

at&t

3B1

HARDWARE REFERENCE

© 1985 AT&T

©1986 AT&T
All Rights Reserved
Printed in USA

NOTICE

The information in this document is subject to change without notice. AT&T assumes no responsibility for any errors that may appear in this document.

Motorola is a registered trademark of Motorola, Incorporated.
Centronics is a registered trademark of Centronics Data Corporation.
Cipher is a registered trademark of Cipher Data Products, Incorporated.
Seagate is a registered trademark of Seagate Technology.
Toshiba is a registered trademark of Toshiba America Inc.

L-244069-5

The AT&T **UNIX**® PC Reference Manual is solely intended for use by qualified service personnel to assist in maintaining the **UNIX** PC. The Reference Manual is NOT intended for development of **UNIX** PC hardware enhancements. A "**UNIX** PC Expansion Bus Specification" can be obtained for this purpose by writing:

AT&T Information Systems
UNIX PC Product Management
1776 on the Green
Morristown, New Jersey 07960
(201) 898-6000

The **UNIX** PC Reference Manual is subject to change without notice. The recipient accepts this Reference Manual "as is." Neither AT&T Information Systems, nor its affiliates, contractors, suppliers, or agents make any representations or warranties, expressly or implicitly, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose, with respect to this Reference Manual and shall not be liable for errors contained in such Manual or from any claim for damages or expenses, including indirect, special, reliance, incidental or consequential losses (including lost profits or opportunities), arising out of the use thereof, whether or not AT&T-IS, its affiliates, contractors, suppliers or agents had prior knowledge of any defect or error in the Reference Manual.

Preface

The AT&T UNIX® PC Reference Manual has been written for technicians doing component-level troubleshooting of the AT&T UNIX® PC logic board.

Organization of this Manual

This manual contains the following sections:

System Features and Functions

Briefly describes the physical features and functional capabilities of the UNIX PC system.

Logic Board Theory of Operation

Describes the logic board hardware and the functions performed by it, including direct memory access and bus arbitration, machine cycle timing, memory management, and input/output handling.

Diagnostics

Describes boot ROM, floppy disk, and expert mode diagnostics, including algorithms, screen displays, and error messages.

Logic Board Test Procedures

Contains a collection of test procedures intended to aid in troubleshooting.

Schematics

Contains the schematic of the logic board for P4 and P5 configurations and schematics for DMA and video gate arrays.

Appendix A: PAL Equations

Contains the logic equations for the arbitor, disk interface, memory management unit, and hard disk data separator PALs.

Appendix B: Mnemonics

Contains definitions of the mnemonics used throughout the theory of operation and the schematics.

Appendix C: Expansion Memory Locations

Contains a table listing of the possible expansion memory configurations and their expansion slot position requirements.

Contents

1	**System Features and Functions**	**1-1**
	Functional Specifications	1-3
	Logic Board	1-3
	Terminal Subsystem	1-4
	Keyboard	1-4
	Mouse	1-5
	Audible Indicator	1-6
	Data Storage	1-6
	RS-232-C Port	1-6
	Centronics Parallel Printer Interface	1-9
	Expansion Slots	1-12
	Physical and Electrical Specifications	1-13
	Logic Board Bus System	1-13
	Simplified Address and Data Block Diagram	1-15
	Onboard Memory	1-17
	System Control Block Diagram	1-17
2	**Logic Board Theory of Operation**	**2-1**
	Direct Memory Access	2-1
	Machine Cycle Timing	2-1
	Memory Management	2-2
	Input/Output Handling	2-2
	Boot ROM Algorithm	2-2
	Fast and Slow Cycles	2-6
	Clock Generation	2-7
	Bus Arbitration	2-9
	Bus Masters and Slaves	2-9
	Arbitration Priority	2-10
	Arbitration Signal Sequence	2-11
	Reset Vector Loading	2-12
	68010 ROM Read Sequence	2-13
	Mapping Virtual Address	2-15

Write Sequence: Processor to Page Map	2-21
608010 Local RAM Read Sequence	2-23
Parity	2-27
Page Status Update and Memory Management Errors	2-29
Generation of Memory Management Interrupts	2-30
Refresh Operation	2-31
System Control and Status Registers	2-33
Reading General Status Register (Address 410000)	2-39
Interrupting the Processor	2-40
Exception Vectors	2-40
Interrupts: Theory of Operation	2-42
Disk Direct Memory Access (DMA)	2-43
Definitions	2-45
Initializing Hard Disk DMA Read	2-57
Executing Hard Disk DMA Read	2-62
Terminal Count	2-64
Underrun/Overrun Errors	2-64
Separating Hard Disk Data	2-65
Floppy Disk Direct Memory Access	2-69
Video Bit Map	2-78
Screen Layout	2-78
State Generator	2-80
68010 Loading of Bit Map	2-81
Bit Map Address Multiplex	2-83
Refresh Address Counter	2-84
Horizontal Synchronization	2-85
Vertical Synchronization	2-85
Shift Register	2-85
Telephony	2-86
Line Control	2-87
Telephony Status Register	2-89
Dialer Interface	2-90
Serial Communication	2-91
RS-232 Serial Port	2-91
Baud Rate Generation	2-93
7201 Serial Controller	2-94
Modem	2-96
Line Printer	2-100
Ports	2-100
Control Signals	2-100
Write Sequence	2-101
Status Signal Description	2-102
Serial Printer Cables	2-103
Realtime Clock Interface	2-104
Keyboard Controller	2-105

3	**Diagnostics**	**3-1**
	Boot ROM Program	3-1
	Bootstrap Jump	3-1
	Initializing the System	3-2
	Initializing the 7201 Serial Port Controller	3-2
	Initializing the Keyboard	3-2
	Initializing the Modem	3-2
	Initializing the Telephone Line Control	3-2
	Clearing the Printer Interrupt	3-3
	Clearing the Dialer Chip	3-3
	Resetting the Disk DMA	3-3
	Testing Video RAM	3-3
	Testing Map RAM Memory	3-4
	Testing RAM	3-4
	Jumping to the Loader Program	3-5
	Floppy Disk Diagnostics	3-6
	Main Menu Diagnostics Summary	3-6
	Diagnostic Subtest Summary	3-7
	Diagnostic Test Descriptions	3-9
	Expert Mode Diagnostics Program	3-24
	Disk Drive Commands	3-26
	Read and Write Commands	3-26
	Seek and Step Commands	3-27
	Register Commands	3-27
	Buffer Commands	3-28
	Miscellaneous Commands	3-29
	Reading/Writing to Hard Disk with Expert Mode	3-29
4	**Logic Board Test Procedures**	**4-1**
	Map RAM EPROM Program	4-1
	Data Test	4-2
	Address Test	4-3
	Error Loops	4-4
	RAM EPROM Program	4-5
	Debugger Program	4-6
	Machine Language Programming	4-9
	Definitions	4-9
	Walking Ones Test Program	4-12
	Read/Write Loop Program	4-12
	Turn On/Turn Off Program	4-14
	Parity Test Procedure	4-15
	Page Status Test	4-17
	MMUERR* Test	4-18
	Dummy DMA Test Procedure	4-19

	Floppy Disk Recalibration Procedure	4-20
	Floppy Disk Read Track Procedure	4-23
	Seek Command	4-25
	Read Address Command	4-25
	Floppy Write Sector Procedure	4-27
	Initialize Floppy Program	4-27
	DMA Looping Program	4-29
	Hard Disk Registers	4-29
	WD1010 Command Register (E0000E)	4-31
	Commands in Hexadecimal	4-31
	Write Precompensation Register (E00002, Write Only)	4-33
	Error Register (E00002, Read Only)	4-34
	Status Register (E0000E, Read Only)	4-34
	Hard Disk Recalibration Procedure	4-35
	Procedures for Reading/Writing Data to Any Sector	4-36
	Read Data to Any Sector	4-36
	Write Data to Any Sector	4-38
	Signals to Check if Procedures do not Execute	4-39
	Hard Disk Data Separator Test	4-40
	Dialer Test	4-42
	Waveform Timing Analysis Procedure	4-42
	Logic Analyzer Setup	4-43
	Typical Waveforms	4-44
	68010 Timing Summary	4-50

A	**PAL Equations**	**A-1**
B	**Mnemonics**	**B-1**
C	**Expansion Memory Locations**	**C-1**

Figures

1-1	Base Unit, Keyboard and Mouse	1-2
1-2	Keyboard Layout	1-5
1-3	Keyboard and Mouse Connections	1-6
1-4	RS-232-C Cabling	1-8
1-5	Loopback Plug Pinning	1-9

Figures (Continued)

1-6	Logic Board	1-14
1-7	Bus System Block Diagram	1-16
1-8	System Control Block Diagram	1-18
2-1	System Bus	2-10
2-2	Parity Generator	2-28
2-3	Floppy Disk Write Timing Diagram	2-76
2-4	Screen Layout	2-79
2-5	State Signal Generator Timing	2-81
2-6	**UNIX** PC to Terminal Cable Pinning	2-92
2-7	Loopback Plug Pinning	2-92
2-8	Write Sequence Timing	2-102
2-9	Printer Cable Pinning	2-103
3-1	LED Locations and Off/On Patterns	3-5
4-1	Timing of AS- (130 ns)	4-44
4-2	Timing of DTACK (210 ns)	4-45
4-3	Timing of LDS and UDS (230 ns)	4-45
4-4	DTACK Latched by Processor (260 ns)	4-46
4-5	AS, LDS, and UDS Deasserted by Trailing Edge of Next 1PCK (380 ns)	4-46
4-6	Length of Machine Cycle (400 ns)	4-47
4-7	Timing of LDS and UDS During a Read Operation (130 ns)	4-47
4-8	Slow Cycle Timing of AS (130 ns)	4-48
4-9	Timing of DTACK (710 ns)	4-48
4-10	DTACK Latched by Processor (760 ns)	4-49
4-11	AS, LDS, and UDS Deasserted by Trailing Edge of Next 1PCK (880 ns)	4-49
4-12	Slow Cycle Ends (Complete Cycle = 1.0 us)	4-50
4-13	68010 to DRAM Write Cycle	4-53
4-14	68010 Read Cycle Summary	4-54

Tables

1-1	RS-232-C Signals	1-7
1-2	The Parallel Interface	1-10
1-3	Interrupt Priorities	1-19
1-4	Bus Arbitration Priorities	1-20
1-5	Hard Disk Drive	1-20
1-6	Floppy Disk Drive (Winchester)	1-24
2-1	68010 Processor Pin Functions	2-3
2-2	Significant Address Decoding	2-6
2-3	Bus Master Priorities	2-11
2-4	Summary of Address Bit Assignments	2-20
2-5	Row Address Strobing	2-25
2-6	General Status Register--Address 410000 (Read Only)	2-33
2-7	Bus Status Register 0--Address 430000 (Read Only)	2-34
2-8	Bus Status Register 1--Address 440000 (Read Only)	2-35
2-9	General Control Register--Address E4X000 (Write Only)	2-36
2-10	Miscellaneous Control Register--Address 4A0000 (Write Only)	2-37
2-11	Interrupt Vector Numbers and Hexadecimal Offsets	2-41
2-12	Interrupts and Their Priority Levels	2-42
2-13	Disk Control Register--Address 4E0000 (Write Only)	2-46
2-14	Miscellaneous Control Register--Address 4A0000 (Write Only)	2-47
2-15	Line Printer Status Register--Address 4A0000 (Write Only)	2-48
2-16	Pin Functions WD1010	2-49
2-17	Task File Registers	2-52
2-18	WD1010 Commands	2-54
2-19	DMA Address Count Register Selection	2-55
2-20	Disk DMA Count Register--Address 460000	2-56

Tables (Continued)

2-21 Signal Status to Assert MREG WR* 2-57
2-22 Signal Status to Assert HDCTLWR* 2-58
2-23 Signal Status to Assert DCNTCS* 2-60
2-24 Pin Functions WD2797 2-70
2-25 WD2797 Registers 2-72
2-26 WD2797 Commands 2-73
2-27 Signal Status to Generate BMSEL 2-82
2-28 Multiple Select Status 2-83
2-29 Bit Map Address Multiplex Assignments 2-84
2-30 Address Status to Select Telephony Control Register 2-88
2-31 Telephony Control Register--Address 49X000 2-89
2-32 Telephony Status Register--Address 450000 (Read Only) 2-90
2-33 RS-232-C Signals 2-91
2-34 7201 Registers 2-95
2-35 Modem Registers 2-98
2-36 Address Status to Select Modem and Dialer Network Control Signals 2-99
2-37 I/O Ports 2-100
2-38 Line Printer Status Register--Address 470000 (Read Only) 2-102
2-39 Realtime Clock Interface--Address 480000 2-104
2-40 Keyboard Controller Addresses 2-105

3-1 Read/Write Command Format 3-26
3-2 Seek and Step Command Format 3-27
3-3 Register Command Format 3-28
3-4 Buffer Command Format 3-28
3-5 Miscellaneous Command Format 3-29

4-1 Logic Board LED States 4-2
4-2 Program Instructions 4-10
4-3 Read/Write Loop Program 4-13
4-4 WD2797 Pin Listing 4-22
4-5 Bit Number 4-31

Tables (Continued)

4-6	Read Sector Commands	4-32
4-7	Write Sector Commands	4-32
4-8	SDH Byte Register (E0000C, Read/Write)	4-33

1 System Features and Functions

The **UNIX** PC is an intelligent desktop workstation that provides users with personal computing and enhanced voice and data communications services. It provides the UNIX System V virtual memory operating system in a telephone network environment. The **UNIX** PC can connect to a telephone system to allow communication with other telephones, workstations, and computers. Direct connection, or connection through a local area network, to other terminals, workstations, or computers, is also provided. The **UNIX** PC can be upgraded to a multiuser system.

The **UNIX** PC consists of the following parts, as illustrated in Figure 1-1:

- Base unit
- Keyboard
- Mouse

The workstation base unit houses the monitor, power supply, hard disk drive, floppy disk drive, logic board, and three expansion slots. The logic board provides the processor logic, bit-mapped graphics logic, communications, and interface logic for all connected input/output (I/O) devices. The monitor is attached to a base that allows it to tilt and swivel.

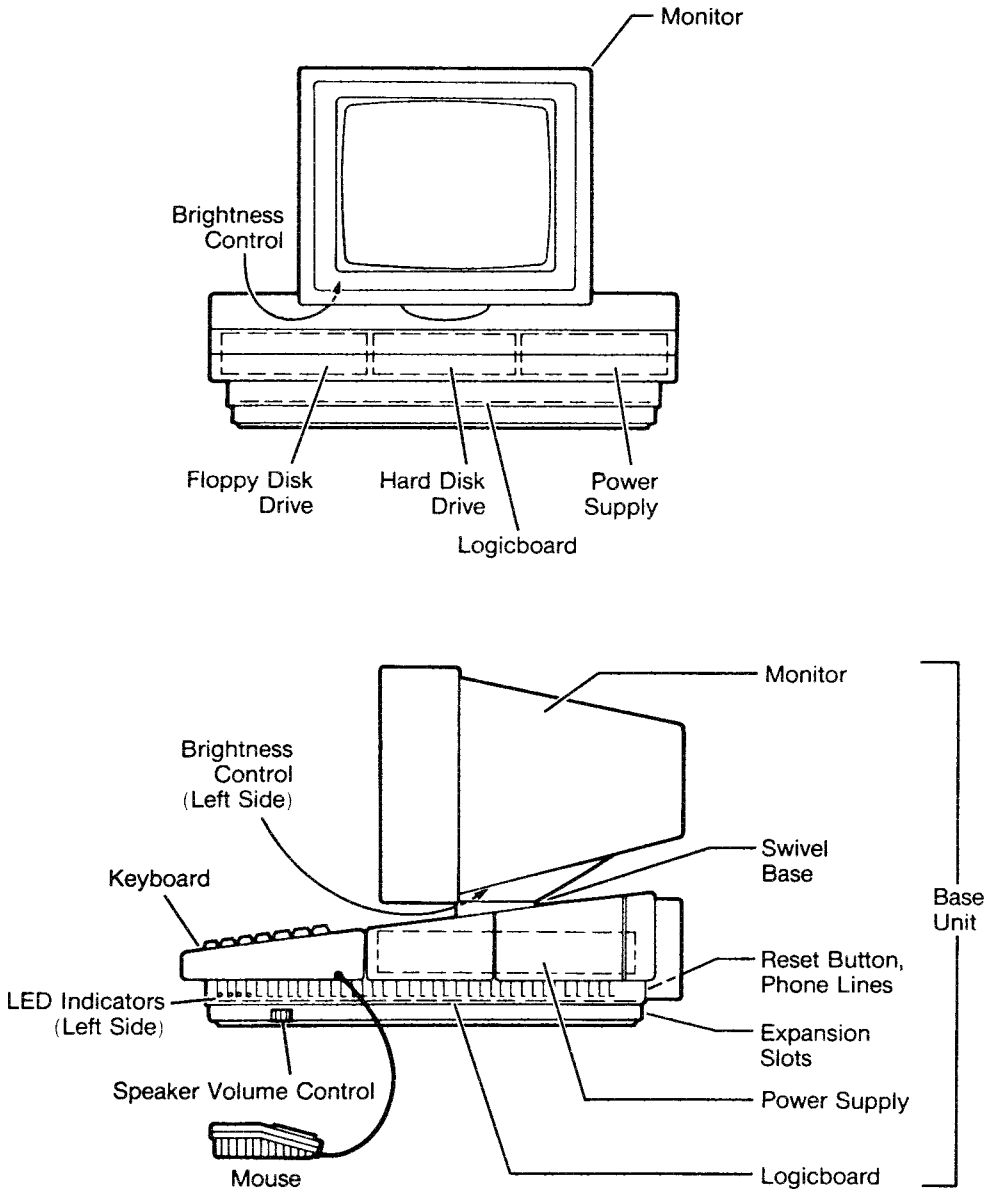


Figure 1-1 Base Unit, Keyboard, and Mouse

Functional Specifications

These specifications describe the major circuitry and general characteristics of the UNIX PC system.

Logic Board

The Logic Board is shown in Figure 1-4.

- Motorola 68010 central processing unit (CPU) with 10-megahertz (MHz) clock
- Virtual memory address space of 4 megabytes (MB)
- 0.5MB, 1 MB standard or 2 MB random access memory (RAM)
- 720 by 348 bit-mapped graphics monitor interface
- DTE RS-232-C serial port
- Centronics-compatible parallel printer port
- Keyboard interface
- Telephone interfaces for voice and data service. Three modular jacks are used: one for connection to a user-provided telephone and the other two for connection to tip/ring telephone lines. Also included is an integrated 300/1200 bits per second (bps) modem compatible with AT&T Models 103 and 212, offering asynchronous operation and autobaud capabilities.
- Hard disk interface
- Floppy disk interface
- Expansion bus interface that allows memory and I/O expansion. The bus has 21 address lines and 16 data lines and supports bus mastership by expansion hardware
- A realtime clock that retains the time and date when the UNIX PC is powered down

Terminal Subsystem

The monitor contains a 12-inch cathode ray tube (CRT), a deflection board, and a yoke. It provides a 20-MHz screen capable of displaying 720 by 348 pixels. The display can be programmed either as light on dark (normal) or dark on light (inverse video).

The monitor is attached to the base. The monitor tilts -5 to +20 degrees relative to the horizontal plane and swivels.

The screen is treated to reduce glare. A brightness control is accessible to the operator, as shown in Figure 1-1.

Keyboard

The keyboard is shown in Figure 1-2.

The keyboard is connected to the base with a flexible, coiled cord that can expand to approximately six feet. One end of the cord has about an inch of straight cord that plugs into the base unit. Both ends of the cable have connectors that prevent accidental disconnection.

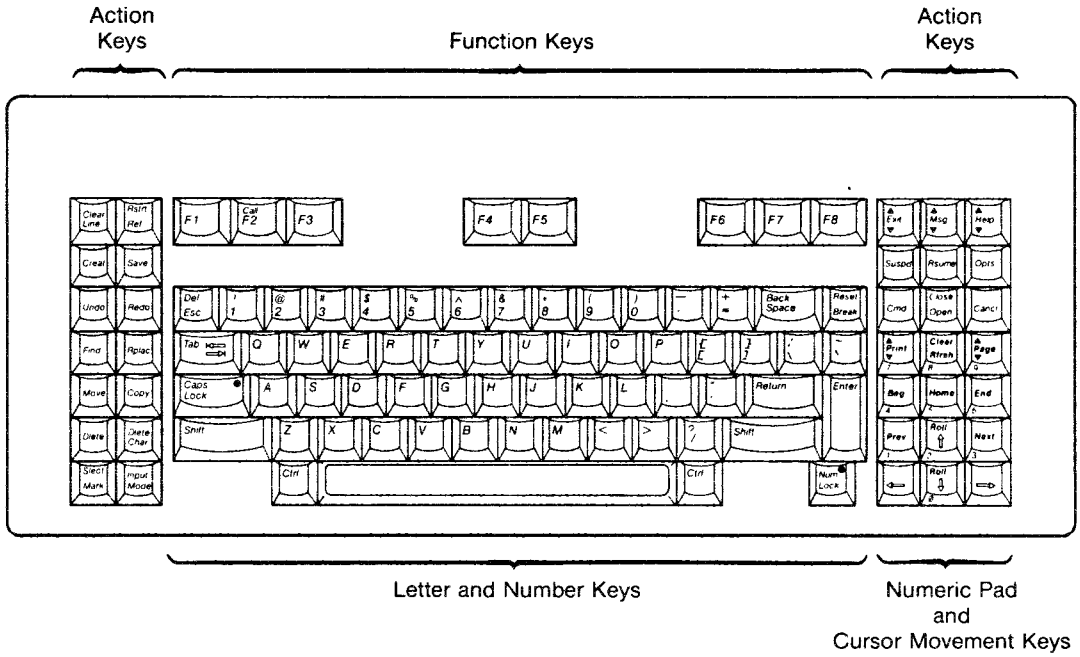


Figure 1-2 Keyboard Layout

Mouse

The mouse connects to the keyboard unit with a lightweight, uncoiled cord that is approximately four feet long. The cable has a connector that locks preventing accidental disconnection. The cord plugs into the keyboard. These connections are shown in Figure 1-3.

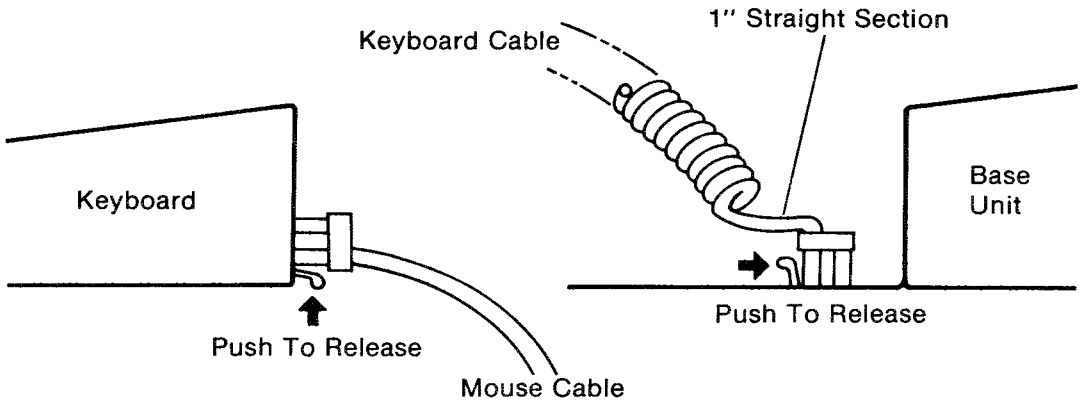


Figure 1-3 Keyboard and Mouse Connections

Audible Indicator

An audible indicator consisting of a small speaker is provided for monitoring telephone calls when using the AT&T **UNIX** PC Telephone Manager. A user-accessible slide volume control, illustrated in Figure 1-1, is located just under the right edge of the base unit.

Data Storage

Data can be stored either on a hard disk which is part of the **UNIX** PC system, or on floppy disks using the floppy disk drive. The storage capacities available are:

- A 10MB, 20MB, 40MB or 67MB hard disk (Winchester) for mass storage

- A double-sided, 1/2-MB (320 Kb formatted), 5 1/4-inch, 48-tpi floppy disk drive

RS-232-C Port

The RS-232-C port supports both synchronous and asynchronous data communications. Asynchronous bit rates of 110 bps to 19.2 Kbps are available.

RS-232-C Signals

The following table of signals applies to the RS-232-C connector. The table gives the pin number, signal name, and direction for the **UNIX** PC.

Table 1-1 RS-232-C Signals

Pin	Name	Direction
1	Ground (shield)	-
2	Transmit data	Output
3	Receive data	Input
4	Request to send	Output
5	Clear to send	Input
6	Data set ready	Input
7	Ground (signal)	-
8	Carrier detect	Input
15	Transmit clock	Input
17	Receive clock	Input
20	Data terminal ready	Output
22	Ring indicator	Input
24	DTE transmit clock	Output

Channel B of the 8274/7201 multiple protocol serial controller is connected to the modem. The following list describes the channel B signals:

8274/7201 Carrier detect <---RS-232-C ring indicator

8274/7201 Receive clock <----Modem receive clock

8274/7201 Clear to send <----RS-232-C data set ready

8274/7201 Transmit data----> Modem transmit data

8274/7201 Transmit clock <---Modem transmit clock

8274/7201 Receive data <-----Modem receive data

RS-232-C Signal Levels

Figure 1-4 illustrates the possible RS-232-C cabling to a printer or terminal.

Signal levels are +/-12V nominal.

UNIX PC to Terminal Cable Pinning

UNIX PC		Terminal
1	----------	1
2	---------->	3
3	<----------	2
4-5-6		4-5-6
7	----------	7
8	---------->	20
20	<----------	8

UNIX PC to Printer with CTS Control

UNIX PC		Printer
1	----------	1
2	---------->	3
3	<----------	2
4	<---------->	4
6-8-20		
5	----------	7

Figure 1-4 RS-232-C Cabling

Diagnostic Loopback Plug

The diagnostic floppy tests RS-232-C functions through the use of a loopback plug, which must be installed when a channel is being tested.

Loopback plug (male) pinning is shown in Figure 1-5.

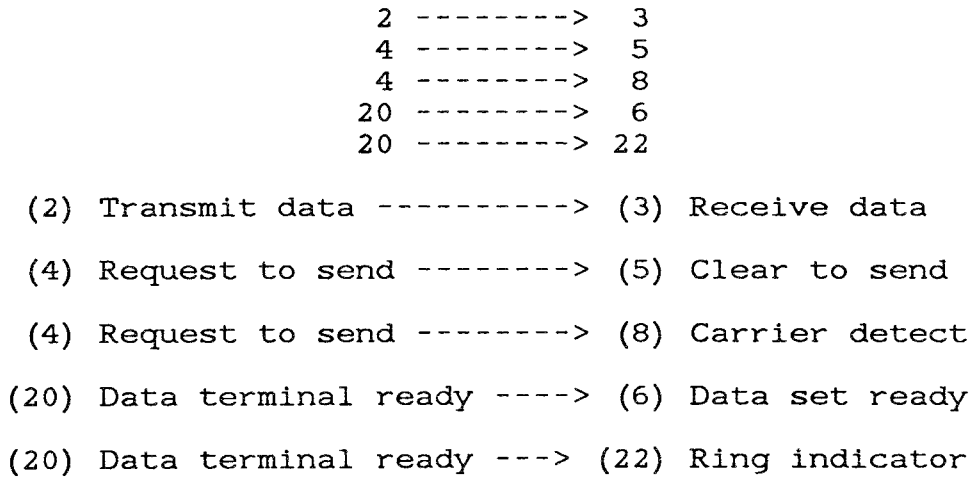


Figure 1-5 Loopback Plug Pinning

Centronics Parallel Printer Interface

Table 1-2 is an example of how a parallel printer cable might be constructed for a Centronics printer. The **UNIX** PC has an Amphenol 57 series 36-pin connector. This is a standard Centronics connector.

Cable Pinning

The following guidelines showing the printer signal requirements do not have to be adhered to strictly when building your own cable (in the case of signal ground). Signal ground is tied to pins 16, 17, 19-30, 33, and 36 on the **UNIX** PC connector.

Make sure your printer is strapped for negative strobes and acknowledges. Do not let any signals float. For example, if you are not going to use BUSY+, ground it.

This table shows typical pin functions for the Centronics printer cable:

Table 1-2 The Parallel Interface

Signal Pin	Return Pin	Signal	Direc-tion	Description
1	19	$\overline{\text{STROBE}}$	OUT	Pulse to read data in. Pulse width should be 0.5 ms at the receiving terminal.
2	20	DATA 1	OUT	These signals represent information of the 1st to 8th bits of parallel data, respectively. Each signal is at HIGH level when data is logical 1 and LOW when it is logical 0.
3	21	DATA 2	OUT	
4	22	DATA 3	OUT	
5	23	DATA 4	OUT	
6	24	DATA 5	OUT	
7	25	DATA 6	OUT	
8	26	DATA 7	OUT	
9	27	DATA 8	OUT	
10	28	$\overline{\text{ACKNLG}}$	IN	Approximately 12-microsecond pulse. LOW indicates that data has been received and that the printer is ready to accept more data.
11	29	BUSY	IN	A HIGH signal indicates that the printer cannot receive data. The signal goes HIGH in the following cases: o During data entry o During printing o When offline o During printer -error state
12	30	PE	IN	A HIGH signal indicates that the printer is out of paper.

Table 1-2 The Parallel Interface (Continued)

Signal Pin	Return Pin	Signal	Direction	Description
13	--	LP SELECT	IN	Pulled up to +5 volts through a 1K-ohm resistor.
14	--	$\overline{\text{AUTO FEED XT}}$	OUT	This signal is pulled up to +5 volts through a 1K-ohm resistor.
15	--	NC	--	Unused.
16	--	OV	--	Logic ground level.
17	--	CHASSIS GND	--	Printer's chassis ground, which is isolated from the logic ground.
18	--	NC	--	Unused.
19-30	--	GND	--	Twisted-pair return signal ground level.
31	--	$\overline{\text{INIT}}$	OUT	This signal is pulled up to +5 volts through a 1K-ohm resistor.
32	--	$\overline{\text{ERROR}}$	IN	This level becomes LOW when the printer is in: o Paper-end state o Offline o Error state.
33	--	GND	--	Same as for pins 19-30.
34	--	NC	--	Unused.
35	--	NC	--	Unused.
36	--	$\overline{\text{SLCT IN}}$	OUT	Signal ground level.

Notes

- The column heading "Return" denotes the twisted-pair return, to be connected at signal ground level. For the interface wiring, be sure to use a twisted-pair cable for each signal and to complete the connection on the return side. To prevent noise, these cables should be shielded and connected to the chassis of the host computer and the printer, respectively.
- The column heading "Direction" refers to the direction of signal flow as viewed from the base unit.
- All interface conditions are based on TTL level. The rise and fall times of each signal must be less than 0.2 microseconds.

Status Signal Description

- LPNOPAPER+: Centronics pin 12, asserted by printer when paper-out sensor senses no paper in the printer.
- LPBUSY+: Pin 11, asserted by the printer to indicate that it cannot receive data. Also indicates a paper empty or fault condition.
- LPSELECT+: Pin 13, asserted by printer to indicate that it is selected and ready to receive data.
- ERROR*: Pin 32 asserted when there is a problem with the printer.
- LPACK*: Pin 10, asserted by line printer to indicate that it has received data.

