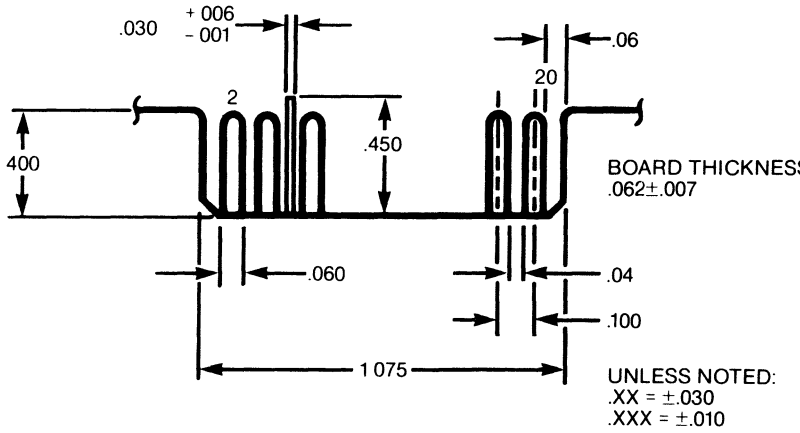
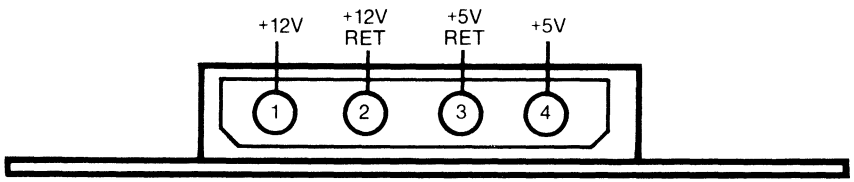


Figure 16 J2 CONNECTOR DIMENSIONS

5.3 **J3/P3 CONNECTOR—DC POWER**

DC power connector J3 is a 4 pin AMP MATE-N-LOCK connector, AMP p/n 350211-1 and is mounted on the component side of the PWA. The recommended mating connector for J3 is AMP p/n 1-480424-0. J3 pins are numbered as shown in Figure 17.

CAUTION: Damage will occur to the drive if the +5V and +12V connections are reversed.



NOTE This is the Drive end of the connector

Figure 17 J3 CONNECTOR LAYOUT

5.4 **J4/P4 FRAME GROUND**

Recommended mating connector AMP 62187-1.

NOTE: DC GROUND is isolated from FRAME GROUND. Frame ground connections are very important to reduce the effects of ground loops and noise which can effect DATA INTEGRITY.

SECTION 6 — PHYSICAL SPECIFICATIONS

6.1 **MOUNTING ORIENTATION**

Recommended mounting is either vertical on either side or horizontal. The only PROHIBITED MOUNTINGS are INVERTED HORIZONTAL or INCLINED such that the actuator is accessing on a NON-LEVEL PLANE. In final mounting, it is IMPORTANT to ENSURE that the SHOCK MOUNTS that ISOLATE the CHASSIS from the FRAME are NOT RESTRICTED.

6.2 **MOUNTING HOLES**

Eight standard mounting holes are provided, four on the bottom and two on each side. The size and location of the mounting holes are identical to the standard mini-floppy drive. Additional chassis attachment points are provided near the corners of the top and bottom surfaces which may be used for mounting or stacking of drives. (See Figure 18 for dimensions)

6.3 **PHYSICAL DIMENSIONS**

Height 3.25 inches, width 5.75 inches and depth 8.00 inches are identical to the standard mini-floppy, allowing a direct physical replacement. (See Figure 18 for dimensions).

6.4 **SHIPPING REQUIREMENTS**

During shipping or transporting, the R/W heads are positioned at the LANDING/SHIPPING ZONE. This is accomplished automatically by removing power to the drive.