Expansion Slots

Three expansion slots are provided as part of the base unit. Expansion cards can be installed in any slot. However, depending on the memory being added, they must be located in accordance with the expansion memory location matrix in Appendix C.

Expansion slots support expansion boards including those listed below:

- 0.5MB or 2MB expansion RAM board
- Three versions of combo boards

  0.5MB, 1MB or 1.5MB with two RS-232 ports
- MS-DOS expansion board
- Two port RS-232 only board
- Interface board for tape backup (floppy tape)
- Interface board for tape backup (QIC-02)

Expansion boards may dissipate up to 12 watts each.

Physical and Electrical Specifications

The basic characteristics of the **UNIX** PC are:

- Base unit: Approximately 18 inches wide, 17 inches deep, and 16 inches high; weighs approximately 40 pounds.
- Keyboard: AT&T 103-key, low-profile design.
- Electrical: 100-130 volts; maximum power under 400 watts.

Logic Board Bus System

This section describes the logic board bus system, including the address and data bus and the system control block diagram, which explains how bus transfers are regulated. Figure 1-6 shows the layout of the **UNIX** PC logic board.

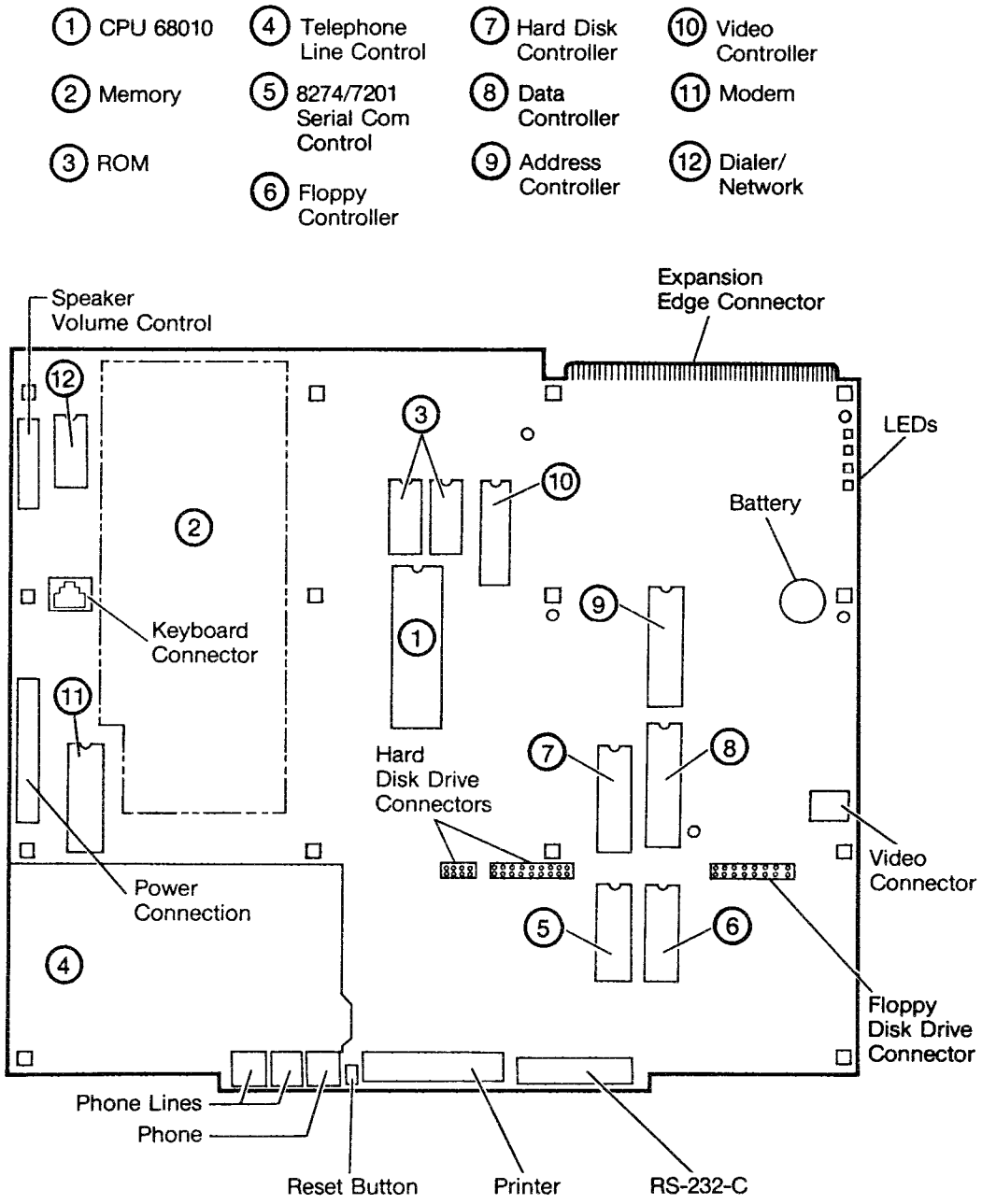


Figure 1-6 Logic board

Simplified Address and Data Block Diagram

The system block diagram in Figure 1-7 shows how the system bus allows the various devices within the **UNIX** PC CPU board to transmit data to each other.

The right half of the drawing shows the peripheral devices. Each peripheral has special control interface circuitry that modifies information coming from or going to the peripheral into a form that is acceptable for the bus and the peripheral. These circuits are indicated in the drawing by the rectangular boxes marked **bit map, printer port, telephony controller,** and so on. The bus accepts data 16 bits wide. The keyboard, for example, generates data in the form of a serial bit stream. These control interface circuits also receive control signals that initiate and terminate data transfers.

The left half of Figure 1-5 shows devices that do internal information processing. These include the 68010 CPU and the three forms of memory: ROM, RAM, and disk storage. The bus itself is really two buses, a data bus and an address bus.

The data bus consists of 16 bits, labeled D0-D15, for transmission of 16-bit data words. The address bus consists of 23 bits, labeled A1-A23. (There is an A0 function that is internal to 68010.)

Data transfers on the **UNIX** PC bus are performed using a master-slave system. A master device such as the 68010 begins a transfer by first putting an address on the address bus to identify the device with which it will perform a data transfer. Then, depending on the direction of the data transfer, either master-to-slave or slave-to-master data is loaded onto the data bus, and the transfer takes place.

In the **UNIX** PC, the 68010 and the DMA (direct memory access) controller are both masters. There are other possible masters that are not shown for simplicity. Any of the devices on the right side of the drawing can be slaves to the 68010. The DMA controller for the disk drives transfers data only to RAM memory, so it has only one slave.

Before starting a DMA transfer, the 68010 must load information into the DMA controller, in which case the DMA controller is acting as a slave to the 68010. During the transfer, the DMA controller generates appropriate control signals that cause the transfer to begin and end and also determine the direction of the transfer, either from master to slave or slave to master. On the drawing, arrows indicate the direction of transfer. Notice that the address bus differs from the data bus in that the address bus allows only a one-way transfer of information, from master to slave. The data bus allows two-way transmission, as indicated by the arrows.

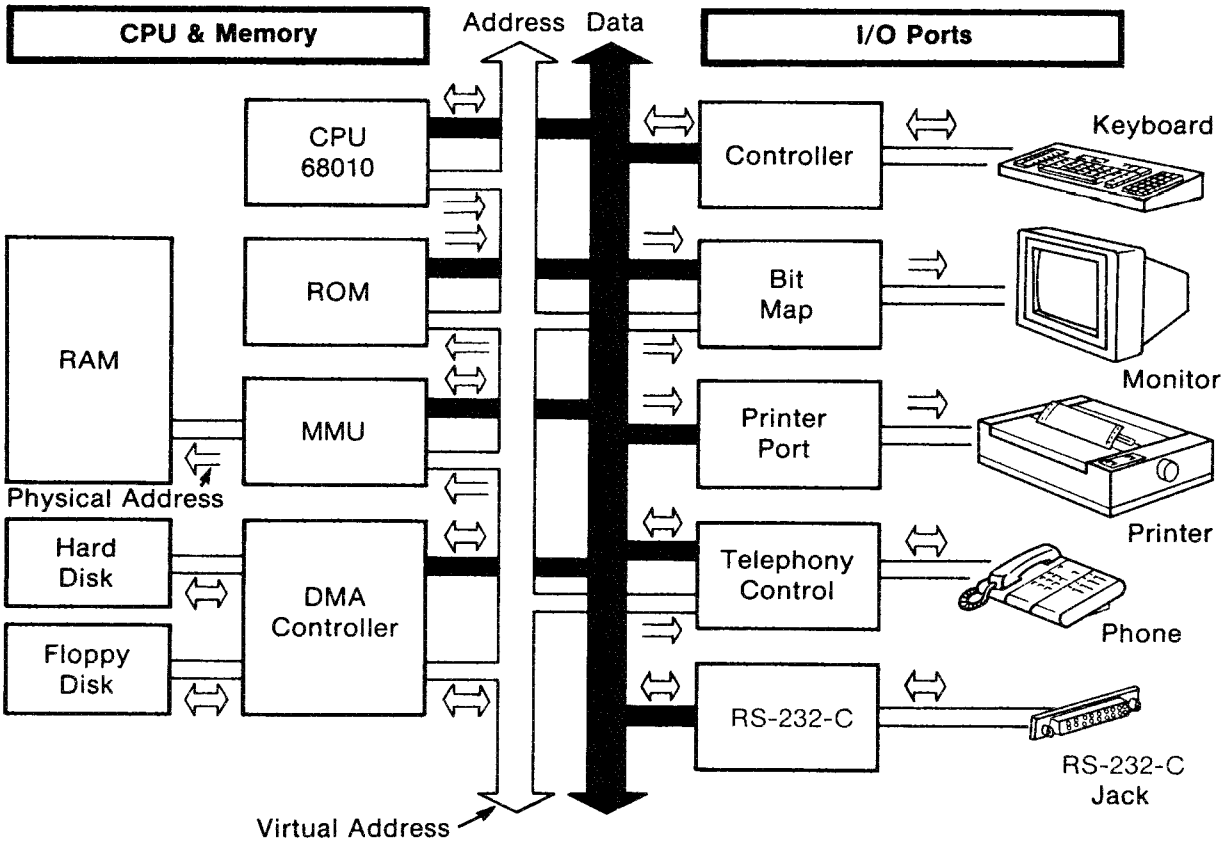


Figure 1-7 Bus System Block Diagram

Onboard Memory

Memory in the UNIX PC consists of both Random Access and Read Only memory, both are located on the logic board.

Random Access Memory

The logic board provides a minimum of 512 KB of onboard RAM, which can be expanded to either 1 or 2 MB maximum. A minimum memory configuration is made up of 72 type 4864, 64K by 1-bit dynamic RAM chips. The **1MB logic board** is made up of 36 type 4256, 256K by 1 bit, dynamic RAM chips. The **2MB logic board** is made up of 72 type 4256, 256K by 1 bit, dynamic RAM chips.

The memory is used for program execution. It is organized into a virtual memory system, which allows the programmer to write programs as if there were a much larger amount of memory available than is physically present. The **UNIX** PC virtual memory system is 4 MB. The hardware provides this function through a special set of memory chips called page map RAMs. These RAMs are 1K by 4-bit static RAM chips.

Read Only Memory

The logic board contains two 2764 8-KB or two 27128 16-KB ROM chips. They hold the initialization program that is run when the power is turned on or the Reset button is pressed, or a software reboot command is exercised.

System Control Block Diagram

The system control block diagram, Figure 1-8, shows how the system determines which bus master controls the bus at any given time. There are three elements to system control: interrupt, memory management, and bus arbitration.

The right side of the drawing shows the I/O controller logic. When a peripheral device such as the keyboard wants to send data to the system, its controller sends an interrupt signal to the interrupt logic. This is one method of communication between an I/O device and the system.

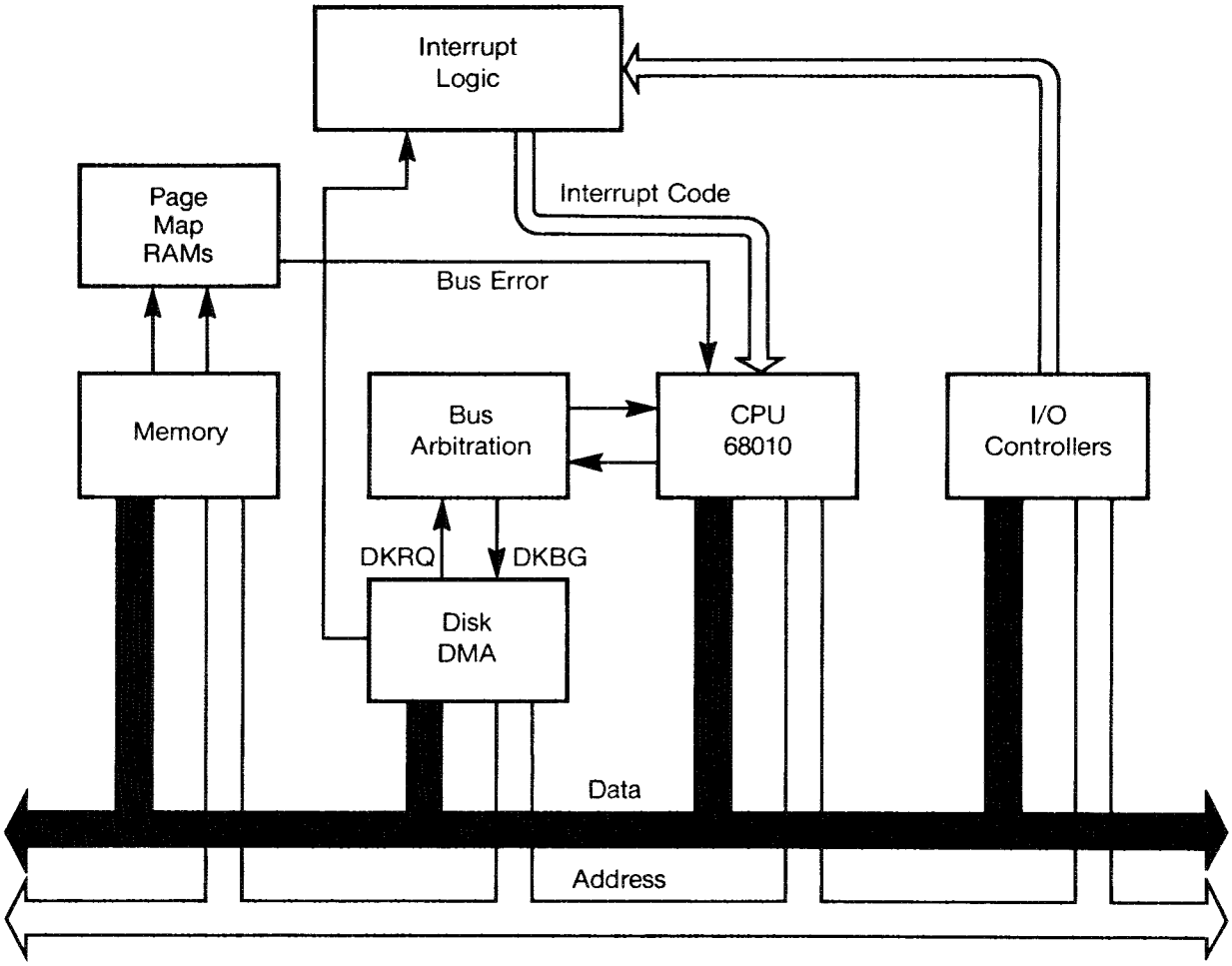


Figure 1-8 System Control Block Diagram

The interrupt signal is compared to a priority list. The highest priority pending at any given time causes the interrupt logic to send a signal to the 68010. The 68010 then responds by performing a sequence of data movements called an interrupt processing sequence. The function of the interrupt acknowledge is to allow the processor to store its current status so it can return to the same state after responding to the interrupt.

First it completes execution of its current instruction and stores the status of its internal registers. Then it jumps to an interrupt service program that determines which device generated the interrupt. Finally, it jumps to a program to service that particular interrupt. Interrupt priorities are listed in Table 1-3.

Table 1-3 Interrupt Priorities

Priority Level	Device
7 (highest)	Parity error or MMU error (logic board)
6	60-Hz (logic board)
5	Expansion slots 1, 2, and 3
4	8274/7201 communication (detection circuit, RS-232-C)
3	Keyboard/mouse, modem
2	Hard disk drive, floppy disk drive, or line printer
1 (lowest)	Expansion slots 1, 2, and 3

Note: Levels 1 and 5 are available to expansion slots 1, 2, and 3.

Memory Management

A second element of system control is the memory management unit shown on the left in Figure 1-6. This unit monitors every access to the dynamic RAM memory chips. Certain accesses cause a memory management error. For example, if a user program attempts to write to a memory address that has been defined as being in disk address space and not in physical RAM, the memory management unit generates a signal called a bus error.

Bus Arbitration

At some point two bus masters will both want control of the bus. The third element of system control, the bus arbitration unit, resolves the conflict. It evaluates requests for bus control from masters and grants bus control on a priority basis. The 68010 has lower priority than the disk controller. The 68010 has to wait for the disk controller to release control of the bus before it can take control.

Table 1-4 Bus Arbitration Priorities

Priority Level	Device
6 (highest)	Refresh
5	Expansion slot 1
4	Disk Interface Hard and Floppy
3	Expansion Slot 2
2	Expansion Slot 3
1 (lowest)	68010 CPU

Data Storage Device Specifications

The following tables list specifications for both types of storage device used on the UNIX PC system. These tables are arranged by manufacturer for each drive offered on the Model 7300 and 3B1 machines.

Table 1-5 Hard Disk Drive

Manufacturer	Specifications		
Miniscribe	Capacity Unformatted	Per Drive	12.0 MBytes 25.0 MBytes 53.0 MBytes 85.0 MBytes
		Per Track	10,416 Bytes
	Formatted	Per Drive	10.0 MBytes 20.0 MBytes 44.0 MBytes 67.0 MBytes
		Per Track	8,192 Bytes
		Per Sector	512 Bytes
		Sectors per Track	16

Table 1-5 Hard Disk Drive (Continued)

Manufacturer	Specifications	
Miniscribe (cont.)	Functional	Rotational Speed (RPM) 3600 Recording Density (bpi) 10,030 for 10, 20MByte Drives 9,950 for 40, 67MByte Drives Area Density (M/bit/sq/in) 5.9 Track Density (tpi) 588 for 10, 20MByte Drives 1000 for 40, 67MByte Drives Total Data Tracks 2,448 Cylinders 612 for 10, 20MByte Drives 1024 for 40, 67MByte Drives R/W Heads 10MByte 2, 20MByte 4, 40MByte 5, 67MByte 8
	Data Transfer Rate	(Mbits per second) 5.0
	Access Time (includes settling)	**10 and 20MByte** Average (msec) 85 Track-to-Track (msec) 15 Maximum (msec) 190 Latency (average, msec) 190 **40 and 67MByte** Average (msec) 30 Track-to-Track (msec) 3 Maximum (msec) 60 Latency (average, msec) 60
	Interface	ST412
	Error Rates Soft Read Errors 1 per 10^{10} bits transferred Hard Read Errors 1 per 10^{12} bits transferred Seek Errors per 10^{6} seeks	

Table 1-5 Hard Disk Drive (Continued)

Manufacturer	Specifications
Miniscribe (cont.)	DC Power Requirements **10MByte** +5V DC +/- 5%, 0.75 amps +12V DC +/-5%, 0.75 amps **20MByte** +5V DC +/- 5%, 0.4 amps +12V DC +/-5%, 1.0 amps **44MByte** +5V DC +/- 5%, 0.6 amps +12V DC +/-5%, 1.0 amps **67MByte** +5V DC +/- 5%, 0.6 amps +12V DC +/-5%, 2.0 amps
	Max Starting (10 sec) 3.5 amps
	Power Dissipation 14 watts

Table 1-5 Hard Disk Drive (Continued)

Manufacturer	Specifications	
Hitachi	Capacity Unformatted	Per Drive 51.0 MBytes Per Track 10,416 Bytes
	Formatted	Per Drive 40.0 MBytes Per Track 8,192 Bytes Per Sector 512 Bytes Sectors per Track 16
	Functional	Rotational Speed (RPM) 3600 Recording Density (bpi) 9,340 Area Density (M/bit/sq/in) 5.9 Track Density (tpi) 784 Total Data Tracks 2,448 Cylinders 714 R/W Heads 7 Disks 2
	Data Transfer Rate	(Mbits per second) 5.0
	Access Time (includes settling)	 Average (msec) 30 Track-to-Track (msec) 8 Maximum (msec) 55
	Interface	ST412
	Error Rates Soft Read Errors 1 per 10^{10} bits transferred Hard Read Errors 1 per 10^{12} bits transferred Seek Errors per 10^{6} seeks	
	DC Power Requirements +5V DC +/- 5%, 0.4 amps typical +12V DC +/-5%, 1.0 amps typical Max Starting (10 sec) 3.5 amps	
	Power Dissipation	14 watts

Table 1-6 Floppy Disk Drive (Winchester)

Manufacturer	Specifications	
Teac	Capacity Unformatted	Per Disk Single Density 250KBytes Per Disk Double Density 500KBytes Per Track 3,125KBytes Single Density, 6.25KBytes Double Density
	Formatted	Per Disk Single Density 163.84KBytes Per Disk Double Density 327.68KBytes Per Track Single Density 2,048KBytes Per Track Double Density 4,096KBytes Per Sector Single Density 256 Bytes Per Sector Double Density 512 Bytes Sectors per Track 8
	Functional	Rotational Speed (RPM) 300 Recording Density (bpi) Single Density 2,938 Double Density 5,876 Area Density (M/bit/sq/in) 5.9 Track Density (tpi) 48 Total Data Tracks 80

2 Logic Board Theory of Operation

This overview summarizes the major functions performed by the logic board hardware. In addition, it describes the boot ROM program algorithm.

The logic board hardware functions include:

- Direct memory access and bus arbitration
- Machine cycle timing
- Memory management
- Input/output handling

Direct Memory Access

The **UNIX** PC bus is shared by the 68010 central processing unit (CPU) and several direct memory access (DMA) devices, such as the disk bus interface unit, dynamic RAM refresh, and expansion boards. During a DMA transfer, the 68010 waits while data is moved directly from a DMA device, such as the disk bus interface unit, into RAM memory. DMA provides high-speed transfer of blocks of data to or from memory.

Machine Cycle Timing

A 68010 machine cycle consists of putting an address on the bus, transferring data, and releasing the bus. Machine cycles are either fast (400 nanoseconds) or slow (1100 ns). Access to the lower half of the address space results in a fast cycle; access to the upper address space results in a slow cycle. DMA machine cycles are considered fast (500 ns). The additional 100 ns is needed in this case to do bus arbitration.

Memory Management

Programs run on the **UNIX** PC are often too large to fit in the RAM memory chips on the logic board. Thus, when the system is booted up, only a portion of the program is loaded into memory. While the program is being executed, it is monitored by memory management hardware. When a portion of the program that is on the disk is needed, the memory management hardware generates an error, causing the DMA to move the required portion of the program from the disk drive into RAM memory.

The processor can address locations anywhere in the entire 16 megabytes (MB) of system space, but the DMA can access only the lower 1/4 of system address space that is used by RAM (physical memory space).

Input/Output Handling

Input and output (I/O) operations are memory mapped--that is, the 68010 does not have separate instructions for I/O operations. I/O ports are accessed by assigning addresses to them. I/O operations are handled either by interrupt or by polling. Polling is used in boot ROM programs where stack operations are forbidden. Table 2-1 lists the 68010 processor pin functions.

Boot ROM Algorithm

The boot ROM is used for program memory following power up, hard reset (reset switch), or a software generated re-boot. The boot ROM program tests memory. Then it initializes the logic board by initializing the status of the memory management hardware and various peripheral controller chips. Then it causes a program to be loaded from floppy or hard disk and jumps to that program.

Table 2-1 68010 Processor Pin Functions

Pin No.	Mnemonic	Description
29-52	A1-A23	23-bit address bus (outputs only)--A unidirectional, three-state bus capable of addressing 16 MB of data. Provides addressing for all CPU cycles except space cycles.
1-5 54-64	D0-D15	16-bit data bus--Bidirectional, three-state bus that is the general-purpose data path. Transfers either words or bytes.
6	AS	Address strobe--Signal indicating there is a valid address on address bus.
7-8	UDS, LDS	Upper and lower data strobes--Signals used with R/W to control data flow on the data bus. UDS enables the upper byte; LDS enables the lower byte. When both are active, words are transferred.
8	R/W	Read/Write Defines the data bus transfer as a read or write cycle.
10	$\overline{\text{DTACK}}$	Data transfer acknowledge--Input indicating that a data transfer has been completed. When received during a read cycle, data is latched one clock cycle later and the bus cycle is terminated. When received during a write cycle, the bus cycle is terminated.
13	$\overline{\text{BR}}$	Bus request--Not used
11	$\overline{\text{BG}}$	Bus grant--Not used
12	$\overline{\text{BGACK}}$	Bus grant acknowledge--Not used

Table 2-1 68010 Processor Pin Functions (Continued)

Pin No.	Mnemonic	Description
23-25	$\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$	Interrupt priority level 0-2--Inputs indicating the encoded priority level of the device requesting to interrupt the CPU. Level 7 has highest priority, and level 0 indicates no interrupts present. Level 7 cannot be masked. These inputs must remain stable until the processor acknowledges, which is accomplished by setting FC0-FC2 and A04-A23 high.
26-28	FC0,FC1,FC2	Function code 0-2--Outputs activated along with address strobe to indicate the state (user or supervisor) and the cycle type currently being executed. FC2 FC1 FC0 Cycle Type Low Low Low (Undefined) Low Low High User data Low High Low User program Low High High (Undefined) High Low Low (Undefined) High Low High Supervisor data High High Low Supervisor program High High High CPU space
15	CLK	Clock--10-MHz input, internally buffered for development of timing needed internally by the processor.
20	E	Enable--Not used
19	VMA	Valid memory address--Not used
21	$\overline{VPA}$	Valid peripheral address--Asserted during an interrupt cycle to inform the CPU that the current interrupt cycle is an autovector cycle. As implemented in the **UNIX** PC, all interrupts are autovectored.

Table 2-1 68010 Processor Pin Functions (Continued)

Pin No.	Mnemonic	Description
22	$\overline{\text{BERR}}$	Bus error--Input informing the CPU that there is a problem with the current cycle. Most commonly used when applied memory address is not in primary memory.
17	$\overline{\text{HALT}}$	Halt--Bidirectional. When used as input, the processor floats all outputs and stops at the completion of the current bus cycle. Other uses are described below.
18	$\overline{\text{RESET}}$	Reset--Bidirectional signal that resets the system upon power up or pressing the Reset button. The reset and halt inputs are tied together to ensure a total processor reset. Pressing the Reset button for 10 clock cycles causes a total system reset. Upon power up, reset and halt must be driven low for at least 100 ms. A software reset causes the reset signal to be driven for 124 clock cycles.

Note

An alphabetical listing of mnemonics used in this section appears in Appendix B.

Fast and Slow Cycles

68010 machine cycles are either fast (400 ns) or slow (1100 ns) depending on the address being accessed. The slow cycle is achieved by delaying the arrival of data transfer acknowledge (DTACK) to the 68010. A custom IC containing the timing circuit determines how much delay to provide through address decoding. DMA machine cycles are fast (500 ns) transfers between the DMA devices and RAM memory or refresh of RAM memory.

Table 2-2 shows how the two most significant address bits are decoded to select fast or slow cycles:

Table 2-2 Significant Address Decoding

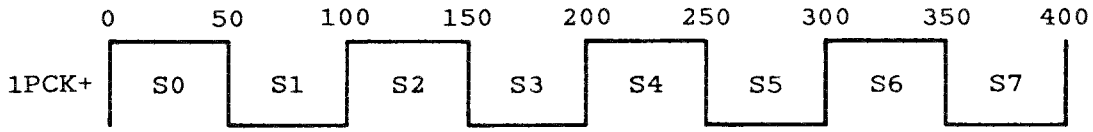
A23	A22	Region Description
0	0	RAM memory: fast-cycle access by 68010 user or supervisor mode
0	1	I/O registers: fast-cycle 68010 access in supervisor mode only
1	0	ROM memory: slow-cycle 68010 read in supervisor mode only
1	1	I/O registers: slow-cycle 68010 access in supervisor mode only

From the program execution viewpoint, fast and slow cycle transfers are identical. An instruction is executed in the same way except that the CPU waits longer for an acknowledge signal during slow-cycle transfers.

The CPU adjusts the length of its machine cycle to meet the requirements of the bus. It drives address and control signals and waits for DTACK data transfer acknowledge (pin 10).

circuits (IC) are referred to by their [illegible]

The 68010 is clocked by a 10-MHz signal called 1PCK. 1PCK is a 50% duty cycle clock with a full-cycle duration of 100 ns. The CPU uses four PCK cycles to accomplish a fast bus cycle. These four cycles are used as eight states (S0-S7) by the 68010:



The CPU samples DTACK* at the trailing edge of S4. If DTACK* is present at the trailing edge of S4, the CPU latches data (for a read), tristates control signals at the trailing edge of S6, and tristates its address lines at the trailing edge of S7.

If DTACK* is not present, the processor begins inserting wait states and sampling DTACK* at the trailing edge of each succeeding 1PCLK. When DTACK* arrives, the processor behaves as described in the preceding paragraph.

For a slow processor cycle, the hardware simply prevents the generation of DTACK* for an extra six clock cycles. The extra time is allotted for the slow response of the ROM and I/O devices. This describes the difference between a fast cycle and a slow cycle.

68010 CPU machine cycles are either instruction fetches or instruction execution cycles. The 68010 outputs status bits that identify the type of cycle being performed. These status bits can be used by a logic analyzer to display only program execution to aid in troubleshooting.

Clock Generation

Sheet 3 refers to schematic sheet 3. Note that integrated circuits (IC) are referred to by their location identifiers. For example, 21F refers to an IC located at position 21F on the logic board. As seen from the front of the **UNIX** PC, parts on the logic board are identified with two coordinates, numbers from left to right (1-28) and letters from front to back (A-P).

System clock signals are generated as follows:

- A 40-MHz oscillator provides the source frequency for the majority of the system clocks.
- This frequency passes through OR gate 21F and provides the clock for dual J-K F/F 20G.
- Pins 9 and 7 of 20G output a 20-MHz signal, and pins 5 and 6 output 10 MHz.
- These signals are buffered by 19G and output as 20MCK, X20MCK+, XPCK+, PCK*, 1PCK+ 2PCK+.
- 1PCK+ feeds the clock input of F/F 16K and causes the generation of the 5-MHz signal, 5MCK+.
- 1PCK+ feeds the DMA address IC (22E, sheet 9) and causes the generation of the 1-MHz signal, 1MCK+.

Bus Arbitration

A bus arbitration programmable array logic (PAL) decides which DMA device is granted accesses to the bus when two DMA devices request the bus at the same time. When no DMA devices are requesting the bus, the 68010 controls the bus.

Bus Masters and Slaves

The **UNIX** PC uses a system address and data bus as shown in Figure 2-1. The system bus makes it possible for bus arbitration to switch the bus between several devices, called bus masters.

The devices connected to the system bus are classified as either bus masters or bus slaves. In every data transfer, one device is the master and one is the slave. The master outputs the control signal that starts the transfer and provides the address of the device to or from which it wants to transfer data . A bus slave is connected to the bus when an address decoder detects the presence of the address assigned to that device on the address bus.

Figure 2-1 shows two bus masters in the **UNIX** PC system, the 68010 CPU and the disk DMA controller, and a single device to which they can both transfer data, the RAM memory array.

When the 68010 CPU transfers data to RAM memory, control signals are asserted that enable the tristate buffers and connect the 68010 address and data lines to the system address and data bus. At the same time, other control signals put the DMA circuits inside the custom DMA address and put data and gate array chips in a tristate condition, thus disconnecting them from the system address and data bus.

When the DMA ICs are transferring data between a disk drive and memory, the DMA address counter and data latch are enabled. The 68010 address and data buffers are in a tristate condition.

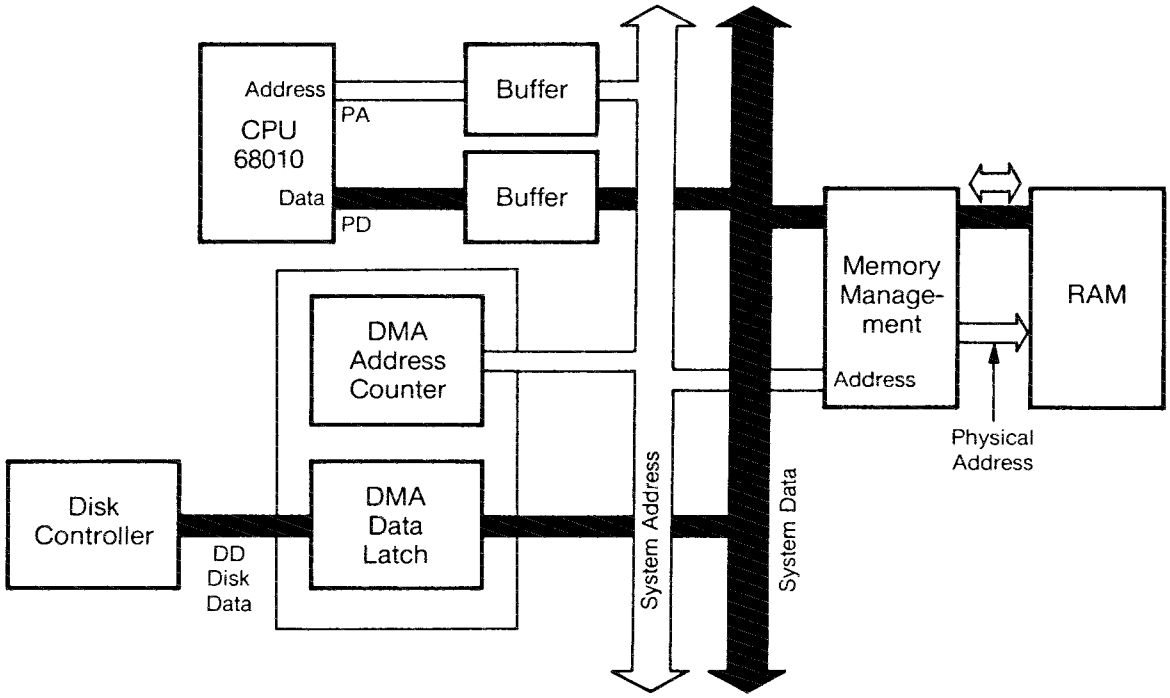


Figure 2-1 System Bus

Arbitration Priority

The process of deciding which bus master receives the bus at any given time is called bus arbitration. It takes place in the bus arbitration PAL 25B shown on sheet 2 of the schematics. For a detailed discussion of PALs, refer to Appendix A, PAL Equations.

The following table shows how priority is assigned to the various possible bus masters. With the exception of the processor, the other bus masters are DMA. The 68010 issues no bus request; it is granted use of the system bus by default (no other requests are present).

Table 2-3 Bus Master Priorities

Priority	Bus Master
1	Disk bus interface unit
2	Not used
3	Refresh controller
4	Expansion board 2
5	Expansion board 1
6	Expansion board 0
7	68010

Arbitration Signal Sequence

When a bus master requires use of the system bus, it issues a bus request to the 25B arbiter. If no higher priority requests are pending, the arbiter sends a bus grant to the requesting circuit.

The bus master then sends a bus grant acknowledge back to the arbiter, drops its request, and executes the bus cycle. The bus grant acknowledge latches the bus grant and locks out other requests.

When the bus cycle is finished, the bus master drops the bus grant acknowledge and the arbiter drops the grant.

The bus arbiter allows continuous transfers to occur. If another transfer is pending at the end of one transfer, the arbiter immediately sends a new bus grant and the next transfer begins.

Memory: Theory of Operation

Onboard memory consists of random access memory (RAM). The logic board contains a minimum of 512K of RAM that can be expanded to 2 MB. The logic board also contains read-only memory (ROM) that holds the initialization program. This program is run when power is turned on or the Reset button is pressed.

Memory Address

The 68010 provides 24 address bits, although address bit 0 does not leave the CPU. This allows the addressing of 16 MB of memory. The data bus contains 16 bits. Memory operations can be performed on either 8-bit bytes or 16-bit words.

All memory or register word accesses must be even-byte aligned. This means word instruction address operands must be even numbers. A0 is always zero. Address numbers are six hexadecimal digits from 000000 through FFFFFF. These correspond to 16 MB. Physical memory is organized in 16-bit words. The first word is address 000000, the second is 000002, the third is 000004, and so on. When even-numbered bytes are accessed, the upper half of the data bus D15 to D8 is used. When odd-numbered bytes are addressed, the lower half of the data bus, D7 to D0 is used. The internal address bit A0 is logically equivilent to the $\overline{\text{UDS}}$ (upper data strobe) pin of the 68010.

Words are accessed only at even addresses when both $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ are active.

Reset Vector Loading

When a system is powered or reset, the CPU automatically addresses ROM. A bootstrap routine directs the CPU to initialize the **UNIX** PC system and read the operating system from the disk.

When a system is reset, internal microcode forces the 68010 to read from RAM address 000000. Address 000000 is referred to as the reset vector. The 68010 loads the stack pointer with the values stored in addresses 000000 and 000002. It reads the values stored in addresses 000004 and 000006 and loads them into the program counter.

Actual address space for ROM is 800000-BFFFFF. The values necessary to initialize the stack pointer and the program counter are actually at addresses 800000 through 800006. Therefore, during a reset, the hardware is responsible for forcing address bit 23 high, thus causing the processor to refer to ROM.

During reset, a signal called ROMLMAP* is generated and ORed with address bit PA23 (see Sheet 5). Although the processor is addressing 000000, ROMLMAP* causes the address sent to ROM to be 800000.

At this point, the hardware forces the processor into the ROM address space, and no RAM references can be made as long as ROMLMAP* is active. The boot code is responsible for vectoring the processor into legitimate ROM address space and must also cause the processor to deactivate ROMLMAP* so RAM can be accessed again.

The processor is vectored into legitimate ROM address space by the first instruction executed after the program counter is loaded with the vector address at 800004 and 800006. This value is loaded into the program counter. The processor then fetches and begins instruction execution.

Now that the CPU is driving PA23 high, ROMLMAP* is no longer necessary. In fact, it must be dropped to access RAM. The first instruction, addressed by the reset vector, causes 8000 to be written to address E43000. This deactivates ROMLMAP*.

68010 ROM Read Sequence

The system must do a slow-cycle transfer to read ROM. The following description refers to the schematics.

Sheet 5--Asserting reset:

- Upon power up or depression of the Reset button, the HALT and RST pins of the 68010 are driven low by the power-up detection and reset switch debounce circuit.

Note

At power up, the RST and HALT pins must be low for a minimum of 100 ms. At reset, they must be low for 10 clock cycles.

- At 60 ns after reset goes high, the CPU outputs the function code--in this case, FC2+, FC1+, and FC0- (supervisor program mode). The processor drives R/W* high (specifying a read operation) and places the address on the PA bus. Once the processor has been vectored into ROM, PA23 is a 1, and PA22 is a 0.

Sheet 6--Asserting ROMLMAP*:

- RST* clears the addressable latch 7K, causing the assertion of ROMLMAP* from pin 7.

Sheet 5--Asserting SPA23 and PDS:

- ROMLMAP* is inverted to a high by 3K. This is applied to OR gate 13E. When it receives a high from 3K, it generates high output at SPA23.

- At 120 ns the processor asserts AS*, UDS*, and LDS*. LDS* and UDS* generate PDS+ via 22D.

Sheet 3--Asserting I/ORQ+:

- PDS+, SPA23, and an inactive INTA* cause gate 27F to assert I/ORQ+.

Sheet 9--Asserting BGACK*:

- I/ORQ+ causes the DMA address IC to assert BGACK*.

(This DMA address IC is a custom IC. In early versions of the **UNIX** PC logic boards, a piggyback gate array simulation board was used instead of custom ICs.)

Sheet 17--Asserting ROMEN*:

- 1 of 4 decoder 6J is enabled by BGACK* and addressed by SPA23 and PA22. SPA23 is high and PA22 is low, putting a binary 2 on the address inputs of the decoder. The decoder drives its Q2 output low, asserting ROMEN*.

Sheet 5--Addressing and enabling ROM:

- $\overline{\text{BGC}}$+ and $\overline{\text{MRAMEN}}$* force the output of NAND gate 25N (pin 11) low, enabling address buffers 16F and 17F. Transparent latch 17G is put into its transparent mode by gate 27H, pin 10 ($\overline{\text{MMUWREN}}$+ and $\overline{\text{MMUWREND}}$+), and the outputs are enabled by $\overline{\text{BGC}}$+.

- ROMEN* disables data transceivers 13C and 13D and enables boot ROMs 14C and 15C. Note that the ROMs output directly to the processor data bus.

- At this point the boot ROMs are addressed, enabled, and asserting data; the processor has been inserting wait states, waiting for DTACK*. When DTACK* arrives, the processor latches the data on the trailing edge of the next PCK* and deasserts address and control signals.

Sheet 9--Disabling BGACK*:

- 650 ns after I/ORQ+, the DMA address IC asserts I/O DTACK+; 900 ns after I/ORQ+, BGACK* is disabled.

Sheet 3--Asserting DTACK*:

- I/O DTACK+ causes gate 27N to assert DTACK*.

Mapping Virtual Address

The memory management circuitry includes page status registers on sheet 16 and the memory management PAL on sheet 2. It performs four general functions:

- Translates logical address to physical address
- Updates the page status registers during each legal RAM access, either by the 68010 or DMA
- If the access is illegal, inhibits the access and generates a memory management unit (MMU) error
- Provides a data path for the 68010 to write page status to the page registers while servicing an interrupt resulting from an illegal access to RAM.

The address output of the 68010 and the DMA devices are virtual address. Virtual address space is fixed at 4 MB. Programs are assigned fixed positions within this large address space. This virtual memory is sectioned off into 1024 pages of equal length (4096 bytes), and the pages reside in either RAM (physical pages) or disk storage (logical pages). From any user program viewpoint, these pages are available storage, and a major task done by the supervisor program is to juggle pages between RAM memory on the logic board and the hard disk drive.

The supervisor program uses map registers (static RAM chips) to locate pages in physical memory (RAM). The amount of physical address space can vary from 0.5MB to 4MB, depending on whether or not memory expansion boards are used. The map register's address input is a virtual or logical page address. Its data output lines are the physical page address plus status bits. These determine whether that page is located in the onboard RAM or in disk storage. There are 1024 map registers, one for each page of virtual memory.

When an instruction attempts to access a page that has been declared not present, the MMU PAL (memory management PAL, sheet 2) generates an error signal called PGF (page fault). This is used for internal processing and has priority over interrupts. To correct the page fault, a new page must be declared present. Since the number of pages that can be declared present is fixed, declaring a new page present means one that was present must be declared not present. The exception processing program determines which new page is declared present and which page that was present is declared not present. This determination is based on how long it has been since a given page was last accessed.

Typically the kernel of the operating system is contained in the lowest portion of memory. This section of memory is unity mapped. Unity mapping means a one-to-one correspondence between virtual address into the map RAMs and physical address output. Virtual page 0 is mapped to physical page 0, virtual page 1 is mapped to physical page 1, and so on. The kernel is the lowest level of the operating system. It is responsible for scheduling processes, executing command sequences to peripheral controllers, and performing other similar tasks never seen by a user running an application program.

The process working set is the set of pages that currently resides in physical memory. Whenever the CPU reads or writes a memory location, the memory management hardware determines whether or not the page addressed is in the process working set. If it is not, the hardware generates a BERR*.

Upon receiving a BERR*, the CPU invokes the supervisor. The supervisor sets up a DMA disk operation to obtain the missing page. Then it returns to CPU user mode, allowing it to do a different user process while the DMA is working. When the disk controller 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 upper size limit of the process working set. As memory size increases, the upper size limit increases, which improves system performance by reducing disk transfers.

Virtual program memory is being addressed if address bits 22 and 23 are equal to 00. The memory control receives the lower 21 virtual address bits, using the logical address bus. The upper 10 bits (21-12) address the map logic to select a page. The maps output a 10-bit-mapped address, MA21-12, which, when combined with the lower 11 address bits (11-1), forms the complete physical memory address.

When a user loads an application program into the computer, the supervisor program sets up the map RAMs for that application, thus mapping the application onto an unused portion of memory. As the user inputs more information, the supervisor sends data back to the disk, remembering where it has sent the data and to which application it belongs.

During each memory access, the map logic updates a table, called the page map table, to indicate the result of the access. This table consists of two page status bits, which indicate the status for each page of memory as follows:

- Not present (memory not installed at that address)
- Present but not accessed
- Accessed but not written to
- Written (in this case, the information in this page of memory must be stored on disk before the page can be overwritten)

Map Addresses MA12-MA21

The page mapping RAMs (integrated circuits 19C through 22C) are shown on sheet 16.

The page mapping RAMs receive 10 bits of virtual address, A12-A21, and output 10 bits of physical address, MA12-MA21. Bits MA12-MA19 are used to address onboard memory.

The page map is composed of static RAM chips (21C, 22C, 19C, and 20C). The RAM chips are addressed by 10 bits from the systemaddress bus (A12-A21). Note that the chips are always enabled on pin 8. The page RAMs output a 10-bit address (MA12-MA21), 5 status bits (PS0-PS4), and a write-enable bit (WE+).

Bits MA12-MA19 go directly to the memory address multiplexers (MUXs) and are considered physical address bits. MA21 is used to determine if memory is in base memory or on the memory expansion board. PS2, PS3, and PS4 are not used by the software as shipped, but are available for future memory management enhancements. PS0 and PS1 are used to provide the operating system with the following page status information:

PS0	PS1	Status of Page
0	0	Page not present
0	1	Present but not accessed
1	0	Accessed but not written
1	1	Written to (dirty)

Selecting RAM Row

Sheet 17 shows the RAM row decoder, 6J. RAM is divided into two rows of RAM chips, each row containing 512KB. Note that for the 0.5MB and 2.0MB designs, RAM chip locations are different. Address bit A1 is decoded to enable a particular row of RAM as follows:

LA1	Row Enabled
0	Row Y
1	Row Z

Address Multiplexing

The dynamic RAM chips (see sheets 19 and 20) have 9 address inputs that are multiplexed to form the required 18-bit address, although the 64K RAM chips use only 16 bits. This is done through the row address strobe (RAS*) and the column address strobe (CAS*) inputs to pins 4 and 15 of these chips. Sheet 18 shows the memory address MUXs. Multiplex chips switch the RAM address input from row to column address. Notice there are two sets of MUXs, one set for each of the two rows of memory. Each set of MUXs outputs to one of the two buses: Y bus, or Z bus. These buses carry 9 bits of address directly to their respective row of memory. They first transfer 9 bits of RAS address and then 9 bits of CAS address.

As inputs, each set of MUXs receives MA12-MA9, A3-A11, and LA2. MA12-MA9 is the page address from the outputs of the page-mapping RAMs. These bits address the selected physical page of memory. LA2, A3-A11 is the word address from the system bus. These bits are not altered by memory management. They address a word on the selected page. LA2, A3-A10 are gated through the MUXs at RAS time. A11 and MA12-MA18 are gated through at CAS time.

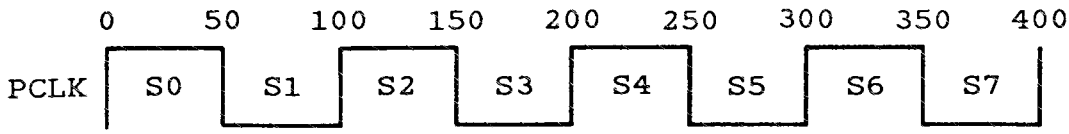
Table 2-4 Summary of Address Bit Assignments*

Address Bit	Assignment
PA23	When = 0, fast bus cycle memory fast register cycle When = 1, slow ROM cycle slow I/O access cycle
PA22	When = 0, fast cycle memory access slow ROM cycle When = 1, fast register cycle slow I/O access cycle
A12-A21	Virtual page address to map RAMs
MA21	When = 0, base memory enabled When = 1, expansion memory enabled
MA12-MA20	Address from page mapping RAMs
A2-A11	Address from system address bus
A1-A2	Word address bits from system bus used to select a bank of RAM chips

*This table reflects the address bit assignments for 1MB design.

Write Sequence: Processor to Page Map

A reference to the page mapping RAM is considered a CPU fast cycle, which is 400 ns or four cycles of the processor clock. When a time reference is made in the following text--for example, DTACK* is generated at 190 ns--it means 190 ns from the rising edge of the first PCLK in the cycle. Note the position of the asterisk in the following diagram:



Sheet 5--Asserting address and control signals:

- At 60 ns into the fast cycle, the 68010 outputs the function code: FC2, FC1, or FC0. In this case, FC2 is high because the CPU has to be in supervisor mode to refer to the page map.
- Also at 60 ns, the processor places the address on PA23-PA1. PA23 and PA22 will be 0 and 1, respectively, signifying a fast-cycle access to RAM.
- At 120 ns the CPU outputs AS-, UDS-, and/or LDS-. The processor also asserts R/W* low at this point.
- Gate 22D asserts PDS+.

Sheet 3--Starting memory timing:

- PA23 applies a low to pin 9 of 16H; pin 8 of 16H is an inactive BGC+ (low). 16H outputs high to 19K; pin 3 of 19K is clocked. 19K clocks F/F 21E clear because of the inactive ROMLMAP* tied to its K input. This asserts ENRAS* (low) and starts delay line 19H.
- ENRAS* is generated at 100 ns. Note that the signals generated from the delay line are time referenced to the generation of ENRAS*.

Sheet 9:

- ENRAS* (low) causes the DMA address IC to assert BGACK* for 200 ns. ENRAS* and R/W* assert FWR*.

The system address space allocated for the page mapping RAM includes addresses 400000-4007FF. These addresses are decoded to enable the page map.

PA23	PA22	PA21	PA20	PA19	PA18	PA17	PA16
0	1	0	0	0	0	0	0

Sheet 17:

- Decoder 6J is enabled by ENRAS*. It is addressed by SPA23 and PA22, which are 0 and 1, respectively. This causes the decoder Q1 output to go low, thus asserting GATE1*.

Sheet 4--Enabling map RAM:

- 1 of 8 decoder 26G is enabled by GATE1* (pin 5), PA19 = 0 (pin 6), the active SUPV+, and the inactive BGC* (pin 4). It is addressed by PA16-PA18, all equal to 0. The decoder drives pin 15 low, generating MRAMEN* (map RAM enable).

- MRAMEN* goes to sheet 5, where it disables the processor data transceivers, and to sheet 16.

Sheet 16--Page map:

- 19C, 20C, 21C, and 22C are the page-mapping RAM chips. They are 100 ns, 1K x 4, static RAMs.

- The processor addresses the page map with address bits PA1-PA10 through buffer chips 16G and 12F. These chips are enabled by MRAMEN*.

- The data is gated to the RAM chips using transceivers 23C and 24C.

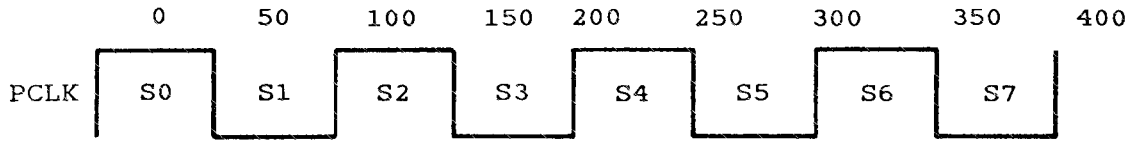
- The RAM is enabled for a write by gate 20E (pin 11).

Sheet 3--Acknowledging transfer by generating DTACK:

- 19F, acting as a MUX, is set up to select the A inputs because we are not in expansion memory. At T90 pin 9 goes high, partially enabling 27F. An inactive BGC* and an inactive BMSEL* fully enable the gate. The high output from 27F (pin 8) generates DTACK* from 27N (pin 5) at 190 ns.
- The processor samples DTACK* at 250 ns, verifies it at 300 ns, and latches the data at 350 ns.

68010 Local RAM Read Sequence

A reference to RAM is considered a CPU fast cycle, which lasts 400 ns or four cycles of the processor clock.



Sheet 5--Asserting address and control signals:

- At 60 ns into the fast cycle, the 68010 outputs the function code: FC2, FC1, or FC0. The R/W* signal is negated (high) because data is being read.
- Also at 60 ns, the processor places the address on PA23-1. PA23 and PA22 are 0, signifying a fast-cycle access to RAM.
- Processor address buffers 16F and 17F are enabled by an inactive MRAMEN* and an inactive BGC* from gate 25N.
- Transparent latch 17G is enabled on pin 1 by the inactive BGC+. It is put into transparent mode using 27H by an inactive MMUWREN+ and an inactive MMUWREND+.
- At 120 ns, the CPU outputs AS-, UDS-, and/or LDS-.
- Gate 22D generates PDS+.

Sheet 3--Starting memory timing:

- PA23 is low and feeds pin 9 of 16H; pin 8 of 16H is an inactive BGC+. 16H outputs high to 19K; pin 3 of 19K is clocked. 19K clocks F/F 21E clear because of the inactive ROMLMAP* tied to its K input. This generates ENRAS* and starts delay line 19H.

- ENRAS* is generated at 100 ns. Note that the signals generated from the delay line are time referenced to the generation of ENRAS*.

Sheet 16--Obtaining physical address from page map:

- The page mapping RAMs (19C, 20C, 21C, and 22C) are always enabled because of the ground on pin 8. Gate 26M sets them up for a read because of the inactive MRAMEN*.

- The RAMs are addressed by virtual address bits A12-A21 and output physical address bits MA12-MA21 along with five page status bits.

Sheet 18--Driving the address to memory:

- The address MUXs are always enabled by a ground on pin 15. The select input (pin 1) is high for the first 160 ns of the cycle and gate through the row address. After 160 ns, the column address is gated to memory.

Sheet 16--Strobing the row address:

- Transparent latch 25F is in its transparent mode before T150 and passes address bits A1 and A2 through to become LA1 and LA2. At T150 pin 11 goes from high to low and 25F latches its data.

Sheet 17:

- The RAM row decoder (6J) is enabled by a low PA22 using gate 20F. LA1 and LA2 select one of the four banks of RAM as shown in Table 2-5, the respective 4J gate and partially enables one of the 3J gates. At 130 ns, RAS goes out to the selected row of memory (T30*).

- On the 1MB machine, RAS is distributed by an LA1 alone. On 0.5 or 2MB machines, the LA1 and LA2 states are as follows.

Table 2-5 Row Address Strobing

LA2	LA1	RAS Enabled	Row Affected
0	0	RAS0*	Row W
0	1	RAS1*	Row X
1	0	RAS2*	Row Y
1	1	RAS3*	Row Z

Sheet 2--Strobing the column address:

- With PA22 low and BGC+ inactive, gate 26M outputs low to 22D, generating CASEN+.
- MMU PAL 24G outputs only CASDIS* during a MMU error. This signal prevents a memory reference in the event of an error condition (such as a page fault).

Sheet 16:

- The inactive CASDIS* and CASEN+ feed gate 24P, forcing a low onto pin 5 of 27H. Pin 6 is low for a read operation. 27H outputs high to 26M and 26M outputs high to 25F, generating ENCAS+.

Sheet 17:

- ENCAS+ partially enables decoders 7J and 9J, LDS* and UDS* further enable the decoders, and at T90, one or both of the decoders are fully enabled. Note that LDS* and UDS* control byte or word selection.
- The decoders are addressed by LA1, LA2, and LMA21. If LMA21 is low, LA1 and LA2 generate CAS to one of the rows of memory. If LMA21 is high, the target memory address is on the expansion board and no CAS goes to base memory.

Sheet 19--Accessing RAM:

- At 130 ns into the bus cycle, RAS strobes in the row address. At 160 ns, the address MUXs switch from row address multiplexing to column address multiplexing. At 190 ns, CAS strobes in the column address.
- After access time (150 ns from RAS), the data is available on the RD bus.

Sheet 20--Putting information on the system data bus:

- 12C and 12D are the RAM data transceivers. They are enabled during CAS time, provided address bit 21 is a 0 (base memory).
- The direction in which the transceivers pass data is controlled by R/W* from the processor. For a read operation, they route the data from the RD bus to the D bus.

Sheet 3--Stopping memory timing and resetting ENRAS* using PDS+:

- Gate 18G stops memory timing. Pin 13 is high when the bit map is not being referenced. Pin 12 is high when PDS+ is active. Pin 11 is high when no bus grant common is active, and pin 1 goes high at T120. These conditions cause 18G to output a low to gate 16H.
- 16H outputs high to the D input of F/F 16K. At the next 1PCK+, MMUWREN+ is generated.
- MMUWREN+ is tied to the D input of F/F 18H, and the trailing edge of the next PCK* sets the F/F.
- 18H outputs low on pin 3 and sets the ENRAS F/F (21E).

Sheet 3--Acknowledging the transfer and generating DTACK*:

- 19F, acting as a MUX, is set up to select the A inputs because we are not in expansion memory. At T90 pin 9 goes high, partially enabling 27F. An inactive BGC* and an inactive BMSEL* fully enable the gate. The high output from 27F generates DTACK* from 27N at 190 ns.
- The processor samples DTACK* at 250 ns, verifies it at 300 ns, and latches the data at 350 ns.

Sheet 10--Checking parity:

- The DMA data IC outputs UPARIN if bad parity (odd) is found in the high byte and LPARIN if a problem occurs in the low byte. Parity is checked during a read. During a write, LPARIN and UPARIN set the data into the memory parity chips so that the data byte plus the parity bit have odd parity.

Sheet 17:

- Gate 13H outputs low at CAS time when a read-to-base memory is being done. This output, along with the data strobes, enables one or both of the gates feeding 12H.

- 12H outputs low when there has been a parity error. When T90 goes from low to high, F/F 6K resets and 20E presents PERR* to the system.

Sheet 6:

- PERR* forces a low out of 28F. If the error-enable bit is active, a low is gated to pin 4 of 18F. Decoder 18F presents a level 7 interrupt to the processor.

Parity

Figure 2-2 is a simplified schematic of the parity generator circuit in the custom IC simulator board.

Parity Write

During a RAM write operation, the parity circuit determines if the number of bits set to a one in the upper or lower data byte is odd or even. If the number is even, the parity bit for that byte is set high. If the number is odd, the parity bit is set low. Parity is checked during a 68010 RAM read cycle. If parity is incorrect, a level seven interrupt is generated. The ENCAS* input to the parity error generating circuit on sheet 17 prevents generation of a parity error during a page fault. The LDS input prevents parity error during a DMA RAM memory access.

Sheet 17:

- During a write, the BP+ bit is low and the select input S to the multiplexer 10J is high; thus UMUXPAR and LMUXPAR are both high during a write.

The odd-parity generators in the DMA data IC (sheet 10) generate the odd-parity signals UPARIN (upper-byte parity) and LPARIN (lower-byte parity). If the data on the data bus has an even number of high bits, the hi input from UMUXPAR makes the number of high inputs to the 74LS280 odd, and the parity bit input to RAM is hi. If the number of highs on the data bus is odd, then adding the hi input from UMUXPAR or LMUXPAR keeps the total input to 75LS280 even and the output of the DMA data IC is low. Thus the two combined inputs to RAM, D0-D7 plus LPARIN and D8-D15 plus UPARIN, always have odd parity.

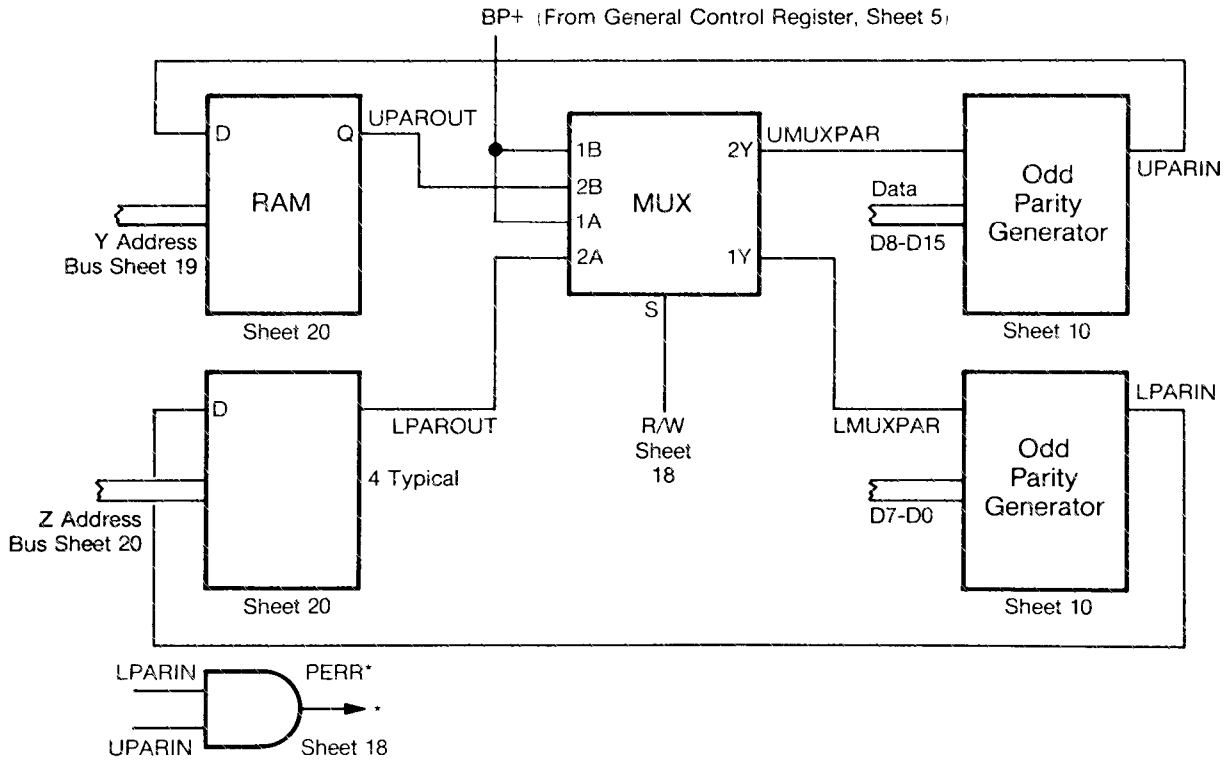


Figure 2-2 Parity Generator

Parity Check

During a memory-read operation, the two parity generator chips check the parity of the 8-bit bytes plus their corresponding parity bit. Thus, during a RAM read, both UPARIN and LPARIN should be low, indicating even parity. If either signal is high, circuitry on sheet 18 generates a parity error (PERR*). The PERR* signal is used by the 68010 interrupt logic on sheet 6. An interrupt is generated if the parity error-enable is asserted at the general control register (GCR). The PERR* signal latches the current bus status into the bus status register (BSR) and the general status register (GSR). If the supervisor program has not masked out the error with the error-enable bit at the GCR, the PERR* generates a level-7 interrupt.