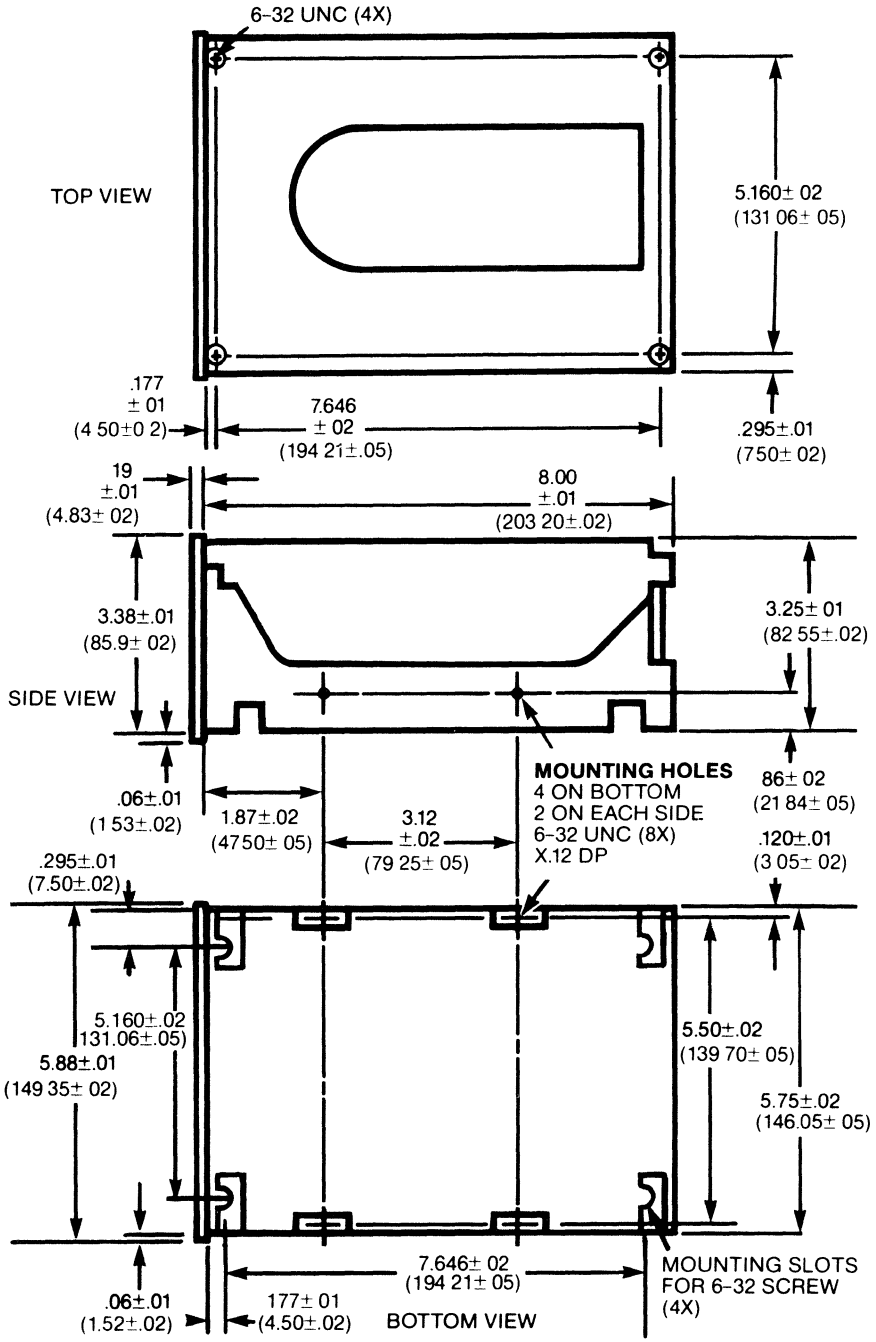


Figure 18 —— MOUNTING DIMENSIONS

SECTION 7 — SURFACE DEFECTS

As shipped, any area which is considered MARGINAL for DATA RECORDING or which has a PERMANENT DEFECT, will be indicated in a listing of the defects in terms of CYLINDER, HEAD and the NUMBER OF BYTES from INDEX. This listing will be supplied with each unit shipped. These areas should not be used for data handling even though they may appear to be good.

INDEX

Reader's Comments

Please comment on this manual. In order to improve our publications, we will carefully evaluate your opinions.

As a learning tool, this book is	poor	good
For locating specific information, this book is	poor	good
The information in this book is	accurate	inaccurate
There is too little/too much information	little	much
There should be fewer/more examples	fewer	more
This book should be	shorter	longer
The writing is	unclear	clear

OTHER COMMENTS:

Name _____

Job title _____

Company _____

Address _____

Seal or tape for mailing—do not use staples
fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1309 SANTA CLARA, CA.

POSTAGE WILL BE PAID BY—

Convergent Technologies™
Data Systems Division
3055 Patrick Henry Drive
Santa Clara, Ca. 95050

ATTN: TECHNICAL PUBLICATIONS

fold

Convergent Technologies™
Data Systems Division
3055 Patrick Henry Drive, Santa Clara, CA 95050 • (408) 980-0850

Printed in U.S.A.