Page Status Update and Memory Management Errors

Each memory access to RAM is checked for validity. If it is a valid access, the status bits in the map RAMs are updated. If it is not valid, the attempted access to memory must be prevented and an error signal must be generated. The circuitry to accomplish this is explained below.

The memory management unit write (MMUWR*) signal originates on sheet 2. On sheet 16, it updates the status of the page status registers. This signal is asserted each time an access to RAM is made that does not produce an access error. On sheet 2 MMUWR* enables tristate buffers that load the new page status bits PS0 (page status bit zero), PS1, and WE (write enable). On sheet 16 MMUWR* puts the map status RAMs into a write condition so that the new status of PS0, PS1, and WE is written into the RAMs during each valid RAM access.

On sheet 2 MMUWR* is asserted by MMUWREN+ (memory management enable) high and MMUERR* (memory management unit error) high, no error, and CASEN+ (column address strobe enable).

MMUWREN+ is enabled each time an access to RAM is made by the 68010 or the DMA interface unit.

The memory management PAL generates the following new page status:

- PS1: Set to 1 regardless of previous state to indicate a read has taken place.

- PS0: Set to 0 during a memory read if there is no memory error. Otherwise, it is set to 1.

- WE+: Set equal to SPA23. Pages are write enabled when and only when they are accessed in supervisory mode by the 68010.

Generation of Memory Management Interrupts

If the EE+ bit is set high on sheet 6, the MMUERR* generates a level-7 interrupt. The memory management PAL generates two signals that drive the interrupt logic on sheet 6: PGF (page fault) and MMUERR (memory management unit error).

Page Fault

This signal is asserted whenever both PS0 and PS1 are 0 and the 68010 is executing a user program, FC2 is low, or a DMA RAM access (BGC+ asserted) is taking place. When PS0 and PS1 are both 0, a page fault is not generated when the 68010 accesses the page in supervisory mode.

Memory Management Error

The three ways to generate a memory management error are:

- Page fault

- User attempt to access the kernel, indicated by:

  Kernel access--A19, A20, and A21 all low
  User program execution--PA22, SUPV, and BGC all low

- User access to a page that is not write enabled, indicated by:

  Page not enabled for a write--LWE low
  User program execution--PA22, SUPV, and BGC all low

Bus Error

The memory management unit also has an output called BERR (bus error). This signal goes directly to the 68010 on sheet 4. It is caused by the following conditions:

- CPU page fault: page not present

- User I/O fault (access to address not in RAM)
- User attempt to write to a page that is not write enabled
- User attempt to write to the kernel.

Column Address Strobe Disable

The column strobe disable (CASDIS) signal is generated by the memory management PAL to prevent the completion of an illegal access to RAM.

Refresh Operation

Circuitry for refresh is contained in the DMA address IC shown on sheet 9. The dynamic RAM chips in physical memory (sheets 19-22) must be refreshed at least every 4 ms to prevent data loss. A RAS-only refresh is used--that is, 256 row addresses are strobed to all rows of RAM every 4 ms. This means that 64 refresh cycles must be accomplished every millisecond.

Two 8-bit counters in the DMA address IC constitute most of the refresh circuitry. One is used as an address counter and the other is used to time the refresh requests.

The request counter is clocked by 1PCK+. It presents RFRQ* to the bus arbitration circuitry on every 144th 1PCK+. In other words, it is configured as a divide by 144 circuit.

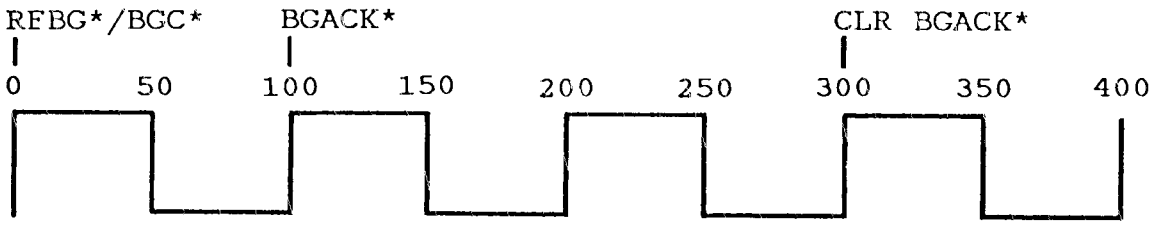
The address counter is incremented by the refresh bus grant (RFBG*). It drives row addresses onto the system bus. This counter-cycles through the 256 row addresses (0-FF), driving a different address onto the system bus for each refresh bus grant.

Sheet 9--Getting bus control and driving address:

- The DMA address IC asserts RFRQ* on every 144th occurrence of 1PCK+ (every 14.4 u sec).
- When the RFRQ* has priority, the arbiter PAL grants the refresh circuitry control of the system bus by returning RFBG* and BGC* (bus grant common).

- The DMA address IC acknowledges receipt of the bus grant by asserting BGACK*, which causes the bus arbiter to latch RFBG* and BGC*.

- BGACK* also determines how long mastery of the bus lasts by latching the grants. BGACK* remains active for two clock cycles during a refresh.



- The refresh circuitry internal to the DMA address IC then drives the 8-bit refresh address onto the system bus (A3-A10).

Sheet 3--Starting memory timing:

- F/F 18H (pin 6) is set when the system is idle, because BGC+ and ENRAS+ are absent. F/F is used to start memory timing for bus masters other than the CPU.

- When bus control is granted to the refresh circuitry by the bus arbiter, the common bus grant is generated (BGC+).

- This causes F/F 18H to reset, 19K clocks F/F 21E clear, and memory timing is started.

Sheet 17--Refreshing RAS:

- RFBG* enables all of the 4J gates to output lows, partially enabling the 3J gates. T30* fully enables the 3J gates, asserting RAS to all rows of memory.

Sheet 2:

CASEN+ is not generated because gate 26M is broken by BGC+, and gate 22D is broken by an inactive NRF/BGC*.

System Control and Status Registers

Various control and status registers are available to the processor. These registers monitor system status and control the hardware. Tables 2-6 through 2-11 describe these various registers.

Table 2-6 General Status Register--Address 410000 (Read Only)

Data Bit	Signal	Description
14	R/W-	0 = Write cycle, 1 = Read Cycle
13	NPC+	1 = Nonprocessor cycle
12	PF-	0 = Page fault; applies to the processor (supervisor or user) and the DMA controller
10	PIE+	1 = Parity interrupts enabled
08	UIE-	0 = User access to memory above 4MB

Information in the general status register is updated with each bus cycle. If the current cycle causes a parity error, MMU error, or processor bus error, the information in the GSR is not updated at the following cycles until the clear-status, register signal is received. This signal is generated when the processor writes to address 4C0000.

The general status register is on sheet 6 (24D) of the schematics.

Table 2-7 Bus Status Register 0--Address 430000 (Read Only)

Data Bit	Signal	Description
15	MMUERR-	0 = MMU error; three possible causes: (1) Processor or DMA page fault (2) User write to write-protected page (3) User access to kernel (0-512K)
14	DKBG-	0 = Disk DMA cycle
13	EXP0BG-	0 = Expansion board 0 DMA cycle
12	EXP1BG-	0 = Expansion board 1 DMA cycle
11	EXP2BG-	0 = Expansion board 2 DMA cycle
10	EXP3BG-	0 = Expansion board 3 DMA cycle
09	UDS-	0 = Upper byte access
08	LDS-	0 = Lower byte access
07	PA23	Processor address bit 23
06	PA22	Processor address bit 22
00-05	A16-A21	Logical address bits 16-21

Note

The bus status register is on sheet 6 of the schematics.

Table 2-8 Bus Status Register 1--Address 440000 (Read Only)

Data Bit	Signal	Description
00-15	A00-A15	Latches address during NMI or BERR

Note

The bus status registers (BSR0, BSR1), located on sheet 6 in the schematics, are composed of transparent latches: 18C and 25D are BSR0; 16B and 21D are BSR1. They latch information pertaining to the status of the bus when there has been a bus, parity, or MMU error. They can be useful when troubleshooting hardware problems or debugging code. The BSRs are not cleared by reset.

Table 2-9 General Control Register--Address E4X000 (Write-Only)

Address	Bit	Signal	Description
E40000	15	EE+	1 = Error enable 0 = Error disable, meaning no level-7 interrupt or bus error can occur
E41000	15	PIE+	1 = Parity error circuit and interrupt are enabled 0 = Parity is disabled
E42000	15	BP+	1 = Memory-write cycle resulted with parity error 0 = Memory-write cycle resulted with good parity
E43000	15	ROMLMAP	0 = Processor is forced into ROM address space 1 = Normal addressing
E44000	15	L1 MODEM*	0 = Modem connected to line 1
E45000	15	L2 MODEM*	0 = Modem connected to line 2
E46000	15	D/N CONNECT*	0 = Dial network connected to line 1

Note

The general control register is located on sheet 6 of the schematics. It is an addressable latch (7K). The processor addresses the GCR with address bits A12, A13, and A14. The various control signals are enabled and disabled using processor data bit 15.

Table 2-10 Miscellaneous Control Register--Address 4A0000 (Write-Only)

Data Bit	Signal	Description
15	CLRSINT-	Dismisses the level-6, 60-Hz interrupt. To dismiss the interrupt, the bit must be toggled from high to low and back to high. When this bit is low, it makes the interrupt.
14	DMAR/W-	0 = Disk DMA write operation 1 = Disk DMA read operation. This bit, along with the IDMAR/W- bit in the disk DMA count register, should be updated before a disk DMA operation.
13	LPSTR+	Strobes data from the line printer data register to the line printer through Centronics interface protocol. After data is set up at the data register, LPSTB+ must be toggled from low to high and back to low to strobe the data to the printer.
12	MCKSEL-	0 = Modem RX clock and TX clock gated to the communication controller's RX clock and TX clock inputs 1 = Programmable timer selected to generate the clock pulses for the communication controller.
11	LED3-	0 = Red LED 3 (CR21) on 1 = Off
10	LED2-	0 = Green LED 2 (CR22) on 1 = Off

Table 2-10 Miscellaneous Control Register -- Address 4A0000 (Write-Only) (Continued)

Data Bit	Signal	Description
09	LED1-	0 = Yellow LED 1 (CR23) on 1 = Off
08	LED0-	0 = Red LED 0 (CR24) on 1 = Off

Note

The miscellaneous control register is located on sheet 15 of the schematics.

Reading General Status Register (Address 410000)

The processor accesses the system status and control registers by performing a fast bus cycle. This operation is similar to the previous fast cycles discussed. The processor drives address and control signals and waits for DTACK*. The address decode is the only difference between this and previous fast cycles.

PA23	PA22	PA21	PA20	PA19	PA18	PA17	PA16
0	1	0	0	0	0	0	1

Sheet 4--Decoding address:

- GATE1* is active for all fast cycles other than accesses to main memory. This signal enables decoder 26G on pin 5.
- The processor must be in supervisor mode to access the GSR, and BGC* is not active during process bus cycles. These conditions cause gate 25N to enable the decoder on pin 4.
- When addressing the GSR, PA19 = 0, which fully enables the decoder (pin 6).
- The decoder is addressed by PA18, PA17, and PA16. As the chart above illustrates, these bits are 0, 0, and 1, respectively. This causes the decoder to generate GSRRD* from pin 14.

Sheet 6--Driving the data:

- The general status register is transparent latch 24D. Its outputs are enabled by GSRRD*, and it Sheet 3--Acknowledging data transfer:
- Gate 27F causes the generation of DTACK* at T90. The processor latches the data on the trailing edge of the third PCK+, the DMA address array.

Interrupting the Processor

Interrupts inform the processor of special situations within the system. They interrupt normal processing and cause the CPU to be vectored into a routine to deal with the special situation. Motorola calls this exception processing.

The processor can mask interrupts by manipulating bits 12, 11, and 10 of the 68010's status register. For example, if these bits are loaded with a binary 100, interrupt priority levels (IPL) of 4 and below are ignored. IPL7 interrupts cannot be masked; this is the nonmaskable interrupt (NMI). If bits 12, 11, and 10 are all set to 1's, all interrupts except the level-7 interrupt are masked.

Exception processing is the only way to go from the user state to the supervisor state. It occurs in these four steps:

- A temporary copy of the status register is made and the status register is set for exception processing (all interrupts equal to the current level and below are masked and the S bit is set, which puts the processor into the supervisor state).
- The exception vector is determined.
- The current processor context is saved (status register and program counter are pushed onto the supervisor stack).
- New context is obtained and instruction processing resumed.

Exception Vectors

Exception vectors are memory locations from which the processor fetches the address of a routine to handle an exception. Exception vectors are words in length, and all exception vectors lie in supervisor data space. Exception vectors are obtained through the use of a vector number, which is an 8-bit value that, when multiplied by 4, gives the offset of the exception vector. The 68010 can accept an externally generated, vector number or generate the vector number internally.

As implemented in the **UNIX** PC, all vector numbers are generated internally by the 68010 microcode. This is called auto-vectoring. Once a vector has been calculated, the processor fetches an address from that location in memory. The processor loads the address into the program counter and begins executing instructions. Typically, it is then vectored into the interrupt service routine.

Table 2-11 shows the interrupt vector number generated for each interrupt and the hexadecimal offset calculated from each vector number. If a level-2 interrupt is received, the 68010 generates a vector number of 26. It then reads two words beginning at address 68H, loads the words into its program counter, and begins executing instructions.

Table 2-11 Interrupt Vector Numbers and Hexadecimal Offsets

Vector Number (Decimal)	Offset (Hex)	Assignment
25	64	Level 1 interrupt autovector
26	68	Level 2 interrupt autovector
27	6C	Level 3 interrupt autovector
28	70	Level 4 interrupt autovector
29	74	Level 5 interrupt autovector
30	78	Level 6 interrupt autovector
31	7C	Level 7 interrupt autovector

The interrupt logic is located on sheet 6 of the schematics. Seven levels of interrupts are implemented through the use of three interrupt priority level signals (IPL0, IPL1, and IPL2). IPL7, the NMI, is the highest priority interrupt. It cannot be masked by software. IPL1 is the lowest priority interrupt. An interrupt priority level of 0 means no interrupts are pending. The following table lists the interrupts and their priority levels:

Table 2-12 Interrupts and Their Priority Levels

IPL	Interrupt
7	Nonmaskable interrupt
6	Realtime interrupt
5	Expansion Board Interrupt
4	RS-232 interrupt
3	Keyboard interrupt
2	Floppy/hard disk/line-printer interrupt
1	Expansion Board Interrupt
0	No interrupts

Interrupts: Theory of Operation

The interrupts operation is as follows:

Sheet 6:

- The various interrupts are active low inputs to encoder 18F. Pin 4 is the NMI, the highest priority interrupt. If this interrupt is active, the encoder drives IPL0, IPL1, and IPL2 all high. This presents a binary-encoded, IPL7 interrupt to the processor.

- Because pin 10 of the encoder is grounded and if no interrupts are present, the encoder drives IPL0, IPL1, and IPL2 low. This informs the processor that no interrupts are present. The IPL signals input directly to the processor.

Sheet 5:

- If any of the IPL inputs to the processor are low, the 68010 detects an interrupt. The processor then acknowledges the interrupt and begins exception processing.

- When the 68010 acknowledges an interrupt, it drives FC0, FC1, and FC2 high. Address bits A4-A23 are also driven high.
- Gate 13H generates a signal called INTA*, which goes to sheet 3 and prevents an I/O request from being generated.
- 13H also partially enables 13E. When PAS* is asserted, 13E generates VPA* (valid peripheral address). During an interrupt acknowledge, this signal replaces DTACK* and handshakes the processor. It also tells the processor that the interrupt being acknowledged is an autovector interrupt. Since this signal is generated with every interrupt being acknowledged, all interrupts are autovectored.

Note

Acknowledging the interrupting peripheral is the responsibility of the software. Typically, this is handled in the interrupt service routine.

Disk Direct Memory Access (DMA)

This section explains logic-board circuitry that transfers information between a disk drive and RAM memory.

Disk Format and Flow

Data from a disk drive passes through three sections of hardware before it reaches RAM memory:

- Serial data from the disk drive, consisting of both data and clock pulses, is separated into a clock signal and a data signal. For the hard disk, this is done by the data separation circuitry shown on sheet 12 of the schematics. For the floppy disk, it is done by the WD2797 floppy disk controller.
- Serial data is assembled into 8-bit bytes by the disk controller. The disk controller for the hard disk is the WD1010; for the floppy disk, it is the WD2797.
- The disk DMA bus interface assembles 8-bit bytes into 16-bit words and writes them into a RAM memory location. The RAM address is contained in the bus interface.

Disk Program Summary

Data on disk is located by three parameters: cylinder, head, and sector. The disk DMA bus interface has two parameters: DMA starting address and word count. These parameters plus the disk commands are written in the following disk program sequence summary:

- System register disk bits are set. These bits select the disk drive and head number; in the case of the floppy disk, they turn on the motor.

  Counters in the DMA disk bus interface are loaded with the starting address and number of words to be transferred. The disk controller registers are loaded with the track and sector to read.

- The 68010 starts the transfer of data from disk by writing a Read-Sector command to the disk controller.

- The disk DMA bus interface takes control of the bus through bus arbitration and transfers data from the disk controller into RAM memory. Data passes through the three sections of hardware as described above. One disk DMA bus interface unit is shared by both the floppy and hard-disk controllers.

  When the sector of a disk is being read, the disk DMA bus interface reads two 8-bit bytes from either the WD1010 hard disk controller or the WD2797 floppy-disk controller, assembles them into one 16-bit word, and writes the word into RAM memory.

  When data is being transferred from RAM memory to disk, the disk DMA bus interface reads a 16-bit word from RAM memory, separates it into two 8-bit bytes, and loads them into the data register of either the WD1010 hard disk controller or the WD2797 floppy disk controller.

  The disk DMA bus interface consists of two custom, gate-array chips: the DMA address IC shown on sheet 9 and the DMA data IC shown on sheet 10. Once a Read Sector command is written to the disk controller command register, the disk DMA bus interface takes control of the bus through bus arbitration each time a word is to be transferred to memory. Between word transfers, the 68010 can access the bus. The software must be written so that the 68010 does not write to the disk controllers while a command is in progress.

- The disk controller informs the 68010 when it has completed a command by asserting its interrupt output.

Definitions

The following definitions describe the register and commands relationship.

Disk Commands

Disk commands control the movement of data to and from memory to the disk storage medium. The disk commands are described below.

DMA read and DMA write are defined with respect to the disk DMA bus interface that is the bus master during the transfer. During a DMA read, data is moved from RAM to the word buffer in the bus interface. During a DMA write, data from the word buffer in the disk DMA bus interface is written to RAM.

The disk controller Write-Sector and Read-Sector commands are defined with respect to the controller. A Read-Sector command moves data from the disk to the disk controller. A Write-Sector command moves data from the controller to the disk, and the controller writes to disk.

Thus, to move data from RAM to disk requires a DMA read and a disk, Write-Sector command. To move data from disk to RAM requires a disk, Read-Sector command and a DMA write transfer.

System Register Disk Bits

Disk operation is selected by writing instruction to the disk control register as explained below.

Disk Control Register (Drive and Head Select), Table 2-13.

Before a disk operation can occur, the drive and head must be selected. The processor does this by writing to address 4E0000, the disk control register (sheet 11).

Table 2-13 Disk Control Register--Address 4E0000 (Write Only)

Data Bit	Signal	Description
07	FDRST-	0 = Floppy disk controller reset 1 = Not reset
06	FDR0+	0 = Floppy drive 0--not selected 1 = Floppy drive 0--selected
05	FDMTR+	0 = Floppy drive motor is not on 1 = Floppy drive motor is on
04	HDRST-	0 = Hard disk controller reset 1 = Not reset
03	HDR0+	0 = Hard disk drive 0--not selected 1 = Hard disk drive 0--selected
00-02	HDHSEL 0-2	These three bits are decoded to select the head

Miscellaneous Control Register (DMA Read/Write), Table 2-14.

This register controls the DMAR/W- bit used by the disk PAL shown on sheet 2 and other bits shown in the following table:

Table 2-14 Miscellaneous Control Register--Address 4A0000 (Write Only)

Data Bit	Signal	Description
15	CLRSINT-	This bit dismisses the level 6, 60-Hz interrupt. To dismiss the interrupt, the bit must be toggled from high to low and back to high. When this bit is low, it masks the interrupt.
14	DMAR/W-	0 = Disk DMA write operation 1 = Disk DMA read operation. This bit, along with the IDMAR/W- bit in the disk DMA count register, should be updated before a disk DMA operation.
13	LPSTR+	This bit strobes data from the line printer data register to the line printer through Centronics interface protocol. After data is set up at the data register, LPSTB+ must be toggled from low to high and back to low to strobe the data to the printer.
12	MCKSEL-	0 = Modem RX clock and TX clock are gated to the communication controller's RX clock and TX clock inputs. 1 = Programmable timer is selected to generate the clock pulses for the communication controller.
11	LED3-	0 = Red LED 3 (CR21) on; 1 = off
10	LED2-	0 = Green LED 2 (CR22) on; 1 = off
09	LED1-	0 = Yellow LED 1 (CR23) on; 1 = off
08	LED0-	0 = Red LED 0 (CR24) on; 1 = off

Line Printer Status Register (Controller Interrupt), Table 2-15.

Bits D2 and D3 of this register are set high when the disk controller completes a command. These bits may be polled by the 68010. They are reset when the status register of the controller is read.

Table 2-15 Line Printer Status Register--Address 4A0000 (Write Only)

Data Bit	Signal	Description
07	LPBUSY+	1 = Line printer is busy
06	LPSELECT+	1 = Line printer is selected
05	LPNOPAPER+	1 = Line printer has no paper
04	ERROR*	0 = Line printer error condition
03	FDINTRQ+	1 = Floppy disk controller interrupt
02	HDINTRQ+	1 = Hard disk controller interrupt
01	PERR*	0 = Main memory parity error has been detected; this bit can be cleared with the CSR command (write to 4C0000)
00	DTDET*	0 = Dial tone is detected
15	CLRSINT-	This bit dismisses the level 6, 60-Hz interrupt

Hard Disk Controller (WD1010)

Hard disk control is provided by the Western Digital WD1010 Winchester disk controller chip. It provides MFM-encoded data and all of the control lines required by disks using the Seagate Technology ST506 or Shugart SA1000 interface standard.

Pin Functions (sheet 11) for the WD1010 are listed in Table 2-16.

Table 2-16 Pin Functions WD1010

Pins	Signal Description
12-19:	Winchester data bits 0-7; used during programming and while transferring data to and from the drive.
9-11:	Address bits 0-2; used by 68010 while programming and reading the chip's various registers.
1:	Buffer chip-select; asserted by WD1010 to enable the reading of or writing to the RAM data buffer.
8:	Chip-select; enabled by 68010 when reading of or writing to a WD1010 register is desired. When CS is low, pins 6 and 7 are used as inputs.
7:	Write-enable; when used as an input, it enables the bus master to write into the controller's internal registers. When used as an output, it allows the controller to write to the RAM buffer.
6:	Read-enable; when used as an input, it allows the bus master to read the controller's internal registers. As an output, it allows the controller to read from the data buffer.
31:	Track 000; asserted by selected drive when heads are located over track 0.

Table 2-16 Pin Functions WD1010 (Continued)

Pins	Signal Description
32:	Seek complete; asserted by selected drive when heads are settled over selected track.
28:	Drive ready; asserted by selected drive to signal its capability to do reads, writes, and seeks. When signal goes low, all commands are deactivated.
30:	Write fault; asserted by selected drive if a fault is detected. When low, all commands are deactivated.
29:	Index; pulsed when index mark is detected. Indicates beginning of track.
35:	Buffer ready; normally used by data buffer to signal to the controller that it is ready to be read (full) or written to (empty). As implemented in the **UNIX** PC, this input is tied to a 0.6144-MHz clock that constantly tells the controller that the buffer is accessible.
3:	Interrupt request; generated by WDC upon termination of a command; cleared when status register is read.
5:	System reset.
27:	Step; 8.4-us pulses to drive stepping motor.
26:	Direction; when high, the heads move inward toward higher cylinder numbers. When low, the opposite is true.
23:	Early; output used to derive delay value for write precompensation. Valid when write gate is active.

Table 2-16 Pin Functions WD1010 (Continued)

Pins	Signal Description
22:	Late; output used to derive delay value for write precompensation. Valid when write gate is active.
33:	Reduced write current; goes high for all cylinder numbers greater than the value programmed into the write precompensation register.
24:	Write gate; an active high when write data is valid; used by drive to enable write current to data heads.
37:	Read data; MFM data pulses from disk.
25:	Write clock; 5-MHz clock used to derive write data rate.
34:	Data run; looks for a string of 0s in the read data, indicating the beginning of an ID field. If the 0s are detected, read gate (RG) is brought high.
38:	Read gate; set high for data and ID fields.
21:	Write data; open drain output that shifts out MFM data at the speed determined by the write clock input.
39:	Read clock; square-wave clock input derived from external data recovery circuits.

Task File Registers

The WD1010 is similar to the other LSIs that have been discussed in that it must be programmed before hard disk operations. The 68010 is required to supply information--such as drive, head, sector, number of sectors to transfer, and so on--to the task file registers before writing a command to the command register.

Table 2-17 Task File Registers

Address	Function
Data register (E00000)	Used during reads and writes to hold one byte of data
Error register (E00002)	Read by 68010 to determine what type of error occurred during a disk operation
Sector count register (E00004)	Data bits 0-7 are used during reads and writes to indicate the number of sectors to transfer
Sector number register (E00006)	Data bits 0-7 are used during reads and writes to indicate the sector address
Cylinder number low register (E00008)	Data bits 0-7 are used during reads and writes to indicate the current cylinder number
Cylinder number high register (E0000A)	Holds the upper byte of the current cylinder address
Sector drive head register (E0000C)	Bits 0-2, head number; bits 5-6 will be 0 and 1 indicating 512-byte sector; bit 7 is set for ECC and reset for CRC
Status register (E0000E)	Read by 68010 to determine current status of hard disk control circuits and data buffer
Command register (E0000E)	Written into by the 68010 to control step rate and issue disk commands

Error Register--E00002 (Read Only)

Bit 7: Bad block detect used for bad sector mapping.

Bit 6: CRC error.

Bit 5: Forced to 0.

Bit 4: ID not found. This bit is set to indicate that the correct cylinder, head, sector number, or size parameter could not be found or that a CRC error occurred on the ID field. This bit is set on the first failure and remains set even if the error is recovered on a retry. When recovery is unsuccessful, the error status bit is set also.

Bit 3: Forced to 0.

Bit 2: Aborted command. This bit is set if command is issued while the DRDY is deasserted or the WF is asserted. The aborted command bit is also set if an undefined command code is written into the command register.

Bit 1: Track 0 error. This bit is set only by the Restore command. It indicates that TK000 has not gone active after the issuance of 1K of stepping pulses.

Bit 0: Data address mark not found. This bit is set during a Read-Sector command if the data address mark is not found after the proper sector ID is read.

Status Register--E0000E (Read Only)

Bit 7: Busy. This bit is set whenever the WD1010 is accessing the disk. Commands should not be loaded into the command register while the busy is set.

Bit 6: Ready.

Bit 5: Write fault.

Bit 4: Seek complete.

Bit 3: Data request.

Bit 2: Always 0.

Bit 1: Command in progress.

Bit 0: Error. This bit indicates that a nonrecoverable error has occurred. If bit 0 is high, read the error register (E00002) to determine the type of error written into the control register.

Commands

The six commands that can be issued to the WD1010 are described in the following table:

Table 2-18 WD1010 Commands

Command	Function
Restore	Returns heads to track 0
Seek	Moves heads to cylinder in cylinder number register
Read Sector	Reads the contents of a sector
Write Sector	Writes data to a sector
Scan ID	Updates head, sector, and cylinder registers
Write Format	Formats one track

If the track selected in the cylinder registers is not the same as the track that the heads are currently positioned over, and a read or write command is issued, the controller executes an implied seek.

Typically, the WD1010 can be commanded to seek a track and then write to a sector. It is important to remember that the controller has to do a read before it can write. The reason for the read is that the controller has to locate the requested sector before it can write information to that sector. The WD1010 does this by reading the ID fields. Once it locates the target sector, the controller reads the first byte from the data buffer, converts the parallel data to MFM data, and transfers this MFM data to the disk.

Disk DMA Bus Interface Registers

Address Register--4DXXXX

When the DMA address counter is loaded, the four least-significant digits of the address bus contain the DMA address. The DMA address counter contains two sections: one to hold the two least-significant hexadecimal digits of DMA address and one to hold the four most-significant digits of DMA address.

To load the counter requires two write cycles. A14 selects the section to be loaded. When A14 is low, the least-significant DMA address counter is selected, see Table 2-19. In this counter, A1-A8 of the address bus correspond to A1-A8 of DMA address. When A14 is high, the most-significant DMA counter is selected. In this counter, A1-A13 of the address bus correspond to A9-A21 of DMA address.

Thus the maximum DMA address that can be loaded is 3FFFFF. This address is loaded in two writes as listed in the following table (note that the contents of the data bus are not used):

Table 2-19 DMA Address Count Register Selection

Address	Data	Description
4D00FF	xxxx	A14 is low; two least significant digits loaded
4D7FFF	xxxx	A14 is high; four most significant digits loaded

The counter is read and incremented when DKBG is asserted during execution of a Read/Write Sector command.

Count Register

The DMA count register is contained in the DMA data IC shown on sheet 10. The bit definitions for this counter are shown in Table 2-20.

The DMA count is incremented after each DMA transfer. When the count reaches 3FFFx, DMA operation is terminated; thus, the maximum number of words to be transferred for a single DMA operation is 16K words. The content of this register should not be modified during a DMA operation unless DMA abortion is desired by deassertion DMAEN+.

Table 2-20 Disk DMA Count Register--Address 460000

Data Bit	Signal	Write Description
15	DMAEN+	1 = Disk DMA enable 0 = Disable Reset clears this bit
14	IDMAR/W-	1 = Disk DMA read operation 0 = Disk DMA write operation
00-13	DC00-DC13	These bits represent the current transfer count
15	U/OERR-	0 = Disk DMA underrun or overrun
00-13	DC00-DC13	These bits represent the current transfer count

Initializing Hard Disk DMA Read

This section lists the signal conditions necessary to initialize system register disk bits and the disk DMA bus interface and to load the WD1010 task file.

Loading Miscellaneous Control Register

The DMA R/W* bit is set by the miscellaneous control register, shown on sheet 15. It is enabled by a signal called MREG WR*, which is asserted by signals shown on sheet 4. The signal conditions that enable MREG WR* are listed in Table 2-21.

Table 2-21 Signal Status to Assert MREG WR*

Sheet	Signal	Status
4	A16	0
4	A17	1
4	A18	0
4	A19	1
4	FWR*	0
4	BGC*	1
4	SUP+	1
4	GATE1*	0
17	PA22	1
17	SPA23	0
17	ENRAS*	0

Loading Disk Control Register

The disk drive and head-select bits are set by the disk control register, shown on sheet 11. It is enabled by HDCTLWR*, which is asserted by signals shown on sheet 4. The signal conditions that enable HDCTLWR* are listed in Table 2-22.

Table 2-22 Signal Status to Assert HDCTLWR*

Sheet	Signal	Status
4	A16	0
4	A17	1
4	A18	1
4	A19	1
4	FWR*	0
4	BGC*	1
4	SUP+	1
4	GATE1*	0
17	PA22	1
17	SPA23	0
17	ENRAS*	0

Loading DMA Address Counter

The status of address bits PA16-PA23 during the loading of the address counters is:

PA23	PA22	PA21	PA20	PA19	PA18	PA17	PA16
0	1	0	0	1	1	0	1

Listed below are the signal transitions to assert the DADD WR* signal, which enables the DMA address counter:

- On sheet 3, PA23 low and BGC+ inactive cause F/F 21E to clock, generating ENRAS*.
- On sheet 17, ENRAS* enables one of four decoder 6J. SPA23 and PA22 are equal to 0 and 1, respectively, causing the decoder to assert GATE1*.
- On sheet 9, the DMA address IC asserts FWR* for fast-cycle writes.
- On sheet 4, gate 25N is enabled because the 68010 is in supervisor mode and BGC* is high. This partially enables gate 26F by putting a low at 26F pin 5.
- GATE1* fully enables 26F, which outputs a low to pin 4 of decoder 28G.
- FWR* enables pin 5 of 28G.
- PA19 is high and fully enables the decoder.
- The select inputs of the decoder are tied to address bits A16, A17, and A18, which are 1, 0, and 1, respectively. This causes the decoder to drive pin 10 low, generating DADD WR*.

DADD WR* inputs to the DMA address IC.

Loading DMA Count Register

The DMA count register, located at address 460000, is loaded when DCNTCS* is asserted. The signal status that asserts DCNTCS* is shown in Table 2-23.

Table 2-23 Signal Status to Assert DCNTCS*

Sheet	Signal	Status
4	A18	High
4	A17	High
4	A16	Low
4	GATE1*	Low
4	SUPV+	High
4	BGC*	High
4	DCNTCS*	Low

During a write, D15 is set to 1 to enable DMA and to a 0 to disable DMA. D14 is IDMAR/W; it is set to 1 for a DMA read and to 0 for a DMA write. Each time D15 goes from 0 to 1, a DMA disk request is generated. Since this is done before a Read/Write Sector command, the data transfer is invalid. If the request is for a DMA write, a transfer occurs to the current address in the DMA address counter. Thus the DMA address must be loaded before the DMA count so that the write goes to a known memory location. Data bits D0-D13 are loaded with the 2's complement of the number of words to be transferred.

Loading WD1010 Task File Registers

These registers must be loaded with information such as sector count, sector number, cylinder number, and command. The processor accesses the task file registers by running a slow bus cycle. The processor asserts chip select to the WD1010 by accessing port E0000X, where X is the address of the target task file register.

The status of address bits PA16-PA23 during the loading of the task file register is:

PA23	PA22	PA21	PA20	PA19	PA18	PA17	PA16
1	1	1	0	0	0	0	0

The signal sequence to assert the chip-select of the WD1010 controller follows:

Sheet 3:

- Gate 27F (pin 12) generates I/ORQ+ since PDS+, SPA23, and INTA* = 1.

Sheet 9:

- I/ORQ+ causes the DMA address IC to assert BGACK* (refer to gate array schematics, sheet 5 of 6).
- I/ORQ+ feeds pin 1 of 1F, which outputs low to 6A. 6A steers F/F 4A to set on the next 1PCK+, generating BGACK*.
- I/ORQ+ also feeds pin 3 of F/F 5B. BGACK+ removes the reset from all of the 5B F/Fs. The slow-cycle timing chain now begins its cycle.
- I/O DTACK+ is asserted by F/F 4B 650 ns after BGACK+.
- Gate 6A (pin 8) has been serving as a latch for BGACK*. When the timing chain has run its course (800 ns after BGACK+), F/F 5B (pin 19) disables 6A and BGACK is dropped at the next 1PCK+.
- Gate 7A (pin 8) is used when writing to the disk controllers, and gate 3B (pin 6) is used when reading from the disk controllers.

Sheet 6 of 6:

- Logic element 8G is the command transceiver. It is enabled by DCS*, which is HDCS* ORed with FDCS*. 8G transfers data between the system bus and the disk data bus when the processor is communicating with the disk controllers.

Sheet 17:

- Decoder 6J is enabled by BGACK* and asserts I/OEN*.

Sheet 4:

- Decoder 27G is enabled because BGC* is inactive, SUPV+ is active, I/OEN* is active, and address bit A21 = 1.
- A16, A17, and A18 are all equal to 0, causing 27G to assert HDCS*.

Sheet 11:

- The WD1010 is being addressed by bits 1-3 from the system address bus. The data is on the DD bus from the DMA data IC.
- All that is required to write data to the addressed task-file register is HDCS* and WR*.

Executing Hard Disk DMA Read

Listed below are signal exchange sequences that occur during a disk DMA Read Sector command. The first sequence begins the cycle, and the rest repeat until a complete sector of data has been read from memory to disk.

Reading Sector Signal Cycles

The cycle begins with a Write-Sector command:

- The 68010 writes the Write-Sector command to the WD1010 controller.
- The controller responds to the Write-Sector command by generating HDBCS* (pin 1), shown on sheet 11. This signal is generated every time the disk controller is ready to receive a byte from the disk DMA bus interface.
- The disk PAL on sheet 2 responds to HDBCS* by asserting TFER* (transfer request). This first transfer request begins the following repeating cycles:

Repeating signal exchange between DMA interface and the arbitration PAL:

- The DMA data array responds to TFER* by asserting DKRQ* (disk bus request), shown on sheet 10. DKRQ* is the request to the bus arbitration PAL for a disk DMA machine cycle. Because the disk DMA bus interface receives 16-bit words from memory and the disk controller receives 8-bit bytes from the DMA data array, DKRQ is asserted once for every two times that TFER* is asserted.
- The arbitration PAL on sheet 2 responds to DKRQ by asserting DKBG*.
- The DMA address array responds to the bus grant by asserting BGACK.
- After 200 ns, the DMA address array negates BGACK.

Repeating signal exchange between disk DMA bus interface and the bus:

- When DKBG* is asserted, the DMA address array loads the DMA address on the bus. Each time the address is loaded on the bus, the address counter in the address array is incremented.
- When BGACK* is asserted, the DMA data array loads DMA data on the bus. A 16-bit word is read into the word buffer in the data array from RAM memory.
- When DKBGA* is negated, both address and data are removed from the bus to complete the machine cycle.

Repeating signal exchange between DMA interface and the disk controller:

- When TFER goes high, the lower half of the 16-bit word stored in the DMA word buffer is transferred by the DMA data array to the disk controller.
- The next time TFER goes high, the upper half of the 16-bit word stored in the DMA data array is transferred to the disk controller. A new bus request is generated by the DMA data array, causing a new word to be written into the word buffer of the DMA data array.

Repeating memory access using arbitration PAL:

- Each time DKBG* is asserted, the arbitration PAL asserts BGC+.
- On sheet 3, BGC+ causes the assertion of ENRAS*, which starts the memory timing sequence.

Controller interrupt ends command execution:

- When the last byte has been written to disk, the controller asserts HDINTRQ, interrupt output pin 3. Sheet 15 shows this signal connected to the line printer status register. If the program is using polling, this register is read to detect command completion. If polling is not used, the interrupt (see sheet 6) generates a level 2 interrupt.

Terminal Count

Terminal count is reached when the DMA count register toggles from FFFF to C000. This means that the programmed number of words has been transferred.

- When counter 7C contains FF and counter 7D contains FF, pin 14 of 7D goes low the next time the counters are clocked. This feeds pin 2 of 4E and pin 13 of 2B. 2PCK+ clocks 2B clear and pin 15 outputs low to the D input of F/F 2B. The following 2PCK+ clocks this F/F clear and TC* is generated.
- TC* is tied to pins 1 and 10 of 1D. This disables 4D and no more disk bus requests are generated.
- The processor is informed of the completion of the disk operation by an interrupt from the disk controller. HDINTRQ+ is generated by the WD1010 upon completion of a command.

If interrupts are being masked by the processor, such as during the bootstrap routine, the interrupts can be detected by polling the line printer status register (bit 2), address 470000.

Underrun/Overrun Errors

An underrun condition occurs when the DMA cannot send bytes to the disk controller fast enough. This is an error condition associated with a disk-write operation. An overrun occurs during a disk read operation when the DMA cannot read bytes from the disk controller fast enough.

The disk controller transfers data to and from a disk spinning at a fixed speed (+ or -1%) and a minimum of 512 bytes must be moved during a Read/Write-Sector command. Thus the disk controller must read or write a byte when it is under the head in the drive. The drive motor cannot speed up or slow down for some other bus master to get off the bus. Thus bus arbitration gives highest priority to disk DMA; otherwise, an underrun/overrun error could occur.

- If the DMA requests the bus while a disk bus grant is still active, a U/OERR is generated.

- The error is detected by the processor by sampling bit 15 of the DMA count register, address 460000.

Separating Hard Disk Data

Data separation is accomplished by a combination of a voltage controlled oscillator, phase detector, and a loop filter. These are called the data separation circuits and they operate as described below.

Modified Frequency Modulation (MFM)

MFM is the method used to encode serial data and clock data written on disk. This method allows the disk controller to distinguish 1s from 0s when the data is read back. Encoding is done by positioning pulses at the beginning or middle of a sequence of 200-ns bit cells, one cell for each binary bit to be written. No more than one pulse can be written in each cell, and certain cells will be empty. A pulse at the beginning of a cell indicates a 0 bit stored in that cell, and a pulse in the middle of the cell indicates a 1 stored in that cell. An empty cell indicates a 0 preceded by a 1. In this last case, the previous cell must contain a pulse in the middle, which ensures that the time between pulses is always equal to or larger than one cell time. The pulse at the beginning of the cell also serves as a clock pulse. When data is read back, a phase locked loop (PLL) is locked to this pulse. The width of the bit cell is fixed by the speed of the disk drive motor, which is 200 ns for the hard disk. The alignment of the bit cell with respect to the data is maintained by the phase-locked loop. An example of MFM coding follows.

200-ns Bit Cells



- An 0 is written at the beginning of a bit cell; 0s also serve as clocks.
- A 1 is written in the center of bit cells.
- A 0 following a 1 is not written because the flux density on the media would be too great. MFM data bits must be separated by at least one bit cell. If a bit cell is empty, the controller knows that it must have been a 0 following a 1.

Data Separation Circuits

The circuits that make up the data separator are:

- Voltage-controlled oscillator (VCO) (IC 14N)
- Phase detector (output at IC 18M, pins 11 and 8)
- Loop filter (output at IC 17N, pin 1)

Voltage-Controlled Oscillator

During a Read Sector command, the VCO is phase locked to the incoming data. Pin 1 is the voltage control pin. When pin 1 is 0 volts, the oscillator output frequency is about 3 MHz. When it is 5 volts, the output frequency is about 18 MHz.

Phase Detector

The phase detector includes the two data flipflops in IC17M, whose outputs are labeled REF and VAR. The REF and VAR F/Fs are used to compare the reference frequency to the variable VCO frequency. NAND gate 18M is connected so that, when REF and VAR are not both set or reset, a pullup or pulldown pulse is generated. At the beginning of a cycle, both REF and VAR are low.

If the frequency of the PLL is too high, the PLL clock goes high at pin 3 of 17M before the delayed reference at pin 11 of 17M goes high. In this case, 18M pins 9 and 10 are both high and a pulldown pulse is generated to pull the frequency back down.

If the frequency of the VCO is too low, pin 3 of 17M goes high after pin 11 of 17M goes high. In this case, pins 12 and 13 of 18M are both high, generating a pullup pulse at 11.

In either case, following pulses set whichever flipflop is not set first, and then reset them both.

Loop Filter

Operational amplifiers (17N) implement a loop filter. The filter is slow in responding to the error pulses produced by the REF and VAR F/Fs and prevents the VCO from reacting to random phase differences. A pullup pulse turns on Q7, which makes the junction of R68 and R67 +5 volts. This makes the output of 17N at pin 7 swing negative at a rate dependent on the value of C253 and R68.

Pin 7 of 17N is connected to the input of an operational amplifier whose output is 17N at pin 1. This operational amplifier is configured as a 2:1 inverting amplifier. Its output is connected to the voltage control pin 1 of the VCO. RP24 adds a positive bias to the output of 17N so that, if pulldown is grounded, pin 1 of the VCO drops to a minimum of 0 volts. If pullup is grounded, pin 1 of the VCO goes to a maximum of +5 volts.

Finding a Valid Address and Data Mark

Beginning the ID field and preceding the data field of a disk sector, there are 14 bytes of 0s. This string of 0s is used by the read circuitry to synchronize the read clock with the read data (remember, 0s double as clocks).

After the 14 bytes of 0s, there is a byte containing A1. This byte is used to validate the 0 string. It is not impossible to have 14 bytes of 0s followed by a byte containing A1 as data. Therefore, to guarantee that the 0s preceding an ID field are indeed marking the beginning of an ID field, a trick is performed on the byte containing A1:

200-ns Bit Cells



This example illustrates the special byte following the 0 string (A1). Normally, a 0 following a 1 is not written; in this case, however, the rule is broken. Notice that in the second instance of a 0 following a 1, the 0 is written. This is the only time that this is true. It guarantees that the preceding 14 bytes of 0s are truly marking an ID field.

The DRUN signal shown on sheet 12 is used to detect strings of 1s and 0s. DRUN is high during a continuous string of either 1s or 0s. It goes low if a data stream that contains both 1s and 0s is present.

The WD1010 checks DRUN during a Read or Write-Sector command. HDRGATE is an output of the WD1010 controller. It is set high when the WD1010 is inspecting data. When HDRGATE is high, the DRUN one-shot input is connected to data from disk. When HDRGATE is low, the DRUN one-shot input is connected to PCK.

These signals are used when the WD1010 is searching for a valid address and data mark. When the WD1010 starts executing a Read-Sector command, it sets HDRGATE low and checks DRUN. DRUN must go high since PCK will look like a continuous string of 0s. The WD1010 sets HDRGATE high and checks DRUN.

If DRUN goes low within five byte times, the WD1010 resets HDRGATE and the sequence starts over. If DRUN remains high for five byte times, the WD1010 starts searching the data for an address mark. If DRUN goes low during this search, the WD1010 deasserts HDRGATE and the sequence starts over.

If an address mark is found, a similar search for a data mark begins. If a data mark is found, the sector is read.

If both address and data marks are not found within eight index pulses, bit 4 in the error register at E00002 is set and the command is aborted.

Write Compensation

The write compensation circuitry is shown on sheet 12. These circuits are used when data is written to disk. They compensate for timing errors that would otherwise occur when data is written to inner sectors.

In the inner sectors, the distance between bit cells is reduced. If not compensated for, the read pulse produced by a bit in one cell reduces the distance between it and the read pulse produced by the bit in the next cell to an unacceptable point. To compensate for this, the time between pulses during write is increased. The signals EARLY* and LATE* control the delay. Associated circuitry is described below:

- The write data is synchronized with PCK+ at latch 16K. It then feeds the D input to F/F 16M.
- The hard disk separator PAL (14M) outputs a signal named MUX on pin 19. During a write operation, MUX has the same phase and frequency as PCK*.
- Since MUX is derived from PCK* in this case, it is a 10-MHz signal. MUX is tied to the clock input (pin 11) F/F 15K.
- This signal inputs to a 50-ns, multitap delay line. Pin 12 is the 10-ns tapoff, pin 4 is the 20-ns tapoff, and pin 10 is the 30-ns tapoff. These outputs are tied to one of four decoders 14K. The 20-ns tapoff also clears F/F 15K.
- The select inputs of 14K are tied to LATE* and EARLY*. These two signals decide how long to delay the data, as dictated by the write precompensation circuitry in the WD1010.

Floppy Disk Direct Memory Access

The Western Digital WD2797 performs the functions of the floppy disk controller/formatter. This device contains a high-performance phase-lock-loop data separator and write precompensation logic. An on-chip VCO and phase comparitor allows adjustable frequency range for 5.25-inch and 8-inch floppy disk interfacing.

Pin Functions (Sheet 13) for the WD2797 are listed in Table 2-24.

Table 2-24 Pin Functions WD2797

Pins	Signal Description
7-14:	D0-D7; eight bidirectional buses used for transfer of commands, status, and data.
5-6:	A0-A1; these inputs select the register to transmit/receive data on the data bus under control of WE-/RE-.
4:	Read-enable; a low on this input controls the placement of data from a selected register on the data bus when CS is low.
2:	Write-enable; a low on this input gates the data on the data bus into the selected register when CS is low.
3:	Chip-select; a low on this input selects the chip and allows computer communication with the device.
22:	Test; a low on this input allows adjustment of external resisters by enabling internal signals to appear on selected pins.
27:	Read data; digitized read-back of the flux reversals on the floppy disk.
32:	Ready; because of strapping, this signal goes low when the door is closed and the drive is selected; it goes high when the door is open and the drive is selected. The 2797 disk controller does not interrupt the processor when the ready signal changes states, so the processor must poll the controller periodically to determine when a floppy disk has been mounted or dismounted.
34:	Track 0; asserted by the selected drive when the head is positioned over track 0.

Table 2-24 Pin Functions WD2797 (Continued)

Pins	Signal Description
35:	Index; asserted by the selected drive when the index hole is passing over the index sensor. This signal pulses every 200 ms and has a nominal duration of 4 ms.
36:	Write-protect; asserted by the selected drive to indicate that a write-protected floppy disk is mounted.
23:	Pump; high-impedance output signal that is forced high or low to increase/decrease the VCO frequency.
26:	VCO; an external capacitor tied to this pin adjusts the VCO center frequency.
15:	Step; pulsed low by the controller to move the head one track in the direction indicated by the direction signal. The 2797 disk controller issues one step pulse every 3 ms. The drive ignores step pulses that would force the head to a negative track number or beyond track 79. The drive also ignores step pulses that occur when write-enable is active or the select line is inactive.
16:	Direction-select; driven low by the controller when stepping to a larger track number; driven high when stepping toward track 0.
25:	Side-select; when driven low by the controller, head 1 is selected; when driven high, head 0 is selected.
29:	Track greater than 43; enables write precompensation for tracks 44-76.
28:	Heads load; controls the loading of the read/write heads against the media.

Table 2-24 Pin Functions WD2797 (Continued)

Pins	Signal Description
30:	Write gate; enables the write circuits on the drive, provided the floppy disk is not write protected.
31:	Write data; MFM or FM output pulse per flux transition. WD contains the unique address marks as well as data and clock in both FM and MFM formats.

WD2797 Registers

The WD2797 has the following internal registers that must be programmed before disk operations:

Table 2-25 WD2797 Registers

Address	Read Information	Write Information
E10000	Status register	Command register
E10002	Track register	Track register
E10004	Sector register	Sector register
E10006	Data register	Data register

WD2797 Commands

The WD2797 floppy disk controller executes a total of 11 commands. It does not have the implied seek function of the hard disk controller, but it does have a Read-Track command that allows examination of an entire track of data for debugging purposes. The commands are listed in Table 2-26.

Table 2-26 WD2797 Commands

Command	Function
Restore	Moves heads to track 0
Seek	Steps heads to track number in track register
Step	Moves heads one step in same direction as last
Step-in	Moves heads in one track
Step-out	Moves heads out one track
Read Sector	Reads one or multiple sectors--aborts if error
Write Sector	Write, same as read
Read Address	Reads first ADX mark and ID field past index
Read Track	Reads track, reports errors--no abort
Write Format	Formats system, provides data for gap and ID
Force Interrupt	Drives interrupt line

Initializing Floppy Disk Write

The following list describes the operations a program must perform to initialize the system before sending a Read Sector command to a disk controller:

Loading System Disk Bits and DMA Interface

- Selects the drive by writing to the disk control register.
- Loads the DMA address registers with the starting logical address. (The address is incremented from low memory to high memory.)
- Loads a 0 to D14 (DMAR/W-) of the miscellaneous register.
- Loads the DMA count register with DMAEN+ = 1, IDMAR/W- = 0, and the 2's complement form of the number of words to be transferred. This causes an invalid transfer of one word and thus requires the reloading of the address register.
- Reloads the DMA address register with the starting logical address.
- Reloads the DMA count register with DMAEN+ = 1, IDMAR/W- = 0, and the 2's complement form of the number of words to be transferred.

The reason for reloading the DMA address register and the DMA count register is that, when the DMA count was initially loaded, the DMAEN+ bit was toggled from low to high. This generates a false disk bus request, and the system does a DMA write to memory. The transfer is false because we have not yet received any bytes from the disk controller. Whatever was latched in the disk buffer was written to memory.

The U/OERR- (bit 15, read) in the DMA count register indicates an underrun or overrun has occurred. With the bus bandwidth and the priority setting of the bus masters, disk DMA should not experience any underrun or overrun error in any bus traffic condition.

Setting Up the Disk Controller

- Loads the controller registers with the desired sector and track number using the disk data bus DD0-DD7.

- Writes the Read-Sector command to the disk controller. Execution of the DMA read begins as soon as the controller receives this command.

The same DMA circuit is being used with the floppy disk as was used with the hard disk. The DMA count register and the DMA address register are loaded in the same manner as discussed earlier.

The major difference between the disk-write operation discussed earlier and a disk-read operation is that, with a read operation, data is transferred from the disk in bytes, assembled into words at the disk buffer, and written to memory.

Executing Floppy Disk Write

Once the DMA has been set up and the WD2797 has been commanded to read, the floppy controller finds the programmed sector and begins to read the MFM data. It then converts the MFM data into binary data and assembles a byte. The byte is placed in the 2797 data register, and a data request is presented to the system.

Sheet 13:

- Read data is input to the floppy controller on pin 27. When a byte has been assembled, the 2797 asserts FDDRQ+ from pin 38.

Sheet 2:

- FDDRQ+ is tied to the D input of F/F 16K. This is an asynchronous signal that is synchronized with the system by 1PCK+ generating FDDRQL+.

- FDDRQL+ inputs to the disk PAL 24H. The PAL generates TFER*.

- FDTFER* starts a timing sequence; QB* is generated on the 1PCK+ following FDTFER*, the next 1PCK+ generates QC*, and the next 1PCK+ generates QD*.

- TFER* goes active at QB* time and generates FDRE*; TFER* and FDRE* remain active until QD* time.

Timing diagram, Figure 2-3, shows the relationship between the preceding signals:

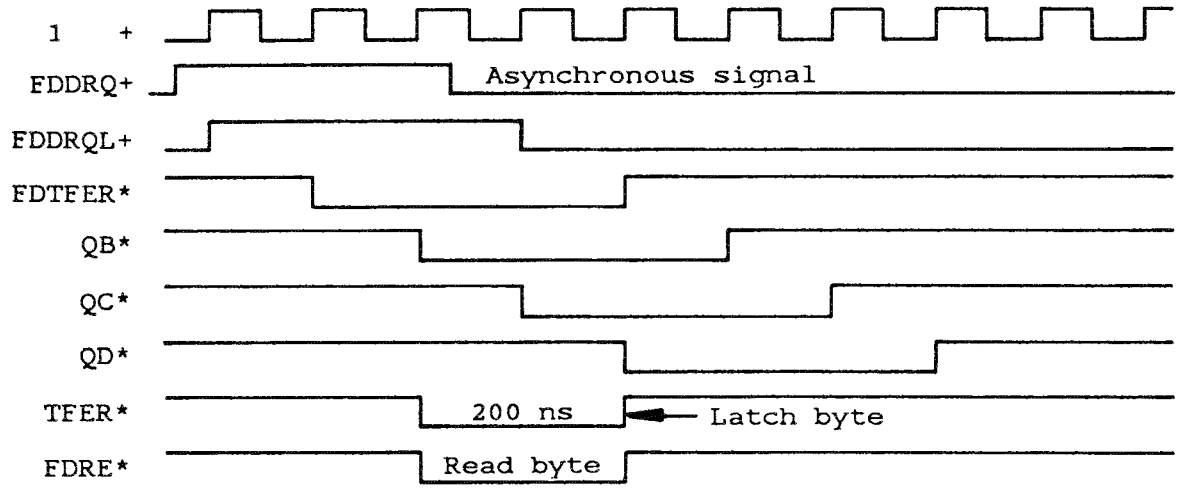


Figure 2-3 Floppy Disk Write Timing Diagram

Sheet 13:

- Chip-select* (pin 3) on the WD2797 is tied low. FDRE* causes the controller to drive the byte onto the DD bus and to disable FDDRQ*.

Now that there is a byte of data on the DD bus, the DMA circuitry must latch the byte in the stacking portion of the data buffer. The DMA waits for another byte and stacks it with the first byte, forming a word. Then a disk bus request is generated. When the bus grant is received, the word is written to a memory address specified by the DMA address counter. (Refer to sheet 6 of 6 in the gate array schematics.)

As mentioned earlier, the first DMA transfer is a false write to memory.

The following sequence describes the generation of DKRQ*:

- Gate 1D pin 10 is high as long as it is not terminal count. Pin 9 is high because F/F 4A is previously clear. Pin 13 is high because we are doing a DMA write. When the DMA is enabled by setting bit 15 in the DMA count register, pin 12 goes from low to high.

- This causes a low to high transition on 4D pin 8, which clocks pin 11 of F/F 4C. The F/F clears and pin 9 goes low.

- The low from pin 9 forces a high from gate 1E pin 3. Pin 3 is tied to the D input (pin 12) of F/F 1C. The trailing edge of the next 2PCK+ clears this F/F, generating DKRQ*.

At this point, the software is reading the DMA count register until it is incremented. This is done to tell when this false transfer has been accomplished. After the software has determined that the count register has incremented, the DMA address register and count register are reloaded. Then the controller is issued a command sequence, and a genuine DMA write operation begins.

- F/F 4A is clear initially (inactive DMAEN+). The Q output is low, partially enabling 4E. TFER* fully enables 4E (pin 13) and pin 11 is driven low.

- TFER* is a 200-ns signal. When it goes inactive, pin 11 of 4E goes from low to high. This latches the byte in the high byte of the buffer and toggles the TFER F/F (4A) set.

The DMA waits for another request (FDDRQ+) from the controller. When the controller requests, the disk PAL again asserts FDRE* and TFER*. FDRE* reads the next byte, which is latched in the low byte of the data buffer when TFER* goes inactive.

- When TFER* goes inactive, the TFER F/F toggles from set to clear. This enables pin 9 of gate 1D (the other pins are still enabled as before), which causes a legitimate DKRQ.

From this point, the operation proceeds the same as the DMA read operation, except that the data is going in the opposite direction.

Video Bit Map

The video circuits appear on two sheets of the schematics. The circuits on sheet 8 contain bit map memory ICs 14-17 and video shift register ICs 14-15. The video IC is shown on sheet 7 and on sheets 2 and 3 of 6. The IC generates timing and control signals, including vertical and horizontal synchronization for the composite video and bit map address and read/write control signals for the bit map memory.

Screen Layout

Video is bit mapped. The screen is divided into thousands of points. Each point is assigned an address and a binary value of 1 or 0 in a 16-bit word. These words are stored in bit map memory. Adjacent points are grouped together so they can be stored in the same 16-bit word.

To create a display, an electron beam scans the screen as shown in Figure 2-4, moving from left to right and top to bottom. The display produced in this way is called a raster scan, and each picture produced is called a field. The bit map memory holds one field; 60 fields per second are produced.

The relationship between bit map address and screen position is shown on the next page. At the beginning of a scan, the beam points to the upper-left corner of the screen. The contents of the video RAM at address 420000 are loaded into a 16-bit shift register. As the beam moves from left to right, the contents of the shift register are shifted out to the monitor. If the current bit is a 1, the beam is on and a point of light appears on the screen. If it is a 0, the beam is turned off and the screen is dark at that point. The horizontal sweep circuit moves the beam to the right until it reaches the right side of the screen. Then the horizontal sync pulse causes the horizontal sweep circuit to swing the beam back to the left side of the screen to start the next horizontal sweep.

At the same time that the horizontal sweep circuit is causing the beam to scan from left to right, the vertical sweep circuit is causing the beam to move from top to bottom. The rate of vertical sweep is such that the beam moves down the screen the width of one line for each horizontal sweep.

When the last bit is displayed in the lower-right corner of the screen, the vertical synchronization pulse and the horizontal synchronization pulse turn on. This causes the beam to return to the upper-left corner to start the next field. In the **UNIX** PC raster, there are 348 horizontal lines, each line containing 720 bits. Each point is called a pixel. Thus, there are 720 x 348 or 250,560 pixels.

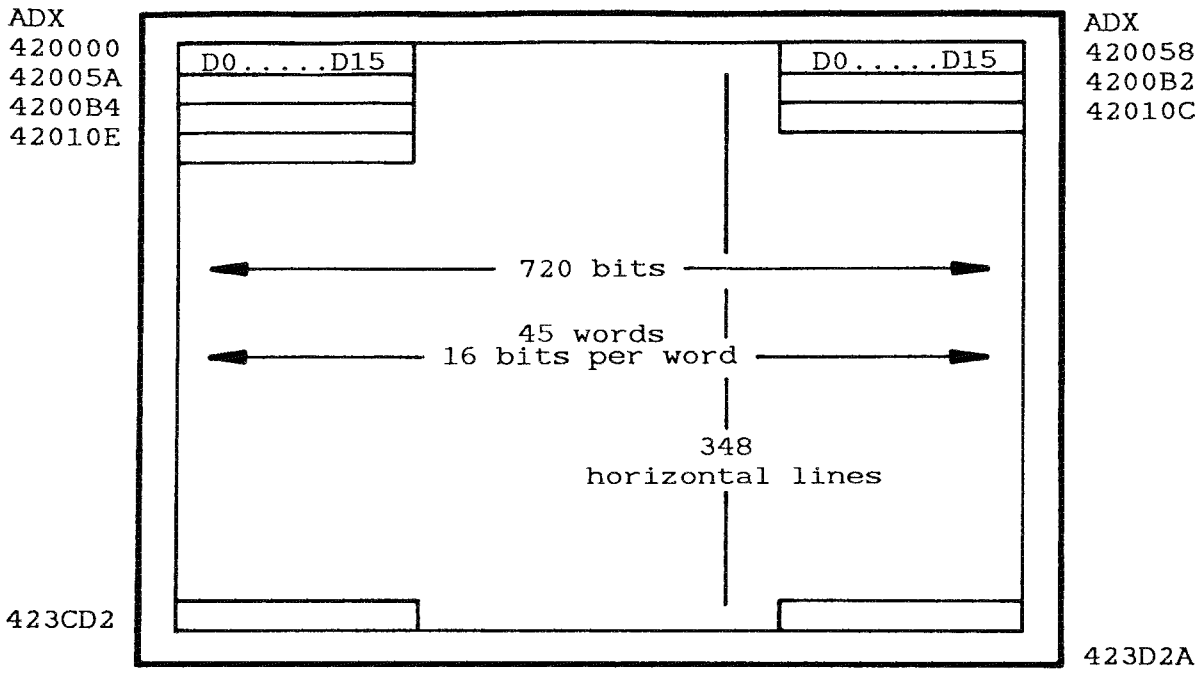


Figure 2-4 Screen Layout

Bit map memory is composed of dynamic RAM. Reading and displaying an address serves a dual purpose of refreshing and displaying the contents of the bit map. The **UNIX** PC allocates 32 KB of system address space for the bit map. This is implemented by a 16K x 16 bit dynamic RAM array. The screen layout above illustrates how the processor must address the bit map. Data written to address 420000 produces the pixels on the first line at the upper-left portion of the screen.

The refresh circuit reads 16-bit words from the bit map, addressing the bit map with an address counter. It then loads the word into a shift register and shifts it out at the speed of a 20-MHz pixel clock. This creates a serial video stream. A 20-MHz clock has a cycle duration of 50 ns, so one word is shifted out every 800 ns. There are 45 words (720 pixels) in each row. The refresh logic reads them consecutively and shifts them out to the deflection board.

After all of the pixels on a line have been displayed, horizontal retrace occurs. This lasts 11 word times (176 pixels) or 8.8 us. Vertical retrace lasts 1079 word times. The beam must be turned off during retrace to prevent diagonal retrace lines from appearing on the screen. If the intensity is turned up too high, retrace lines may appear even though the video signal from the logic board is correct.

State Generator

The state generator circuit is shown on Video IC sheet 1. The state generator produces four state signals: S0-S3, see Figure 2-5. These signals generate timing signals that allow the 68010 to write to the bit map and the refresh address generator to read. The effect is such that a person looking at the screen is not aware of the loading operation. The screen appears to change instantaneously.

The state signals constantly repeat the same sequence of state conditions shown below. Each state lasts 50 ns clocked by the 20-MHz signal. Each state is given a number by assigning binary weight values to the state signals: S0 has a weight of 1, S1 a weight of 2, S2 a weight of 4, and S3 a weight of 8. Thus, each 50-ns period has a state value, calculated by adding weights, that repeats over and over. The period of 0-50 has a state value of 1. The period of 50-100 has a value of 3. Thus, the repeating sequence of state values is 1, 3, 7, 15, 14, 12, 8, 0.

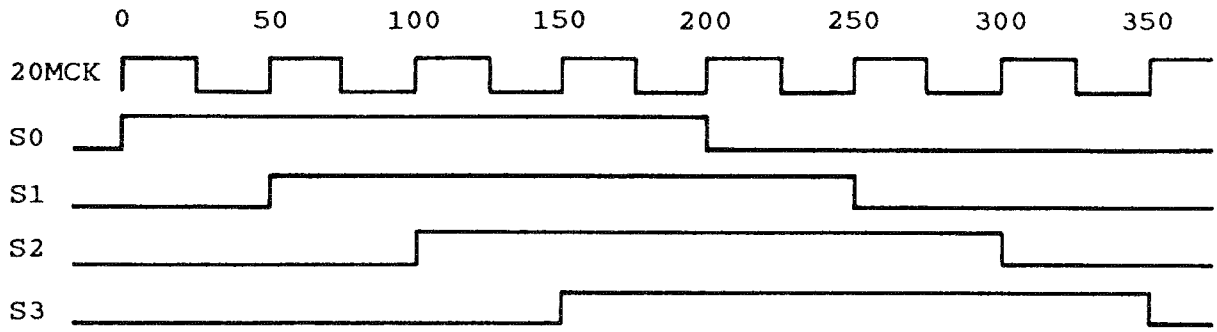


Figure 2-5 State Signal Generator Timing

These states are used to perform the following timing functions:

- Generation of a signal called BMWINDOW, which allows the 68010 and the refresh address counter to write to the bit map by switching the select input to the address multiplexer.
- Generation of BMRAS (bit map row address strobe) and BMCAS (bit map column address strobe) signals to load and refresh 68010 row and column address into the bit map.
- Generation of BMACK, which in turn generates DTACK to tell the 68010 to finish a bit map write machine cycle.
- Generation of SRLD to load 16-bit words from the bit map into the shift register.

68010 Loading of Bit Map

32 KB of system address space is allocated for the video bit map, addresses 420000-427FFF. When the processor needs to alter information on the monitor, it does so by loading data into the video bit map RAM chips. The processor must be in supervisor mode to do this. The following discussion describes a 68010 write operation to the bit map.

First, the signal BMSEL (bit map select) is generated by address decoding. This signal tells the video IC that the 68010 wants to address the bit map. The signals to generate BMSEL are listed in Table 2-27.

Table 2-27 Signal Status to Generate BMSEL

Sheet	Signal	Status
17	SPA23	0
17	PA22	1
17	GATE1*	0
4	PA16	0
4	PA17	1
4	PA18	0
4	PA19	0
4	BGC*	1
4	SUPV	1

On sheet 2 of 6, a signal called CCK is combined with the output of flipflop 2F (pin 6) to generate BMWINDOW*. The F/F 2F (pin 9) is set for one cycle through the states and cleared for the next cycle. It keeps toggling with every cycle, so it generates CCK every other cycle through the states.

CCK switches the video RAM address input between the processor and the raster refresh circuitry. This is done by the output of IC 1F (pin 8). When 1F (pin 8) is low, multiplex chips select address input from the system address bus. 1F (pin 8) is low when CCK* and 2F (pin 6) are high. 2F (pin 6) is clocked high within 800 ns after BMSEL* goes low.

A processor reference to video RAM can be a fast cycle (400 ns), but if CCK is active when BMSEL* arrives, the processor has to wait until the next cycle of states. For the longest possible cycle, it could take the processor 800 ns to reference the bit map.

Gate 1F (pin 8) low switches address multiplex chips IC2J, 2K, 2L, and 2M so that bit map address output BRA0-BRA7 is loaded from address bus A1-A14. BMACK+ is generated at S3 (150 ns) and is used on sheet 2 to assert DTACK* to the 68010 to complete the machine cycle.

BMWINDOW* enables bit map data transceivers 13A and 17B on sheet 8. It also gates the R/W* signal through 26F, creating BMR/W*. The direction of the transceivers is controlled by R/W*. For our discussion, that signal will be low, causing the transceivers to pass the data from the system data bus to the bit map data bus.

On sheet 8 the bit map is set up for a read operation unless the 68010 is performing a write. Gate 26F controls bit map read/write (BMR/W*) at pin 3.

BMRAS* and BMCAS* strobe in the address, and the data is output 120 ns after RAS is asserted.

Bit Map Address Multiplex

The bit map address MUXs are 2J, 2K, 2L, and 2M. They select an address from the system address bus when the 68010 is accessing the bit map, or they select the address from the refresh address counter (1K, 1L) during raster refresh.

The MUX's output row address when S2 is low and column address when S2 is high. In other words, the row address is gated until the 100-ns mark, and then the column address is gated. Tables 2-28 and 2-29 illustrate which address bits are selected and the functions of these bits with respect to the select inputs (A and B) of the bit map address MUXs:

Table 2-28 Multiple Select Status

B	A	Function
0	0	68010 row address
0	1	68010 column address
1	0	Refresh row address
1	1	Refresh column address

Table 2-29 Bit Map Address Multiplex Assignments

B	A	7	6	5	4	3	2	1	0
0	0	A14	A13	A12	A11	A10	A9	A7	A6
0	1	GRND	A6	A5	A4	A3	A2	A1	GRND
1	0	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7
1	1	GRND	BA6	BA5	BA4	BA3	BA2	BA1	GRND

The row address and column address strobe signals are generated as described below on sheet 2 of 6.

Gate 2E (pin 6) enables RAS and CAS. Pin 5 is used during a 68010 reference and pin 4 is used during screen refresh. Gate 1J (pins 3 and 5) is enabled by S1 and/or S2, and it outputs BMRAS* at 50-300 ns. Gate 1J (pins 9 and 13) is enabled by S0 and/or S3, and it outputs BMCAS* at 150-400 ns. Gate 2E (pin 6) enables the 1J gates to generate BMRAS* and BMCAS* when CCK+ is high, as previously described (BMRAS* at 50-300 ns, BMCAS* at 150-400 ns).

On sheet 8, ICs 14A, 15A, 16A, and 17A are 16K x 4, 120-ns dynamic RAM chips that compose the bit map.

BMRAS* strobes in the row address, BMCAS* strobes in column address, and BMR/W* causes the data on the BD bus to be written into memory.

Refresh Address Counter

On sheet 3 of 6, chips 1K and 1L make up a 16-bit binary counter. These logic elements are used as the refresh address counter, which addresses the bit map during a refresh operation. The counter is incremented at the end of each video window when HSYNC* is inactive. HSYNC* is active only during horizontal retrace. The counter is cleared by CLR RFADD+ at the end of vertical retrace.

The counter generates addresses 0000-4163 hex before it is reset. Addresses 0000-3D2A are valid screen addresses. Addresses 3D2C-4163 are generated during vertical retrace and therefore are not used for obtaining pixels.

Horizontal Synchronization

On sheet 3 of 6, logic element 1G, the horizontal counter, generates the timing for horizontal synchronization/retrace. It is clocked by CCK+. It executes a repeating cycle as follows.

Assume that this counter starts counting from 0 and that HSYNC is high. This starts the horizontal retrace. It is clocked by CCK+ when it reaches a count of 10. Gate 3G (pin 8) outputs a low and resets HSYNC low. When the counter reaches a count of 54, IC1H (pin 8) goes low. This sets HSYNC high and resets the counter, thus starting the cycle over. Note that HSYNC inhibits the bit map address counter during horizontal retrace.

Vertical Synchronization

Gate 1N decodes the selected address bit from the BA bus. It outputs low at address 3D2C. This address is driven from the refresh address counter after the last word has been displayed in the lower-right portion of the screen. At this time,1N clears F/F 1A, causing the generation of VSYNC+, which marks the beginning of vertical retrace. F/F 1A also disables DISPEN*.

The refresh address counter is not disabled during vertical retrace. It is incremented 1079 times. The highest address generated is 4163H.

Gate 1H outputs low at refresh address 4163H, marking the end of vertical retrace. VSYNC+ is disabled using F/F 2H (pin 2). This flop also generates CLR RFADD+, which resets the refresh address counter.

Shift Register

On sheet 2 of 6, gate 2D (pin 6) generates SRLD* at 300-350 ns. This signal causes the data output from the video RAMs to be loaded into a shift register, IC14B and 15B on sheet 8. The shift register serializes the data, creating the video stream.

SRLD+ latches the word from the bit map into shift registers 14B and 15B. The data is then shifted out at the speed of 20MCK+, which is sometimes referred to as a pixel clock.

The pixels leave the shift register from pin 17 of 15B. The video stream is synchronized again with 20MCK+ at F/F 18D (pin 2). After the video passes through XOR gate 26B (pin 8), it is routed directly to the CRT deflection board.

Telephony

There are three telephone line connections to the **UNIX** PC. One is for a user telephone, which is referred to as the handset. The other two are active telephone lines, designated L1 and L2. These lines are controlled by circuitry shown on sheet 25.

JPH3 is the handset connector for the user telephone. The **UNIX** PC telephony circuitry supports the following functions:

- Data transmission--through the switched capacitor modem 882A modem shown on sheet 26
- Number dialing--dial network 838A chip on sheet 26; serial dial data generation on sheet 7 (DIALER EN* and DIALER TXD)
- Telephone line management functions and control signals--through circuitry on sheets 25 and 26 listed below:

  Detection of ringing on line 1 or line 2--L1 RING, L2 RING (sheet 25)

  Message waiting detection on line 1--MSG WAIT* (sheet 25)

  Handset offhook detection--OFF HOOK* (sheet 25)

  Line 1 or line 2 on hold--L2 HOLD, L1 HOLD (sheet 25)

  Dial-tone detection on line selected for data transmission--DT DET (sheet 26)

  Handset line select--HDSET RELAY* (sheet 25)

  Modem line select--L1 MODEM, L2 MODEM (sheet 25).

Line Control

Line control circuits shown on sheet 25 are described below. There can be two lines coming in, as well as one handset attached.

Line 1 comes in at location 8C (see guides on margins of schematics). If we follow tip (T/R1) and ring (R/T1), we find that they come into bridge rectifier (CSB1) at 6C. This rectifies the 90VAC ringing signal. The time constant of R35-C240 blocks short pulses. The long ringing time causes opto-isolator 7P to conduct, giving us L1 RING*. This is a status line back to sheet 4/C7, to register 9K, which is available to the CPU as a read-only register at 450000. The opto-isolator prevents potentially hazardous line voltages from coupling into the circuit.

A similar circuit is at 4D, except that there is no R-C time constant, so that MSG WAIT-* appears with a short pulse. This pulse is so short that it is latched into the status register by an extra flop on sheet 4/C6, so that its presence can be observed by the CPU without a prohibitively high polling rate. The CPU looks for a change in status on this line, rather than a high or low.

Sheet 4/3B is the telephony control register. Unlike the status register, which is four bits read at a specific address, this register is treated as eight addresses, 490000-497000. PD14 is written to the appropriate addresses to control telephony.

Writing a 0 to 492000 gives HOOK RELAY 1*, which turns on Q14 at sheet 25/D6, energizing relay K1. This connects the primary of the audio transformer T1 across tip and ring, presenting a 600-ohm load to the CO and drawing loop current. Audio is passed through T1, safety limited by CR5, and then down to analog switch 5M.

The DG201 analog switch is controlled by inputs from sheet 6/5C, a control register at address E4X000. A low at pin 1 (A1) causes a bidirectional link from pin 2 to 3 (IN1 to OUT1). Audio then passes to sheet 26/C8 and into the 838A dial/network. The signal L1 HOLD+ at sheet 25/C1 turns on Q4, supplying a 510-ohm path to ground and making line 1 quiet while on hold.

Most of this circuitry is duplicated for the line 2 interface, with the exception of MSG WAIT*. At 7C, relay K3 determines which line is routed toward the handset, with line 1 being the power-loss default, and K6 at 4B determines whether the handset is attached to the active line. The assemblage at 4C detects current drawn from the line as an indicator that the handset is offhook. If K6 has the handset disconnected from the line, this offhook signal is provided through CR12, at the bottom of K6, which also provides enough voltage so that the handset does not appear dead.

Line control signals are written by the 68010 on sheet 4 by first decoding address bits A16-A23 for status as shown below:

Address Status to Enable Telephone Control Register

23	22	21	20	19	18	17	16	Address bits
0	1	X	X	1	0	0	1	Control register

After status is determined, address bits A12, A13, and A14 are used to address the 74LS138. Processor data bit PD14 then writes a high or low into whichever bit is addressed, as shown in table 2-30.

Table 2-30 Address Status to Select Telephony Control Register

Address Bits				Control Bit Selected
15	14	13	12	
X	0	0	0	HDSET RELAY+
X	0	0	1	LINE SEL2*
X	0	1	0	HOOK RELAY1*
X	0	1	1	HOOK RELAY2*
X	1	0	0	L1 HOLD+
X	1	0	1	L2 HOLD+
X	1	1	0	L1 A-LEAD*
X	1	1	1	L2 A-LEAD*

Table 2-31 shows hex addresses that can be used to write to the telephony control register.

Table 2-31 Telephony Control Register--Address 49X000

Address	Bit	Signal	Description
490000	14	HDSET RELAY+	1 = Handset on line
491000	14	LINE SEL2*	0 = Line 2 select
492000	14	HOOK RELAY1*	0 = Hook relay 1 on
493000	14	HOOK RELAY2*	0 = Hook relay 2 on
494000	14	L1 HOLD+	1 = Line 1 hold
495000	14	L2 HOLD+	1 = Line 2 hold
496000	14	L1 A-LEAD*	0 = Line 1 A-lead connect
497000	14	L2 A-LEAD*	0 = Line 2 A-lead connect

This register is forced with 0s after reset.

Telephony Status Register

Also on sheet 4 is circuitry used by the 68010 to read the status of four line control outputs by loading an address on the system address bus. This enables a buffer that loads the status of these bits on data bus bits D0-D3. The address status to enable this buffer is:

Address to Enable 68010 Line Status Read

23	22	21	20	19	18	17	16	Address bits
0	1	X	X	0	1	0	1	Status

The status bits read and the corresponding processor data bus bits are shown in Table 2-32.

Table 2-32 Telephony Status Register--Address 450000 (Read Only)

Data Bit	Description	Signal
00	OFF HOOK*	0 = Offhook
01	L1 RING*	0 = Line 1 ring
02	L2 RING*	0 = Line 2 ring
03	MSG WAIT*	Toggled each time a message waiting pulse is detected

Dialer Interface

On sheet 4 a control signal called TM/DIAL WR* is decoded to enable circuitry used on sheet 7 to load data from the system address bus and to convert it to a serial data stream for the dial network chip on sheet 26. TM/DIAL WR* is asserted when the address bits have the following status:

Address Bus Status to Enable Serial Dial Data

23	22	21	20	19	18	17	16	Address bits
0	1	X	X	1	0	0	1	Status

On sheet 4 of the Video IC schematic, TM/DIAL WR* enables a three-to eight-line decoder inside the video IC. Address bits A10 and A11 are decoded by the decoder to select one of two functions.

First, if A10 = 1 and A11 = 0, then eight bits of data from the address bus are loaded into a shift register whose output is named DIALER TXD. This is the lower byte.

Then with address bits A10 = 0 and A11 = 1, a load signal loads a second set of eight bits (upper byte) of data from the address bus into a second shift register. This load also starts the shift registers shifting data out at DIALER TXD at the rate of 4800 baud. At the same time, an internal counter is counting. After the sixteenth pulse, the registers are disabled.

Serial Communication

The **UNIX** PC provides an RS-232-C channel that is capable of making either synchronous or asynchronous serial data transfers.

RS-232 Serial Port

Serial data at the RS-232-C port is converted to parallel data by a 7201 dual universal synchronous/asynchronous receiver/transmitter (USART).

RS-232-C Signal Levels

Voltage levels on the RS-232-C interface are +/-12V without load.

RS-232-C Signals

Table 2-33 lists the signals applied to the DB-25, RS-232-C connector. The list gives the pin number, signal name, and direction with respect to the **UNIX** PC.

Table 2-33 RS-232-C Signals

Pin	Name	Direction
1	Ground (shield)	Bidirectional
2	Transmit data	Output
3	Receive data	Input
4	Request to send	Output
5	Clear to send	Input
6	Data set ready	Input
7	Ground	Bidirectional
8	Carrier detect	Input
15	Transmit clock	Input
17	Receive clock	Input
20	Data terminal ready	Output
22	Ring indicator	Input
24	DTE transmit clock	Output

Figure 2-6 shows an example of RS-232-C terminal cable pin assignments.

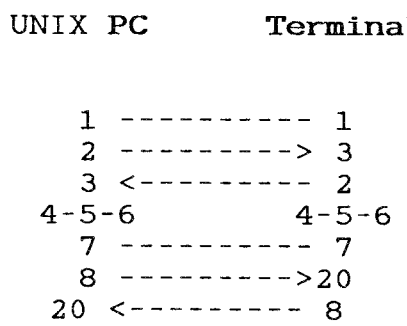
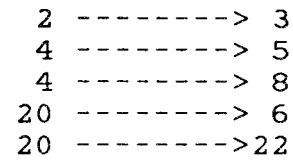


Figure 2-6 UNIX PC to Terminal Cable Pinning

Standalone Diagnostic Loopback Plug

The standalone diagnostics test RS-232-C functions through the use of a diagnostic loopback plug, which must be installed when a channel is being tested. Figure 2-7 shows the loopback plug pin assignments.

Loopback Plug Pinning



(2) Transmit data	---	(3)	Receive data
(4) Request to send	---	(5)	Clear to send
(4) Request to send	---	(8)	Carrier detect
(20) Data terminal ready	---	(6)	Data set ready
(20) Data terminal ready	---	(22)	Ring indicator

Figure 2-7 Loopback Plug Pinning

Baud Rate Generation

The baud rates are generated in the video IC.

The baud rate generation circuitry consists of binary counters 4F and 3D, latch 4L, and comparitors 4G and 4J.

The baud rate is programmable to provide a wide frequency range (307 KHz-1.2 KHz). The baud rate is selected by loading a value into latch 4L. The latch is loaded when the processor refers to address 4B0XXX, where XXX is the value to be loaded (the latch receives its data from the address bus). Once the latch has been programmed, the baud rate generation circuit outputs a signal named TMOUT, the resultant baud rate. The frequency is computed using the following formula (where N is a value between 0 and 255, previously loaded into latch 4L):

$$TMOUT = [1/(4 \times N)] \times 1.2288 \text{ MHz}$$

The input to counter 4F is a 614-KHz clock. This frequency is easily divisible to obtain the range of internally generated baud rates.

Comparitors 4G and 4J compare the variable count from counter 4F with the reference count from latch 4L. When the variable count equals the reference count, 4G outputs a high on pin 6. This high is synchronized with the 1.2288 MHz COMMOSC frequency at F/F 3H (pin 3). When this F/F sets, pin 5 outputs a high that clears counter 4F and clocks F/F 3H (pin 11).

The signal generated from pin 11 is the baud rate.

7201 Serial Controller

The 7201 Multi-Protocol Serial Controller performs all RS-232-C communications control, including serial data in/out, status update, and CPU interrupts. This chip can be used in either interrupt-driven mode or polled mode. **UNIX** PC software uses interrupt-driven RS-232-C communications.

7201 Functions

The controller must be loaded with a set of parameters before actual data transfers are executed. Once programmed with data length, parity generation/detection, number of stop bits, and so on, the chip is capable of transmitting bytes written into its data register by the 68010. During receive operations, the controller assembles characters received from the serial line and places them in its data register so the 68010 can read them.

The controller operates in interrupt-driven mode. When a byte is needed for transmission or when receive serial data has been assembled into a byte, the controller interrupts the processor. Upon receiving the interrupt, the 68010 vectors itself to the appropriate RS-232-C interrupt handler (software).

The 7201 controller is shown on sheet 14. The 68010 transfers data to the controller using the lower eight bits of the data bus, and it addresses control registers in the controller using system address bits A1 and A2.

The chip-enable and interrupt signals are described later in the "Modem" section of this chapter.

The controller features two serial channels: channel A interfaces the RS-232-C port, and channel B interfaces the **UNIX** Pc's internal modem.

7201 Clock Selection

The 7201 controller chip is divided into two separate channels. Each channel has the option of communicating either synchronously or asynchronously. If a channel is programmed for synchronous operation, it must use the clock rate produced by the baud rate generation circuitry. If a channel is programmed for synchronous operation, it can use an external clock provided by the device with which it is communicating. This device is generally a modem.

The TXCKA and RXCKA clocks to channel A are selectable between the TXCK and RXCK clocks, respectively, or the programmable baud rate generator described above.

The output of the baud rate generator is called TMOUT. The select control is controlled by the DTR output at channel B. A 0 selects the RS-232 clock, and a 1 selects the baud rate generator.

The TXCKB and RXCKB clocks to channel B are selectable between the MODEM TXCK and MODEM RXCK clocks, respectively, or a fixed 19.2K-baud generator. The select control is controlled by the MCKSEL bit at the miscellaneous register. A 0 selects the modem clock, and a 1 selects the fixed 19.2K-baud generator.

7201 Registers

Table 2-34 lists addresses that access the internal registers of the 7201 controller:

Table 2-34 7201 Registers

Address	Read	Write
E50000	Ch. A: data read	Ch. A: data write
E50002	Ch. B: data read	Ch. B: data write
E50004	Ch. A: status read	Ch. A: command/parameter
E50006	Ch. B: status read	Ch. B: command/parameter
E68000		Transceiver control 1
E69000		Transceiver control 2
E6A000	Transceiver status	

Modem

The bandwidth available on a common telephone line supports frequencies of 300-3300 Hz. Since the dc levels 0 and 1 cannot be sent on the telephone line, frequencies representing those levels are sent. This method is called frequency shift keying. The frequencies used must be twice the desired data change rate to be deciphered accurately. They also cannot exceed 3300 Hz, the top of the bandwidth window.

At speeds of 1200 baud and higher, this scheme no longer works, because a frequency above 2400 Hz must be used to signal in one direction and insufficient bandwidth is left for signaling in the other direction, at least not simultaneously. So a method called phase shift keying is used, in which the modem sends out a continuous carrier during transmit time. Data is signaled by changes in the phase of this carrier, rather than its presence or absence. The two modems must be in sync with each other, so that a change in phase can be detected. By using high-quality modems, speeds of 9600 can be obtained, but 300, 1200, and 2400 baud are the most popular. On the UNIX PC, 1200 baud and full duplex are used.

RS-232 data transmission is described as being either sync or async. The UNIX PC uses async data transmission. Async data has a start bit and a stop bit associated with every character, whereas sync data has phase sync fields at the beginning and end of each burst of characters.

When a modem auto-answers, it is quiet for two seconds, and then it sends a 2025-Hz answer tone. At this point, the originating modem detects the answer tone and responds with a mark signal, either at 1270 Hz, indicating a 300-baud transmission, or 1200 Hz, indicating a 1200-baud transmission.

At 300 baud, the modem transmits full duplex (both directions simultaneously) by dividing the frequency range into two distinct bands, originate and answer. The modem that originates the call transmits 1070 Hz to indicate a space and 1270 Hz to indicate a mark. The modem that answers the call transmits 2025 Hz to indicate a space and 2225 Hz to indicate a mark.

At 1200 baud, the RS-232 data delivered to the modem is still in async format, but data delivered onto the telephone line by the modem is actually sync format. The originating modem sends carrier at 1200 Hz, whereas the answer modem sends carrier at 2400 Hz. Data is indicated by changes in the phase of the carrier.

Modem Bus

On sheet 26 the processor data bus is connected to the modem data bus. The processor data bus loads control data into the modem before data transmission. When data transmission takes place, the 68010 sends parallel data to the 7201 serial port controller, channel B on sheet 14. The serial port then provides serial data to the modem.

The chip select for the modem (MODEM CS*) and the 7201 are generated on sheet 4. The status of data bus bits to address the modem and the 7201 are:

Address Status for Modem and Serial Controller Enable

23	22	21	20	19	18	17	16	Address bus bits
1	1	1	0	X	1	0	1	7201 chip select
1	1	1	0	X	1	1	0	Modem chip select

Bits A23 and A22 are decoded on sheet 17 to produce I/OEN used on sheet 4.

Modem Clock Select

On sheet 14, a signal called MODEM CK SEL* selects clock signals to send to the 7201 serial controller when the serial controller is being used with the modem. MODEM CK SEL* is generated on sheet 15 using system bus data bit D12 and chip-enable signal MREG WR* generated on sheet 3. MREG WR* is part of the miscellaneous control register. It selects the programmable baud rate generator clocks from the modem when low or a fixed 19.2K-baud clock when high.

Address bus status to assert MREG WR* is:

Address Status to Assert MREG WR*

23	22	21	20	19	18	17	16	Address
1	1	X	X	1	0	1	0	MREG WR*

Modem Registers

Table 2-35 lists the addresses that access the internal registers of the modem:

Table 2-35 Modem Registers

Address	Write
E60000	Line control
E63000	Relay and lamp drivers
E64000	Options A/S and handshake
E65000	Options CCITT and disconnect
E66000	RD, SD control, and chip test

Modem Line Select

On sheet 4 the 68010 writes to the general control register using a control signal called GCRWR* (general control register write) from sheet 4. This signal enables a 74LS259 addressable latch, shown on sheet 6. The latch receives inputs from system address bus bits A12, A13, and A14, which are decoded to select an output. The status of the output is determined by the logic level on 68010 data bus bit PD15. In this way the 68010 can use the status of PD15 to enable the modem line select bits L1 MODEM or L2 MODEM or the dial network control signal D/N CONNECT (dial network connect). These signals listed in Table 2-36 are used on sheet 25.

The status of the system address lines to generate GCRWR* and select a given modem line is:

Address Status to Enable GCRWR*

23	22	21	20	19	18	17	16	Address
1	1	1	X	X	1	0	0	GCRWR*

Table 2-36 Address Status to Select Modem and Dialer Network Control Signals

Address Bits				
15	14	13	12	Control Written to
X	1	0	0	L1 MODEM
X	1	0	1	L2 MODEM
X	1	1	1	D/N CONNECT

Modem/7201 Signals

Channel B of the 7201 is connected to the modem. The following list describes the usage of the signals associated with channel B:

7201 carrier detect <---------- RS-232 ring indicator

7201 receive clock <---------- Modem receive clock/19.2K baud

7201 clear to send <---------- RS-232 data set ready

7201 transmit data ----------> Modem transmit data

7201 transmit clock <---------- Modem transmit clock/19.2K baud

7201 receive data <---------- Modem receive data

In addition to the preceding RS-232 signals, the communications interface also supports the external transmitter clock output, which is pin 24 of the RS-232 connector. This signal is the programmable baud rate generator output.

With the RS-232 DSR (data set ready) and RI (ring indicator) connected to channel B port bits, channel B port bit interrupt should not be used to prevent invalid interrupt being generated by the floating condition of DSR and RI.

Line Printer

There is one parallel printer interface connector on the **UNIX** PC. It is located at the rear of the machine and is controlled by the printer data register.

Ports

The three input/output ports associated with the parallel printer interface are listed in Table 2-37.

Table 2-37 I/O Ports

Port	Description	Read/Write
4F0000	Line printer data register	Write
470000	Line printer status register	Read
4A0000	Miscellaneous control register	Write

Data on data bus bits D0-D7 are strobed into line printer data register bits PRD1-PRD8.

Control Signals

The printer port is shown on sheet 15 of the schematics.

To send data to the printer, this circuitry performs the following three functions:

- Sends eight bits of data from the lower eight bits of the system data bus to the 74LS374 eight-bit data buffer
- Reads the printer status from the 74LS244 eight-bit buffer
- Responds to interrupts from the printer

Address status to decode LPDATAWR* and LPSTATUS RD* is:

Line Printer Status and Data Address Decode

23	22	21	20	19	18	17	16	Address
0	1	X	X	1	1	1	1	LPDATAWR*
0	1	X	X	0	1	1	1	LPSTATUS RD*

The line printer is operated through an eight-bit parallel Centronics interface. The line printer interface is interrupt driven and operates in byte mode transfers only.

Write Sequence

The write sequence that initiates the output to the printer is shown on sheet 15 of the schematics. Timing characteristics are shown in Figure 2-8.

Sheet 15:

- Latch 16P is the line printer data register, address 4F0000. The first step necessary to output to the printer is to write a byte to the data register. This is a normal fast-cycle transfer.
- The next step is to toggle bit 13 in the miscellaneous control register (27B). The processor must write a 1 to bit 13 and then a 0. This strobes the word to the line printer. LPSTROBE must be active for at least 1 us.
- When the printer detects LPSTROBE*, it latches the byte and responds with LPACK*. The trailing edge of LPACK* clocks F/F 25H clear and generates LPINT+.

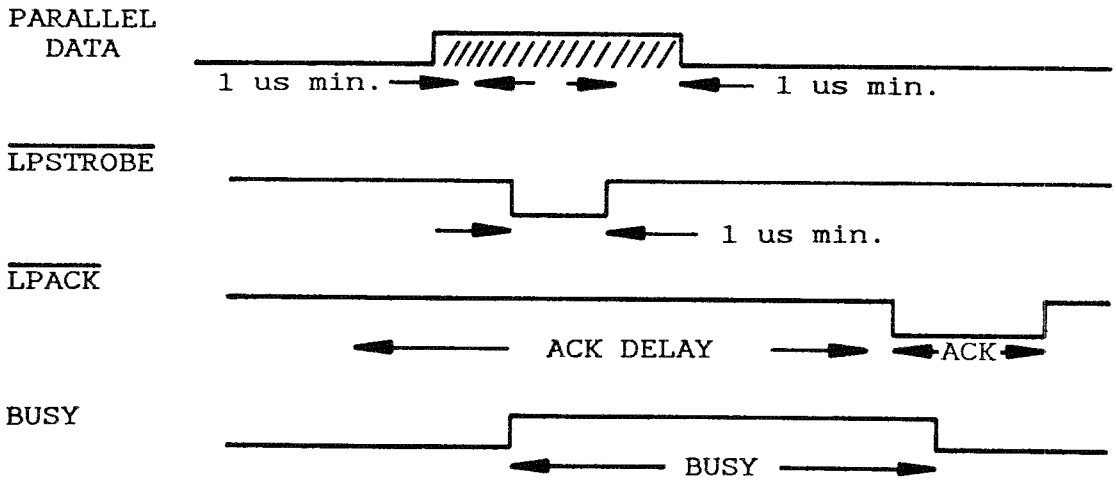


Figure 2-8 Write Sequence Timing

Status Signal Description

Status signals that are sent from the printer to the **UNIX** PC are listed in Table 2-38.

Table 2-38 Line Printer Status Register--Address 470000 (Read Only)

Data Bit	Signal	Description
07	LPBUSY+	1 = Line printer is busy
06	LPSELECT+	1 = Line printer is selected
05	LPNOPAPER+	1 = Line printer has no paper
04	ERROR*	0 = Line printer error condition
03	FDINTRQ+	1 = Floppy disk controller interrupt
02	HDINTRQ+	1 = Hard disk controller interrupt
01	PERR*	0 = Main memory parity error has been detected; this bit can be cleared with the CSR command (write to 4C0000)
00	DTDET*	0 = Dial tone is detected

The 68010 reads the status of the line printer by asserting LPSTATUS RD*.

Serial Printer Cables

The cable pinning for typical serial interface printers is as follows:

Diablo 630 API Cable Pinning

UNIX PC		Printer
1	----------	1
2	----------	3
3	----------	2
4	----------	5
7	----------	7
6-8	--------	20
20	----------	6

UNIX PC to Printer with CTS Control

UNIX PC		Printer
1	----------	1
2	---------->	3
3	<----------	2
4	---------->	4
5	<----------	5
6-8-20		
7	----------	7

Figure 2-9 Printer Cable Pinning

Realtime Clock Interface

The realtime clock integrated circuit is the Toshiba TC8250 with power-down battery backup. The TC8250 interface protocol, as listed in Table 2-39, is implemented with software.

Table 2-39 Realtime Clock Interface--Address 480000

Data Bit	Signal	Description
15	RTCCE+	1 = Realtime clock chip-enable (write-only)
14	RTCALE+	1 = Realtime clock address latch enable (write-only)
13	RTCR/W+	0 = Realtime clock-read 1 = Realtime clock-write (write only)
08-11	RTCD 0-3	Realtime clock data bits 0-3 (write-only)
Address E30000 (Read-Only)		
00-03	RTCD 0-3	Realtime clock data bits 0-3

Keyboard Controller

The 6850 serial keyboard controller is shown on sheet 14 of the schematics. The keyboard controller alerts the 68010 to read data by generating an interrupt KBINT. The keyboard sends serial data to the controller with a signal called KBTXD. The 68010 then reads data from the controller with processor data bits PD8-PD15. The 68010 enables the controller with KBEN generated on sheet 4. On sheet 4, KBEN is derived from a signal called 6850 CS*, which is decoded from the processor address bits as shown below:

Address Status to Assert KBEN

23	22	21	20	19	18	17	16	Address
1	1	1	X	X	1	1	1	GCRWR*

Table 2-40 lists addresses to access the internal registers of the controller:

Table 2-40 Keyboard Controller Addresses

Address	Read	Write
E70000 E70002	Status register Receive data register	Control register Transmit data register

3 Diagnostics

There are two types of diagnostics: ROM diagnostics that consist of a program that is an integral part of the hardware, and a floppy disk program contained on a single floppy disk.

Boot ROM Program

The primary function of the boot ROM program is to boot a program that loads the operating system from the hard disk drive or the diagnostics from the floppy disk drive. In addition, the boot ROM includes a number of diagnostic tests. It tests ROM, RAM memory, and video memory; it also programs the initial status of the memory management hardware and various peripheral controller chips.

During execution, the boot ROM program turns a set of LEDs on and off in appropriate binary number patterns as it completes its tests. The LEDs are visible through the ventilation slots on the left side of the system. (The ROM test executes so fast that the blinking on and off of the LEDs is not be noticed.) The successful completion of each test starts the next. If a test fails, the binary number pattern of that test continues to be displayed as long as the power is on.

Pressing the Reset button or turning on the power causes the boot ROM diagnostics program to execute.

The following descriptions of boot ROM program steps include address and data information for setting up a logic analyzer to trace program execution.

Bootstrap Jump

When the Reset button is released, the initial value of the stack pointer is loaded from addresses 800000 and 800002. The address of the first executable instruction is loaded into the program counter from reset vector locations 800004 and 800006. Then the program:

- Writes a data word with D15 high to address E40000, which sets IC7K (pin 7) high (ROMLMAP*, sheet 6).
- Turns off the LEDs by writing 0F00 to address 4A0000. (The LEDs are controlled by the miscellaneous control register, shown on sheet 15 of the schematics in Chapter 5.)

Initializing the System

To initialize the system, the program writes 0700 to address 4A0000, which sets the LEDs to binary 1. (Note: Data and address are in hexadecimal notation.)

Initializing the 7201 Serial Port Controller

To initialize channels A and B of the 7201 serial port controller, the program:

- Sets LEDs to 1.
- Resets the error registers by writing 18 to addresses E50004 and E50006.
- Writes F0 to addresses E50004 and E50006.

Initializing the Keyboard

To initialize the keyboard, the program:

- Resets the keyboard controller by writing 0300 to E70000.
- Sets the keyboard controller to eight bits per character with one stop bit by writing 9500 to E70000.

Initializing the Modem

To initialize the modem, the program:

- Disconnects the modem from lines 1 and 2 by writing 8000 to E44000 and E45000.
- Resets the modem by writing 0001 and then 0000 to E60000.

Initializing the Telephone Line Control

To initialize the telephone line control, the program:

- Enables the handset by writing 4000 to 490000
- Selects line 1 by writing 4000 to 491000
- Reads the telephone status register at 450000.

If bit D0 is a 0, the handset is onhook, so the program writes 4000 to 492000 and 496000 to open relay 1 and line 1.

If bit D0 is a 1, the handset is offhook, so the program writes 0 to 494000 and 495000, and then 4000 to 493000 and 497000, to maintain line 1.

Clearing the Printer Interrupt

To clear the printer interrupt, the program writes 0000 to 4F0000.

Clearing the Dialer Chip

To clear the dialer chip, the program writes 0 to 4B0400 and 4B0800.

Resetting the Disk DMA

To reset the disk DMA, the program writes 0000 to 460000 and 4D0000.

Testing Video RAM

To test video RAM, the program:

- Sets LEDs to 2
- Writes 0000 to 420000, the lowest video address
- Reads the same address (420000)

If the contents are not the same, the program jumps to an error loop. If the contents are the same, the program writes 0001 to address 420002 and continues to read and test.

- Continues incrementing address and data until the last video memory address (427FFF) has been tested.
- Reads back each address and checks to see that the contents are correct.

After each address is read, it is written again in case writing to it will affect the contents of the next address.

- Writes 0000 to all video addresses to clear the screen.

Testing Map RAM Memory

To test map RAM memory, the program:

- Sets LEDs to 3
- Performs the same tests on map RAM memory that were performed on video memory from addresses 400000 through 4007FF
- Sets the unity map by writing to all map RAMs, starting with 400000 and ending with 4007FF
- Writes A000 to address 400000 to map page 0

This sets page status bits D15, D14, and D13 = to 101, which corresponds to page status of page present, write enabled, not yet written to, and unity-mapped.

- Increments the address and data and writes to pages 1, 2, 3, and so on until all pages have been declared present, write enabled, not written to, and unity mapped.

Testing RAM

To test RAM, the program:

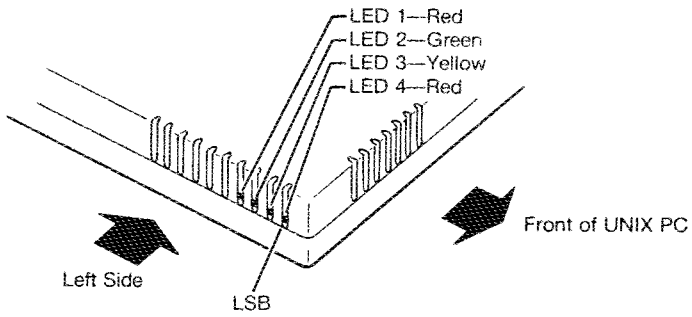
- Sets LEDs to 4.
- Performs the same memory test as before from addresses 000000 through 7FFFFF
- Sets LEDs to 5
- Sets LEDs to 6
- Puts a small inverse video block on the upper-left corner of the screen
- Searches for a loader program

The floppy disk is searched first. If no loader is found or if an error is found, the hard disk is searched. If the hard disk has no loader or has an error (for example, not ready), an inverse video block is added to the screen, and the process is repeated until a loader is found.

Jumping to the Loader Program

When the program finds the loader, it advances the LEDs to 7 and the processor jumps to the loader program.

Figure 3-1 shows the LED locations and off/on patterns for each test.



ROM Tests

	LED Status			
Test Number	**4**	**3**	**2**	**1**
Test 1	Off	Off	Off	On
Test 2	Off	Off	On	Off
Test 3	Off	Off	On	On
Test 4	Off	On	Off	Off
Test 5	Off	On	Off	On
Test 6	Off	On	On	Off
Test 7	Off	On	On	On

Figure 3-1 LED Locations and Off/On Patterns

Listed below is the status of the logic board for each LED test number:

Test 1: Failed telephone initialization
Test 2: Failed video RAM test
Test 3: Failed map RAM test
Test 4: Failed to set map RAM to unity map
Test 5: Failed dynamic RAM test
Test 6: Failed initialization
Test 7: Failed to find loader on disk

Note

Any other failure codes indicate a failure to execute the loader program.

Floppy Disk Diagnostics

The floppy disk diagnostics are in two parts, a Main Menu set which exercises the main components of the system, and a subsystem set.

Main Menu Diagnostics Summary

The following list summarizes how the various diagnostic tests from the UNIX PC Diagnostics Main Menu are divided into subtests or sequences. Tests that require operator evaluation are labeled interactive. The others, which run without user intervention, are labeled automatic. The automatic tests report failures with error messages. To abort any test still in process, with the exception of formatting tests, type "shift-break".

Test 1: Full System Test (Interactive)

This test consists of the following subsystem tests:

Note

These tests are from the Subsystem Menu, selection 6 of the Main Menu. They are run in order as listed.

- Test 2: Floppy disk
- Test 1: Hard disk
- Test 5: Memory and parity
- Test 9: Processor
- Test 11: Real Time Clock
- Test 7: Modem

Test 2: Initialize Hard Disk (Interactive)

- Requests type of drive
- Formats

Test 3: Enter Bad Blocks (Automatic)

- Modifies bad block table
- Displays VHB

Test 4: Park Disk Heads (Interactive)

- Parks disk

Test 5: Remote Diagnostics (Interactive)

- Allows diagnostics to be run from a remote site

Test 6: Goto Subsystem Menu

- These are a series of tests for subsystem checks

Selection 7: Reboot System

- Reboots the system

Diagnostic Subtest Summary

The following list summarizes the diagnostic tests from selection 6 of the Main Menu. These tests are also divided into subtests or sequences. Tests that require operator evaluation are labeled interactive. The others, which run without user intervention, are labeled automatic. The automatic tests report failures with error messages.

Test 1: Hard Disk (Automatic)

- Recalibration of hard disk
- Random seek of hard disk

Test 2: Floppy Disk (Interactive)

- Formats the floppy disk
- Random seek of floppy disk

Test 3: Keyboard (Interactive)

- Determines whether the keyboard and mouse work properly

Test 4: Video (Interactive)

- Tests all screen parameters

Test 5: Memory & Parity (Interactive)

- Performs data test
- Performs address test
- Performs random pattern test

Test 6: Communications (RS232 Ports) (Automatic)

- Self-test of the RS232 Expansion Ports
- Transfer test at 300 Baud through 19200 Baud

Test 7: Modem (Automatic)

- Internal loopback sequence test used to check:
  - 1200 or 300 Baud
  - no parity, odd parity or even parity
  - 7-bit or 8-bit characters

Test 8: Dialer (Interactive)

- Tests touch-tone pulses
- Tests rotary-dial pulses

Test 9: Processor (Automatic)

- Map RAM test:
  - Data
  - Address
  - Random pattern
- Parity test:
  - Read/write
  - Execution test
- Map translation test:
  - Subtest 1
  - Subtest 2
- Page fault test
- User I/O interrupt test:
  - Subtest 1
  - Subtest 2
- Clock test
- Page protection test

Test 10: Parallel Printer (Automatic)

- Subtest 1 (status test)
- Subtest 2 (transfer test)

Test 11: Real Time Clock (Interactive)

- Read Write test
- Operation test

Test 12: Return to Main Menu

- Returns you to the Main Menu diagnostics

Diagnostic Test Descriptions

The following descriptions include test algorithms, error messages, and screen displays for the floppy disk diagnostic tests.

Memory Test

This test has three subtests: data, address, and random pattern. The same algorithm is used to test map RAM, dynamic RAM, and video memory.

Memory Subtest: Data Test

This test writes a walking ones pattern to a memory location and then reads it back. It tests all memory locations. If the data read back is incorrect, it reports the address, data written, and data read back in the following error message:

Memory error at Address X; Wrote X; Read Back X

Memory Subtest: Address Test

The address test has the following three-step algorithm for each address tested:

- Writes data to a memory location.
- Reverses one bit in the address and writes different data.

- Checks the first address to see if there is a change. If data changes, the test generates an error message reporting the address line that has changed and the memory bank being tested. For example, it writes AAAA to addresss 000000 and then writes 5555 to address 000001. It then reads the contents of address 000000. If the contents are not still AAAA, address line A0 is bad. The following error message appears:

Memory error: Connection on address line X is bad at Bank X

Memory Subtest: Random Pattern Test

During this test, a random number generator sequence generates a 64KB random pattern of 16-bit words. This pattern is then written to memory, repeating every 64KB. Then the random number generator is invoked again, and the pattern is read back from memory and compared with the function generator output.

Memory Test Error Message

If the two patterns differ, the address data written and data read are displayed in the following error message:

Memory error at Address X; Wrote X; Read Back X

Memory Test Screen Display

```
MEMORY TEST

      DATA TEST
      Memory test will begin at 2F000, end at 7FFFF
      Testing.....................................
      ADDRESS TEST
      Memory test will begin at 2F000, end at 7FFFF
      RANDOM PATTERN TEST
      Memory test will begin at 2F000, end at 7FFFF
EXPANSION MEMORY TEST
```

Note

The preceding message is displayed if expansion memory is not present. If it is present, the preceding memory tests are repeated on expansion memory.

Parity Test

This test forces bad parity on each location and checks for an error during each memory access.

It first writes 8000 to address E40000 to set EE+ high at IC7K (pin 4) high. Next it writes 8000 to address E41000 to set PIE+ high (IC7K, pin 5). Then memory is written to and read to see if an interrupt results. If a parity error does occur, the following message appears:

Unexpected parity error at location X

Then the BP+ at IC7K (pin 6) is set low by writing 0 to address E42000. This causes bad parity to be written during any access to memory. Next, a selected memory location is written. Then that location is read. A level 7 interrupt results during the read. This interrupt causes the current address and data to be stored in bus status registers BS0 at address 430000 and BS1 at 440000 and the error bit status to be stored in the general status register at 410000.

Parity Test Error Messages

If the interrupt does not occur, the following messages appear:

No Parity Interrupt at location X
BSR incorrect after parity error at location X
BSR0 = X, BSR1 = X

Parity Test Screen Display

Parity Test

```
PARITY TEST - READ/WRITE TEST
Memory test will begin at 2F000, end at 7FFFF
     Reached Address 30000
     Reached Address 40000
     Reached Address 50000
     Reached Address 60000
     Reached Address 70000
PARITY TEST - SUBTEST 2 EXECUTION TEST
```

Map Translation Test

This test performs the following steps:

- Initializes all pages to one-to-one correspondence between logical and physical address space.
- Checks status bits of map registers. First it writes page present to all addresses and then reads them all. The status bits should show all pages present and read. Next it writes to every location in memory and checks to see that every page shows that it has been written to.
- Interchanges physical and logical page addresses and verifies that content has been interchanged.

The map translation test checks the status bits and page table entry swapping. The subtests are:

- Read/write memory test (checks access and dirty bits)
- Map test (checks page table entry swapping)

Subtest 1 (read/write memory test) performs the following steps:

- Sets status register of a page to valid (01)
- Reads from a memory location on that page
- Reads the status of the page and verifies that it has changed to read (10)
- Writes to a memory location on that page
- Reads the status of the page and verifies that it has changed to written to (11).

Subtest 2 checks mapping to the correct physical memory location as follows:

- Writes to two locations on different pages.
- Swaps the page table entries for the two pages.
- Reads back the two locations and verifies that the values are swapped.

Map Translation Test Error Messages

Subtest 1 can return the following error messages:

Page access bit not set for page number
Page access bits wrong: page number and page bits are X
Page mapping error
Page dirty bit not set for page number

Map Translation Test Screen Display

MAP TRANSLATION TEST
MAP TRANSLATION SUBTEST 1
MAP TRANSLATION SUBTEST 2

Page Protection Test

The page protection test has the following three parts:

o CPU page fault

During this portion of the test, a page is declared not present and the processor generates bus errors by reading and writing to it.

The following error messages may occur during this test:

No page fault received on write
Map add = X, Map = X, Mem add = X

No page fault received on read
Map add = X, Map = X, Mem add = X

Bus Error when none expected
Map add = X, Map = X, Mem add = X

o Writing to page, not write enabled, by user

This portion of the test sets the processor to user mode and writes to a memory page that is write-disabled. It accesses the first location of every page except those used by diagnostics.

The following error messages may occur during this test:

No Bus Error received on write, after page write being disabled
Map add = X, Map = X, Mem add = X

o Access below 512 KB by user

This portion of the test sets the 68010 to user mode and writes to every location below 512 KB except that contained by diagnostics. It verifies BSR0 for MMU, BSR0 and BSR1 for faulted address, and RAM for write-disabled.

The following error messages may occur during this test:

No Bus error detected, while user accessing below 512k memory
Disabled Ram writing failed at mem loca X

BSR incorrect after Bus error at location X
BSR0 = , BSR1 =

Page Protection Test Screen Display

PAGE PROTECTION TEST

Format Disk Test

This test has the following three parts:

- Reads volume home block (VHB) and bad block table (BBT)

This portion of the test reads the VHB and BBT and calculates a check sum. If the check sum is correct, it saves the VHB and BBT to rewrite them later. If the check sum is incorrect or the VHB is not present, new VHB and BBT are written after formatting is complete. Whether check sum is correct or not, it continues formatting.

- Formats all sectors

This portion of the test formats the disk one track at a time. After each track is formatted, it reads the status register to see if an error has occurred.

If the status register reports an error, the following message appears:

Error during Disk Format: Response = XX

(XX is the hexadecimal contents of the status register.)

After the disk is formatted, the VHB and BBT are written and read back and a check sum is calculated. If the check sum is incorrect, the following message appears:

VHB write failed. Disk needs to be re-initialized

If the BBT check sum is bad, the following message appears:

Bad block table write failed. Disk needs to be re-initialized

Format Disk Test Screen Display

FORMAT DISK
Formatting cylinder xx

Recalibration Test

This test has the following three parts:

- Seeks to track 0
- Reads status register for error

If an error is present, the following message appears:

Can't Recal: Response = XX

- Reads VHB and BBT and calculates the check sum

If the check sum is incorrect, the following error message appears:

Recal Failed

Recalibration Test Screen Display

Recal Disk

Surface Test

This test writes 6DB6DB6D to all byte locations in one track, 16 sectors. While writing this pattern, it checks the status register for errors.

If an error occurs, the test determines the number of the sector that has the error by writing 6DB6DB6D to one sector at a time. After each sector is written, the status register is checked. When the error is found, the BBT is updated with the number of the sector that generated the error.

If no errors are generated while the track is being written, the test reads the contents of all 16 sectors and checks the data for any errors.

If an error is found, it reads the sectors one at time to determine the one containing the error and updates the BBT. If no error is found, it checks the next track.

If all tracks are written without error, the test reads all tracks one at a time and checks for error. If an error occurs, it reads each sector to find the one that contains the error and adds it to the BBT.

Surface Test Error Messages

Can't Write the new VHB: Response = XX
Can't Write the new Bad Block Table: Response = XX
Error on Write: Response = XX, Start Block = XX
Error on Re-Read: Response = XX, Start Block = XX
Re-Read Data Fail: Start Block = XX, Byte = XX,
Received XX, Expected XX
Error on Check-Read: Response = XX, Start Block = XX
Check-Read Data Fail: Response = XX, Start Block = XX
Initiating Check Read for pass XX
Bad Block Table Overflow when adding Sector XX
Bad Block Table: Multiple use of alternate XX

Surface Test Screen Display

```
                Surface test
Volume Name: FLOPPY
Pass 1
Testing blocks xxx...yyy
```

User I/O Interrupt Test

This test tries to create a bus fault by having a user program attempt to access a register outside its space. If this does not cause a bus fault, the following error message appears:

No bus error when user access I/O address X

User I/O Interrupt Test Screen Display

```
USER I/O INTERRUPT TEST
     USER I/O INTERRUPT SUBTEST 1
     USER I/O INTERRUPT SUBTEST 2
```

Clock Test

This test sets up a clock interrupt so that a digit appears on the screen every second and counts down from 9. The screen shows 9, 8, 7, 6, ..., 1.

The following error message may appear during the clock test:

```
Time out while waiting for 60 Hz interrupt
```

Clock Test Screen Display

```
CLOCK TEST
     Subtest 1 - Timer 1/Counter 2 TEST
          9 8 7 6 5 4 3 2 1 0
```

Printer Test

The printer test has two parts. First the status register is read, and then a barber pole printing pattern is sent tc the printer.

Printer Test Screen Display

```
LINE PRINTER TEST
     LINE PRINTER SUBTEST 1, Status test
Line printer is selected.
     LINE PRINTER SUBTEST 2, Transfer test
Line printer is selected.
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Communications Test

This test does a loopback test and checks combinations of different baud rates, number of bits, and types of parity as listed below:

- Baud rate is 300, 1200, 2400, 4800, 9600, or 19,200
- Bit count is 5, 6, 7, or 8
- Parity is none, odd, or even

Communications Test Error Messages

```
Overrun error
Framing error
Parity and framing error
```

Communications Test Screen Display

```
COMMUNICATION TEST (Self-test & Transfer test)

SELF-TEST
TRANSFER TEST

300 Baud
7 bits/character    no parity
7 bits/character    odd parity
7 bits/character    even parity
8 bits/character    no parity

1200 Baud
7 bits/character    no parity
7 bits/character    odd parity
7 bits/character    even parity
8 bits/character    no parity

2400 Baud
7 bits/character    no parity
7 bits/character    odd parity
7 bits/character    even parity
8 bits/character    no parity

4800 Baud
7 bits/character    no parity
7 bits/character    odd parity
7 bits/character    even parity
8 bits/character    no parity

9600 Baud
7 bits/character    no parity
7 bits/character    odd parity
7 bits/character    even parity
8 bits/character    no parity

19200 Baud
7 bits/character    no parity
7 bits/character    odd parity
7 bits/character    even parity
8 bits/character    no parity
```

After 19,200 baud, the program returns to the main diagnostics menu.

Modem Test

During this test, an internal loopback test sequence checks all possible combinations of the following parameters:

- 300 or 1200 baud
- 7- or 8-bit characters
- No parity, odd parity, or even parity

Modem Test Error Messages

```
SELF TEST failure detected
MODEM TRANSFER TEST, receive error - parity error,
     expected XX
```

Modem Test Screen Display

```
MODEM TEST (Self-test & Transfer test)
     SELF-TEST(B212 mode) at 300 baud..
          Test passed with no error detected.
     SELF-TEST (B212 mode) at 1200 baud..
          Test passed with no error detected.

     DATA TRANSFER TEST at 300 baud..
7 bits/character              no parity
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********

7 bits/character              odd parity
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********

7 bits/character              even parity
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********

8 bits/character              no parity
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
```

```
        8 bits/character                odd parity
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********

        8 bits/character                even parity
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********

             Test passed with no error detected.


DATA TRANSFER TEST at 1200 baud
        7 bits/character                no parity
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********

        7 bits/character                odd parity
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********

        7 bits/character                even parity
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********

        8 bits/character                no parity
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********

        8 bits/character                odd parity
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
        ******** ******** ******** ******** ******** ********
```

```
8 bits/character                    even parity
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
******** ******** ******** ******** ******** ********
```

Test passed with no error detected.

Dialer Test

This interactive test generates either touch tones or rotary dial pulses.

Connect outside telephone lines. Each number is dialed as you enter it, and you can hear it on the speaker. To connect the handset to the telephone line when answering a call, type **c** after the dialed number answers. To disconnect, type **q**.

Note

You must have the handset off hook before you type **c**. You then type **h** to put the line on hold.

Select Test 10 from the floppy disk main diagnostics menu. The following screen appears (DTMF means dual-tone multifrequency):

```
-- INTERACTIVE DTMF TEST --
Select line 1 or line 2 for testing:
```

When you select a line, the following message appears:

Enter digit 0-9;*,# for dialing. To connect the hand set for conversation, enter C/c. To toggle line between hold and active, enter H/h. To quit enter Q/q:

When you select a line and enter a number, you hear the touch tones as the number is transmitted, and the number is displayed on the screen. If you enter a series of numbers faster than they can be transmitted, they are stored and transmitted.

When you type **q**, the preceding message returns. When you select a line and enter numbers for the second time, dial pulses are generated.

Note

Pulse dialing may not work on certain internal electronic telephone systems.

When you type **q** the second time, the following message appears:

```
--      AUTO ANSWER  --
Waiting for incoming call! type Reset/Break to exit!
```

At this point, dial the number of the telephone that is connected to your **UNIX** PC from another telephone.

Video Test

This interactive test displays the following menu of test patterns:

```
VIDEO TEST
----------

 0) All black
 1) All white
 2) Half tone
 3) Vertical bars
 4) Horizontal bars
 5) Mosquito net
 6) Black pattern
 7) White pattern
 8) All m's
 9) Exit

At the end of a test, press any key to get back to this menu.

Please select test number 0 - 9 :(Return is 0)
```

Screen Descriptions

Test 0 makes the screen solid black.
Test 1 makes the screen solid light.
Test 2 displays a fine dot pattern.
Test 3 displays a sequence of vertical bars.
Test 4 displays a sequence of horizontal bars.
Test 5 displays a combination of Tests 4 and 5.
Test 6 displays a cross for checking screen alignment.
Test 7 is the same as Test 7 with inverse video.
Test 8 fills the screen with the letter m.

Trap Errors

Trap errors are general errors that can occur during any test.

Type = XX, GSR = XX, BSR0 = XX, BSR1 = XX
Repeated fault
Bad GSR after page fault
Bad GSR after execution Page fault
BSR not correctly set after Page Fault
BSR0 = XX, should be XX BSR1 = , should be XX
Unexpected bus error, GSR = XX, PC = XX, RPS = X
Unexpected NMI, PC = XX, RPS = XX
GSR = XX BSR0 = XX BSR1 = XX
Unexpected interrupt from level XX
PC = XX RPS = XX GSR = BSR0 = XX BSR1 = XX
HD or DMA interrupt from level XX, PC = XX, RPS = XX
GSR = XX BSR0 = X BSR1 = XX
XX Interrupt level XX, PC = XX, RPS = XX
User I/0 flag in GSR not set during User I/0 test
LWT in GSR set during User I/0 fault
Bus Grant set to DMA cycle during User I/0 fault
BSR not correctly set after User I/0 fault
BSR0 = XX, should be XX BSR1 = XX, should be XX

Expert Mode Diagnostics Program

The expert mode diagnostics program contains more subtests than are available from the floppy disk diagnostics main menu. In addition, it allows selection of specific subtests without running the complete test.

To enter expert mode diagnostics, type **s4test** and press <Return> instead of selecting a test number from the floppy disk diagnostics main menu. To exit from expert mode diagnostics, type: u and press <Return>.

A command> prompt is displayed to show that expert mode diagnostics has been entered.

To see a menu of expert mode diagnostics, type ? and press <Return>. The following menu scrolls on the screen:

```
Commands formatted at follows:
     [<repeat-count>]<command-letter>[L | C]      or
     [<repeat-count>]:<test-number>[,<subtest-number>][L |C]
           P[E | M] toggles Parity Enable | Page Mode
           LE toggles Parallel Printer Echo Mode
           V for Diagnostic Version,     U to return to User Menu
           Multiple commands may be separated by ;
Command    Description
1    Full System test.
2    Initialize Hard disk.
3    Enter Bad blocks.
4    Park Disk Heads.
7    Reboot System.
11    Hard Disk test.
12    Floppy Disk test.
13    Key board & Mouse Test.
14    Video test.
15    Memory & Parity test.
16    Communication test.
17    Modem Test.
18    Dialer test.
19    Processor test.
20    Parallel Printer test.
21    Real time clock test.
31    Interactive device test.
```

Note that entering page mode prevents the top items from rapidly scrolling off the screen. To enter page mode, type **PM** and press <Return> before typing **?**. Press <Page> to see the next page of the menu. Type **PM** and press <Return> again to exit page mode.

Disk Drive Commands

When the **command>** prompt appears, type **i** (Initialize command) and press <Return>. The **disk>** prompt appears. To return to the **command>** prompt, type **q**.

To select the drive that you want to test, type **dr n**, where n is either 0 (default) for the floppy disk or 1 for the hard disk. Then issue the Initialize (i) command or manually turn on the motor, select the drive, and reset the drive.

To include more than one test in a single command, use semicolons. For example:

disk> step 10; rd 1; sb; step - 10; rd 1

To repeat a command, type **;** r at the end of the command. To terminate the repeat option, press any key.

Read and Write Commands

All Read and Write commands operate on the specified sector(s) on the current track. To read from another track, first execute a Seek or a Step command.

Table 3-1 Read/Write Command Format

Command	Short Form	Syntax	Comment
Read	rd	rd [[sector] sector]	
Write	wr	wr [[sector] sector]	
Read ID	ri	ri	FD only
Read Track	rt	rt	FD only
Write Track	wt	wt	FD only

The Read and Write commands use the sector buffer.

The Read ID command uses the ID buffer.

The Read Track and Write Track commands use the track buffer.

Seek and Step Commands

Restore and Seek commands read the ID field at the destination track. The Step command issues step pulses without verifying.

Table 3-2 Seek and Step Command Format

Command	Short Form	Syntax	Comment
Restore	rs	rs	
Recal	rc	rc	
Home	h	h	
Seek	s	s track	
Step	(none)	step number	FD only

Register Commands

Register commands with no argument show the register. Register commands with an argument write to the register.

Table 3-3 Register Command Format

Command	Short Form	Syntax	Comment
Register	rg	rg	
Status	st	st [number]	
Command Register	cr	cr [number]	
Track Register	tr	tr [number]	FD only
Sector Register	sr	sr [number]	FD only
Data Register	dr	dr [number]	FD only
Sector Count Register	sc	sc [number]	HD only
Sector Number Register	sn	sn [number]	HD only
Cylinder Low Register	cl	cl [number]	HD only
Cylinder High Register	ch	ch [number]	HD only
Size Drive Head Register	sdh	sdh [number]	HD only

Buffer Commands

Buffer commands with no arguments show the buffer. Buffer commands with an argument fill the whole buffer with that number. The sector buffer holds 512 bytes, the ID buffer holds 6 bytes, and the track buffer holds approximately 8 KB.

Table 3-4 Buffer Command Format

Command	Short Form	Syntax
Sector Buffer	sb	sb [number]
Track Buffer	tb	tb [number]
ID Buffer	ib	ib [number]

Miscellaneous Commands

These commands are floppy and hard disk control commands.

Table 3-5 Miscellaneous Command Format

Command	Short Form	Syntax	Comment
Drive	dr	dr [number]	0=FD, 1=Internal Hard Disk
Initialize	i	i	
Mtron	(none)	mtron	FD only
Mtroff	(none)	mtroff	FD only
Format	fm	fm track	
Force Interrupt	fi	fi	FD only
Help ?	?		
Radix	(none)	radix [number]	
Quit	q	q	
Head	hd	hd [number]	
Reset	(none)	reset	
Repeat	r	r	

The Mtron (motor on) and Mtroff (motor off) commands control the floppy motor.

The Head command selects a head for further operations.

The Format and Radix commands do not work.

Reading/Writing to Hard Disk with Expert Mode

Use the following sequence of commands and prompts to read and write data to a sector of the hard disk drive using expert mode diagnostics.

1 Load floppy disk diagnostics.

2 When the test select prompt below the diagnostics main menu appears, type **s4test** and press <Return>.

The **command>** prompt appears.

3 Type **i** (the Initialize command) and press <Return>.

The **disk>** prompt appears.

4 Type **dr 1** and press <Return> to select the hard disk.

The **disk>** prompt reappears.

5 Type **i** and press <Return> to initialize the drive.

The LED on the drive lights up.

From this point, use the Sector Buffer (sb), Read Sector (rd), and Write Sector (wr) commands to fill the sector buffer with data, write the contents of the sector buffer to a sector, read the sector into the buffer, and then read the sector buffer contents.

For example, to fill sector 0 with 5555s and then read it back, issue the following sequence of commands after the **disk>** prompt:

1 Type **sb** to read the sector buffer.

The current contents of the sector buffer appear.

2 Type **sb 5555** to fill the sector buffer with 5555s.

3 Type **sb** to verify that the sector buffer contains 5555s.

4 Type **wr 0** to write the contents of the buffer to sector 0.

5 Type **sb 0** to fill the sector buffer with 0s, so you can see that the buffer contents have changed when you read sector 0 from the drive into the buffer.

6 Type **sb** to verify that the sector buffer contains all 0s.

7 Type **rd 0** to read from sector 0 and put the contents into the sector buffer.

8 Type **sb** to read the sector buffer, which now shows 5555s.

4 Logic Board Test Procedures

This section is a collection of test procedures for component-level troubleshooting of the **UNIX** PC logic board. In addition to standard test equipment, the following items are required:

- A set of debugger EPROMs
- A set of map RAM test EPROMs
- A set of RAM test EPROMs
- A set of silent loader EPROMs
- A logic analyzer
- An unintelligent terminal
- A master set of PALs
- A supply of the three semicustom gate arrays

The following reference books will also be useful:

- Motorola's MC68010 16-Bit Virtual Memory Microprocessor
- Motorola's MC68000 Programmer's Reference Manual
- Western Digital's Storage Management Products Handbook
- Intel's Microprocessor and Peripheral Handbook

Information provided with the board usually indicates to an experienced technician the tests that should be used. If the nature of the problem is not known, however, the first step in troubleshooting is to determine the highest level of diagnostics that will run. The higher the level, the larger the portion of the board that must be working and thus can be ruled out as the cause of the problem. Procedures associated with that level are then used to pinpoint the cause of the problem. (Some of these procedures have training value as well as testing value, so it is recommended that they be practiced on a known good board.)

The diagnostic levels are listed below in ascending order:

- Map RAM EPROM
- RAM EPROM
- Silent loader
- Debugger program
- Floppy disk diagnostics

MAP RAM EPROM Program

The map RAM EPROM program executes the following tasks:

- Performs data test
- Performs address test

Data Test

The map RAM EPROM tests the map RAMs and outputs data to the LEDs as listed in Table 4-1, (LED 4 is the one nearest the corner of the board):

Table 4-1 Logic Board LED States

Function	LED 4	LED 3	LED 2	LED 1
Testing data	On	Off	Off	Off
Testing address	On	On	Off	Off
Data error	On	Off	Off	On
Address error	On	On	On	Off

If a data error pattern appears, the EPROM is executing a loop that continuously reads and writes to an address where the data read back differs from the data written. This memory cycle can be observed using a logic analyzer. Set the analyzer to trace from the beginning of this loop. In the EPROM currently being used, this loop is located at 80007A hexadecimal, the address of the first instruction in the loop. The loop contains the following assembly instructions:

```
1$    MOVEW     D1,A0@
      MOVEW     A0@,D0
JMP   1$
```

The MOVEW instruction moves a 16-bit word from internal 68010 register D1 to the memory location whose address is contained in 68010 address register A0. Thus the first instruction is a memory write instruction. The second instruction is similar except that it reads from memory to register D0. The third instruction is a jump that causes the first instruction to repeat. Thus, once this loop is entered, these instructions will repeat as long as the power is on.

When the logic analyzer is set to trace starting at the location of the first instruction, it shows the address and data that were written during the first instruction and read during the second. The data read must differ from the data written for the program to have entered the loop. An example of address and data follows:

Address	Data	Type of Cycle
400000	0000	Data write
400000	0800	Data read

In this example, data bit D11 is a 0 during a write and a 1 during a read. Each static RAM chip contributes 4 bits to the data bus. The bit assignment is as follows:

Data Bits	SRAM Reference Designator
D08 & D13-D15	19C
D12-D09	20C
D04-D07	22C
D00-D03	21C

For the example above where D11 fails, device 20C is the most likely cause. Other causes could be the 74F245 buffers which connect the system data bus (D00-D15) with the static RAM data pins. Also, a bit stuck at 1 or at 0 at the first memory location, address 400000 could be caused by the static RAM and the buffers. A bit stuck at only one address (400020 for instance) and no others, suggests the static RAM and not on the buffer is at fault since its data bit is correct for other addresses.

Address Test

In the address test, the program first writes various data to all memory locations. It then checks all memory to see if there are any locations that do not have proper data. If an address line is faulty, shorted, open, or has some other problem, the program assumes that, at some point in the process of writing to all locations, a write in some location (call it new) changes the contents of a previous location (call it old). When it reads all locations and finds one that is incorrect, the program considers this location to be old. The old location is stored.

The next phase of the test determines the new location. To do this, the program starts a second writing operation to all memory locations. Each time it does a write, the program checks the old location to see if the contents have changed. When the program detects a change in the contents of the old address, it stores the address written to, because this must be new.

Error Loops

The following tables show the addresses of machine code that executes when the map RAM EPROM program finds an address or data error. These serve as trace addresses when a logic analyzer is used to determine the error addresses and data.

Data error (LEDs 1 and 4 on; LEDs 2 and 3 off):

Address	Register Notation	Description
800084:	MOVW D1, (A0)	Writes contents of D1 to address in A0
800086:	MOVW (A0), D0	Reads contents of address in A0
800088:	JMP 800084	Returns to top of loop

Both error addresses found (LEDs 4, 3, and 2 on; LED 1 off):

Address	Register Notation	Description
8000FA:	MOVW (A4), D0	Reads memory address in A4
8000FC:	MOVW D3, (A4)	Writes to memory; A4 holds address that causes contents of memory address in A0 to change when it is written to
8000FE:	MOVW (A0), D0	Reads memory address in A0
800100:	MOVW D7, (A0)	Writes to address in A0
800102:	JMP 8000FA	Returns to top of loop

Second error address not found (LEDs 4, 3, 2 on; LED 1 off):

If an address error occurs but the program cannot find the second error address, the following instruction loop is entered:

```
800110:   MOVW (A0), A0
800112:   JMP 800110
```

Data failures for data bits D10-D15 will be found in the address test instead of the data test as these bits are not fully exercised in the data test.

RAM EPROM Program

The RAM EPROM program executes the following tasks after setting unity map:

- Performs data test
- Performs address test

Data Test

The program performs a walking ones and walking zeros test of 512KB of memory. Each address is written to and then the contents are read back. If the contents read differ from those written, the program jumps to the data error loop.

Address Test

The address test of RAM is similar to the test of map RAM above except that several different data patterns are written. First a data pattern of 0-256 is written to all memory addresses. Then memory is divided into eight 64K blocks. Each block is then tested by dividing it into 256 subblocks of 256 bytes each and writing a different data word to each subblock. As in the map RAM address test, there are two error loops, one when both error addresses have been found and one when the second address cannot be found.

Error Loops

The following addresses are for machine code that executes when the RAM EPROM program finds an address or data error. These serve as trace addresses when a logic analyzer is used to determine the error addresses and data.

Data error loop (LEDs 1 and 4 on; LEDs 2 and 3 off):

```
8000A4:   MOVW D1, (A0)
8000A6:   MOVW (A0), D0
8000A8:   JMP 8000A4
```

Both error addresses found (LEDs 4, 3, 2 on; LED 1 off):

```
8001EE:   MOVB (A4), D0
8001F0:   MOVB D3, (A4)
8001F2:   MOVB (A1), D0
8001F4:   MOVB D7, (A1)
8001F6:   BRA 8001EE
```

Second error address not found (LEDs 4, 3, 2 on; LED 1 off):

```
800202:   MOVB (A1), D0
800204:   BRAL 800202
```

Debugger Program

The debugger is a breakpoint monitor program. With the debugger, the user can modify memory and run programs. That is, the user can enter and run programs, disassemble instructions, and set breakpoints. The debugger includes a load command that allows downloading diagnostics into memory from another computer. This can be used to load diagnostics from a machine that can load diagnostics from a floppy into one that cannot.

To set up the debugger program:

1 Replace the boot ROMs with the debugger ROMs.

2 Connect the **UNIX** PC to an unintelligent terminal using the RS-232-C port.

3 Push the Reset button.

The following message appears:

```
S4      MC68010      ROM      DEBUGGER      V1.0

COPYRIGHT 1984 BY CONVERGENT TECHNOLOGIES INC.

SR=XXXXXXXX PC=00800F66 SP=00800F66 UP=XXXXXXXX
D0=XXXXXXXX D1=XXXXXXXX D2=XXXXXXXX D3=XXXXXXXX
D4=XXXXXXXX D5=XXXXXXXX D6=XXXXXXXX D7=XXXXXXXX
A0=XXXXXXXX A1=XXXXXXXX A2=XXXXXXXX A3=XXXXXXXX
A4=XXXXXXXX A5=XXXXXXXX A6=XXXXXXXX A7=XXXXXXXX
00800F66          bras   0x00800F64
DBG>
```

The program counter (PC) shows the address of the next instruction to be executed. SP and UP are supervisory and user stack pointers, respectively, D0-D7 are data registers, A0-A7 are address registers, and SR is the status register.

After the debugger (DBG>) prompt, type **he** (for the Help command). The following message appears, listing the debugger program commands:

```
COMMANDS:  BR, BC, BO, DB, DF, DI, DM, DR, DW, GO, HE, LO,
           MB, MM, MR, MW, WM, WW, WB, TR
```

These commands are entered after the DBG> prompt is displayed. The first group, the breakpoint group, consists of BR and BC. BR followed by an address sets a breakpoint at that address; BC followed by an address clears the breakpoint at that address.

For example, to set and clear a breakpoint at address 40000:

1 After the DBG> prompt, type **br 40000** and press <Return>.

The following message appears:

```
1 BREAKPOINTS SET AT:  00040000
```

2 Type **bc 40000** and press <Return> to clear the breakpcint.

Using the debugger ROMs, you can set and clear as many breakpoints as necessary depending on your application. If you try to clear a breakpoint at an address where one does not exist, the following message appears:

```
NO BPT AT THAT ADDR
```

When a program containing several breakpoints stops at any given breakpoint, type **go** and press <Return> after the DBG> prompt to continue program execution to the next breakpoint.

The next group of commands, the display group, includes DB, DF, DM, DR, and DW. When followed by an address, command DB (display byte) causes the next 16 bytes of data to be displayed. Pressing <Return> causes 16 more bytes of data to be displayed; typing / and pressing <Return> recalls the DBG> prompt.

For example, to display 16 bytes starting at address 40000:

1 After the DBG> prompt, type **db 40000** and press <Return>.

The following message appears, displaying the 16 bytes:

00040000: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

2 Press <Return> again.

The screen displays the next 16 bytes:

00040010: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

3 Type / and press <Return> to recall the DBG> prompt.

The DF (display all registers) command presents all registers internal to the 68010. To use this command, simply type df and press <Return> to display all registers.

The DM (display 32-bit word) command, when followed by an address, displays the 32-bit word starting at that address. Pressing <Return> gives the next 32-bit word; typing / and pressing <Return> recalls the DBG> prompt.

The DR (display register) command displays each of the 68010 registers just as the DF command does.

The DW (display 16-bit word) command, when followed by an address, displays the 16-bit word located there. Pressing <Return> gives the next 16-bit word; typing / and pressing <Return> recalls the DBG> prompt.

The BO (boot the **UNIX** PC) command first checks to see if there is a bootable floppy inserted. If not, it boots from the hard disk.

The GO command, when followed by an address, causes program execution to begin at that address.

HE is the Help command, which has already been discussed.

LO, the Load command, downloads a program into memory from the RS-232-C port.

The next group of debugger commands, the modification group, includes MB, MM, MR, and MW. With this group, the user can inspect and change the contents of various addresses and registers in the **UNIX** PC.

The MB (modify byte) command, when followed by an address, displays the 16-bit word at that address and enables the user to modify the word. The least significant bit of the address indicates whether the upper or lower half of the 16-bit word can be modified. If it is clear (0), the upper half of the 16-bit word can be modified; if it is set (1), the lower half can be modified. Pressing <Return> allows the next byte to be modified; typing / and pressing <Return> recalls the DBG> prompt.

The MM (modify 32-bit word) command enables the user to modify an entire 32-bit word at a specified address. Pressing <Return> allows the next 32-bit word to be modified; typing / and pressing <Return> recalls the DBG> prompt.

The MR (modify register) command displays each of the 68010 registers so the user can find the one that needs to be modified. It then allows the user to modify that register. Pressing <Return> displays the next register; typing / and pressing <Return> recalls the DBG> prompt.

The MW (modify word) command lets the user modify the 16-bit word at the specified address. Pressing <Return> gives the next word; typing / and pressing <Return> recalls the DBG> prompt.

The write commands, including WM, WW, and WB, enable the user to change the contents of various addresses just as the modify commands do, but the write commands do not display the previous contents. Pressing <Return> lets you write to the next address; typing / and pressing <Return> recalls the DBG> prompt.

The TR (trace trap) command allows single-step execution through the user program. TR followed by an address starts the trace at that address.

Machine Language Programming

Table 4-2 on the next page, contains a few instructions for writing short programs, which can be loaded and run using the debugger program.

Definitions

Dn and An refer to address and data registers inside the 68010, where n is any number from 0 through 7. (An) refers to the contents of memory pointed to by the address in address register An.

Instructions consist of one op code word followed by extension words, if used.

2An in the op code instruction means 2 times the number of the address register. For example, to use the move instruction below (3/2An/8/Dn) to write the contents of data register D2 to the memory location contained in address register A1, the op code is 3282.

Table 4-2 Program Instructions

Machine Code	Register Notation	Description
3/2An/8/Dn	MOVE Dn to (An)	Writes contents of data register Dn to memory location contained in An.
3/2Dn/3/C [ext. word]	MOVE #word to Dn	Loads extension word into data register Dn.
2/2An/7/C [00XX]	MOVEA <ea>, An	Moves the two extension words into address register An. The first extension word contains the most significant address digits.
5/3/4/Dn	Dn = Dn - 1	Subtract Quick. Decrements data register Dn.
B/2Dn/7/C [immed. data]	CMP <ea>, Dn	Compares immediate data to Dn. Subtracts source operand from the specified data register and sets condition codes according to the result; the data register is not changed.

Table 4-2 Program Instructions (Continued)

Machine Code	Register Notation	Description
6/6/X/X	BNE <label>	Branch not equal. If this instruction is preceded by a CMP instruction (above) and the data words compared are not equal, then program execution continues at a location obtained by adding the displacement represented by <label> and the address of the next instruction. The displacement is a 2's complement form.
6/0/X/X	BRA <label>	Program execution continues at the address of the next instruction plus the displacement. The displacement is a 2's complement form.

To calculate the 2's complement displacement for a BNE or BRA instruction (see preceding table):

1 Count the number of bytes back to the opcode in the instruction that is to be branched to.

2 Convert this number to 2's complement by:

- Writing it as an 8-bit binary number
- Changing each 0 to 1 (binary 1's complement)
- Adding 1 (binary 2's complement)
- Converting to hexadecimal

Walking Ones Test Program

The following program sets a pattern of walking ones in the first 1/2KB of memory:

```
070000: 303C   MOVE 1h to D0
070002: 0001

070004: 307C   MOVEA 0100h to A0
070006: 0100

070008: 3100   MOVE D0, -(A0)  (loop)

07000A: E358   ROL D0

07000C: B0FC   CMPA (compare A0 with 0)
07000E: 0000

070010: 66F6   BNE (branch if not equal to loop)

070012: 4EF9   JMP (return to debugger)
070014: 0080
070016: 0EE0
```

Read/Write Loop Program

The use of an oscilloscope in troubleshooting the logic board is limited because most signals, such as address and data, do not have repeating waveforms that an oscilloscope can synchronize on. This problem can be solved with a short machine language program that continuously reads and writes the same address in an endless loop.

The program in Table 4-3 can be loaded into memory using the debugger memory write (MW) command and executed by typing go followed by the starting address of the program. Once the program is executing, an oscilloscope connected to a chip-select pin being addressed by the program synchronizes easily because the chip-select pulse repeats about every 2 micro-seconds. A second oscilloscope channel is used to measure logic levels on address and data lines when the chip-select is active.

To see how this works, execute the following procedure on a known good logic board. In this example, data 0F00, which turns off the LEDs, is written to address 4A0000.

1 Load the program shown below using the MW command.

Register notation and description are given to aid understanding. Enter address and data only.

Table 4-3 Read/Write Loop Program

Machine Code	Register Notation	Description
070200:303C 070202:0F00	MOVE 0F00	Moves 0F00 to data register D0.
070204:207C 070206:004A 070208:0000	MOVEA.1	Loads address register A0. Two most significant address digits 4A. Four least significant address digits 0000.
07020A:3080	MOVE D0 to (A0)	Writes D0 to memory location in A0.
07020C:3210	MOVE (A0) to D1	Reads contents of memory location in A0 and puts it in D1.
07020E:60FA	BRA	Branch always--returns program execution to data write instruction at 07020A.

2 Verify correct loading by typing **di** (the disassemble command) followed by **070200**, the starting address of the program.

The instructions disassemble as shown below. Press <Return> to show the next instruction.

```
070200: #0F00, D0
070204: #004A00, A0
07020A: MOVEW D0, A0@
07020C: MOVEW A0@, D1
07020E: BRA 0x007020A
```

3 Run the program by typing **go 070200** <Return>.

The program executes and the DBG> prompt disappears from the screen.

4 Observe the waveforms by connecting an oscilloscope to IC27B (pin 11).

5 Stop program execution by pushing the Reset button.

The DBG> prompt reappears.

Note that the starting address to load this program is arbitrary. You can load it anywhere you like in memory without changing the program code. The data word in the instruction at 070202 and the address data in the instruction at 070206 are also arbitrary. The first two digits of the address at 070206 are always set to 0, because it takes two words, or eight hex digits, to load an address that uses only six digits; thus, the first two are not used.

Turn On/Turn Off Program

The following program writes data to the same address twice, so that a device can be turned on and then turned off. A delay of 64K machine cycles occurs between turning on and turning off.

```
303C  MOVE # to D0
XXXX  Immediate turn on data

323C  MOVE # to D1
XXXX  Immediate turn off data

343C  MOVE # to D2
XXXX  Immediate delay data

207C  MOVE address to A0
00XX
XXXX

3080  MOVE D0 to (A0)

5342  Decrement D2

B47C  Compare immediate data to D2
XXXX  Immediate data set to 0

66F8  BNE (branch not equal)

3081  MOVE D1 to (A0)

60F2  BRA (branch always)
```

Parity Test Procedure

This procedure verifies the ability of the parity circuits to generate a level 7 interrupt when a memory location containing bad parity is read. In addition, it tests the ability of the bus status registers to store the address at which the error occurred.

First, the starting address of the debugger is loaded into the level 7 interrupt vector location so that, when the interrupt occurs, the program returns to the debugger. Then the error enable EE+ and PIE+ bits are set high to enable the circuits that generate the interrupt. Next a program that writes bad parity to address 000000 is loaded and run. Then address 000000 is read, causing the parity error circuit to generate a parity error. The parity error causes the address of the error, 000000, to be stored in the bus status registers and a level 7 interrupt to be generated. The bus status registers are then read to verify that they were loaded properly.

1 Set up the interrupt vector by writing the following data to level 7 interrupt vector address locations 00007C and 00007E:

```
00007C: 0080
00007E: 0EE0
```

Note

800EE0 is the starting address of the debugger.

2 Enable errors and interrupts by writing the following data to enable the EE+ and PIE+ bits:

```
E40000: 8000   Sets EE+ high; IC7K (pin 4) goes high
E41000: 8000   Sets PIE+ high; IC7K (pin 5) goes high
```

3 Load the following program, which writes bad parity to address 000000 (register notation and comments are included to aid understanding):

```
070300: 303C   MOVE #00008000, D0
070302: 8000

070304: 4241   CLR, D1

070306: 207C   MOVEA.1 #00E42000, A0
070308: 00E4
07030A: 2000
```

Loads address to which bad parity is written in A1:

```
07030C: 227C   MOVEA.1 #000000, A1
07030E: 0000
070310: 0000
```

Sets BP+ high so that bad parity is written:

```
70312: 3080   MOVE D0, A0@
(Writes D0 to memory location in A0)
```

Writes to test address to cause parity error:

```
070314: 3280   MOVE D0, A1@
```

Resets BP+ low:

```
070316: 3081   MOVE D1, A0@
```

Jumps back to debugger:

```
070318: 4EF9   JMP 0x00800EE0
07031A: 0080   Returns to debugger
07031C: 0EE0
```

4 Run the program by typing **go 70300** <Return>.

The program performs the following steps:

- Writes 8000 to address E42000, which sets the BP+ bit at IC7K (pin 6) high.
- Writes to address 000000, which sets bad parity at address 000000.

- Resets the BP+ bit low so that bad parity is not written to any other address after the program returns to the debugger.
- Jumps back to the debugger.

  The debugger message scrolls back on the screen after the program has run.

5 Use the debugger to read address 00000.

This causes the parity error circuit to latch the parity error signal and load the address at which the error occurred into bus status registers 0 and 1. The parity error also produces a level 7 interrupt, which causes the debugger message to scroll on the screen.

6 Read BSR0 430000 and BSR1 440000.

These bus status registers contain FC00 and 00000, showing that address 000000 is the address that caused the error.

Note

BSR0s least two significant digits contain the two most significant address digits. BSR1 contains the four least significant address digits.

7 Check PERR* bit at IC20E (pin 3), which is latched low.

To repeat the test, press the Reset button to clear the latched bits and repeat the preceding steps, omitting step 3.

Page Status Test

This test verifies proper operation of the map RAM page status bits PS0 and PS1. First, the status of page 0 is set to page present, write-enabled, not read. Then page 0 is read and the page status checked to see that it has changed from not read to read. Finally, page 0 is written to and the page status is checked to verify that it has changed to written.

1 Write A000 to address 400000.

This sets status of page 0.

2 Read but do not write to address 0.

3 Read address 400000.

It contains C000, which indicates that page 0 has been read.

4 Write any data to address 0.

5 Read address 400000.

It now contains E000, which means that page 0 is present and written to.

MMUERR* Test

This test procedure checks the memory management unit error (MMUERR*) signal. In summary, it:

- Declares page 0 not present
- Sets EE+ high to enable the MMUERR* signal
- Reads page 0, producing the MMUERR* signal

To test the MMUERR* signal:

1 Connect a oscilloscope to the bus error pin (22) of the 68010.

2 Write 400000: 0000.

This sets the status of page 0 to not present.

3 Write 8000 to address E40000.

This sets EE+, IC7K (pin 4), high.

4 Read page 0 by reading address 000000.

A negative pulse appears on the oscilloscope, indicating that a MMUERR* signal has generated a bus error to the 68010. The debugger program crashes.

The following list summarizes possible page status conditions:

```
400000: 00000   Page 0 not present
400000: 20000   Page 0 present, not write-enabled, not read
400000: A0000   Page 0 present, write-enabled, not read
400000: C0000   Page 0 present, write-enabled, read
400000: E0000   Page 0 present, written, and read
```

Dummy DMA Test Procedure

The dummy DMA test procedure confirms that the disk bus interface logic can write to a RAM memory location and change its contents. Using the debugger program, follow these steps:

1 Set the contents of memory location 000000 to some arbitrary value--for example, FFFF--by typing **mw0** and pressing <Return>.

The system responds with the following address and its current contents:

000000:4EF9

4EF9 is the current contents of location 0. To change the contents, type **FFFF** and press <Return>. The system responds with:

000000:FFFF

2 Escape from the current location and get back to the DBG> prompt by typing a period and pressing <Return>.

3 When the DBG> prompt appears, write 0 to the following locations (note that, because these are write-only registers, they show FFFF each time they are read):

4d0000	Disk DMA address register
4d4000	Disk DMA address register
4a0000	Miscellaneous control register

4 Write 0 to location 460000, the DMA count register (it shows C000); then write 8001 to it (it shows C001).

You have toggled the least significant bit.

5 Examine location 0; it will now be different.

If dummy DMA fails, check the following three signal groups:

- IDMAR/W, DCNTCS, DCNTWR
- DMAREN, MCRWR, DCNTCS
- DMAEN, DKRQ, DKBG, DMA DATA

Floppy Disk Recalibration Procedure

Refer to schematic page 13 for ICs referred to in this section.

This procedure uses the debugger program to manually check the operation of the floppy disk system when the floppy disk drive cannot boot diagnostics.

The procedure verifies the ability of the disk controller to execute the Restore and Seek commands. Data for these commands is shown in two forms: seek with and without verify. When the verify form is used with a formatted floppy disk, the WD2797 disk controller reads the track number from the disk and compares it with the number in its track register, thus verifying the ability of the controller to read data from the floppy disk. The no verify form can be used with an unformatted floppy disk.

1 Adjust the VCO capacitor and RPW potentiometer.

- Check the 1-MHz clock on the disk controller chip (pin 24).
- Reset the floppy disk controller 22M (pin 5) by setting MR (master reset) low and then setting it back high.

To do this, use the debugger program to write the following data to the disk control register:

Address	Data	MR Status (22M, Pin 19, WD2797)
4E0000	0000	Low (resets disk controller)
4E0000	0080	High (clears reset)

- Ground the test pin.

Put a jumper across the two pins marked E4 and E5 next to the disk controller at 22M. This grounds pin 22, the test pin on the WD2797. For CPU boards numbered 60-00225, jumper E4 and E5 does not exist. Use test clipo to connect 22M pin 22 to ground.

- Adjust the frequency.
- Adjust the read pulse width.

Put an oscilloscope on pin 29 (RG43) and adjust R141 for a pulse width of 500 ns.

Put an oscilloscope on pin 16 (DIRC) and adjust the trimmer cap for a frequency of 250 kHz. After this is done, remove the jumper.

2 Verify that you can write to and read back from controller registers.

Using the following addresses, write data to each address and then read it back to check the disk controller's ability to store data in its internal registers. If this does not work, check the data and the read/write lines at the controller.

Register	Address
Track	E10002
Data	E10006

3 Put a scratch floppy disk in the drive, select the drive, and turn on the motor by writing data FFE0 to address 4E0000.

The motor comes on and FDRIVE0, 17H (pin 6), will be high.

4 Issue the Restore command by writing FF0F (with verify) or FF03 (no verify) to address E10000.

If the head is not at track 0 already, it is driven there. After the command is executed, read the track register (E10002), which shows 0 (FF00).

Check that the controller (pin 39) generated an interrupt when the Restore command was completed.

5 Drive head to track 40 (inside track).

- Load hexadecimal 28 (decimal 40) into the data register by writing FF28 to address E10006.
- Issue the Seek command by writing FF13 (no verify) or FF1F (with verify) to address E10000.

During execution, a step pulse occurs at pin 15 each time the head moves one track.

When the Seek command is completed, the track register (E10002) contains 28 hex and a positive pulse appears on pin 39, indicating interrupt at command completion.

Table 4-4 lists the address, data, and WD2797 controller chip pins checked during this procedure:

Table 4-4 WD2797 Pin Listing

Address	Data	Pin	Notes
E10000	FF03	--	Restore command, no verify
E10000	FF0F	--	Restore command, with verify
E10000	FF13	--	Seek command, no verify
E10000	FF1F	--	Seek command, with verify
E10002	XXXX	--	Track register
E10006	XXXX	--	Data register
4E0000	FFE0	--	Data to select drive and motor
4E0000	FF00	--	Data to deselect drive, motor off
		39	Interrupt at end of command
		15	Step pulses during execution of Seek command

3 Put a scratch floppy disk in the drive, select the drive, and turn on the motor by writing data FFE0 to address 4E0000.

The motor comes on and FDRIVE0, 17H (pin 5), is high.

4 Issue the Restore command by writing FF0F (with verify) or FF03 (no verify) to address E10000.

If the head is not at track 0 already, it is driven there. After the command is executed, read the track register (E10002), which shows 0 (FF00).

Check that the controller (pin 39) generated an interrupt when the Restore command was completed.

5 Drive head to track 40 (inside track).

- Load hexadecimal 28 (decimal 40) into the data register by writing FF28 to address E10006
- Issue the Seek command by writing FF13 (no verify) or FF1F (with verify) to address E10000.

During execution, a step pulse occurs at pin 15 each time the head moves one track.

When the Seek command is completed, the track register (E10002) contains 28 hex and a positive pulse appears on pin 39, indicating interrupt at command completion.

Floppy Disk Read Track Procedure

This procedure uses the debugger program to read information from a floppy disk for analyzing problems involving booting from the floppy disk drive.

1 Initialize the system by loading the following data at the addresses shown:

4A0000:	0	Sets DMA R/W* bit low (see sheet 16 of schematics)
4E0000:	0	Resets the floppy disk controller
4E0000:	E0	Selects the floppy disk drive and turns on the motor

This step selects the disk drive and sets up the disk PAL for a floppy disk write to memory from disk. This has to be done only once unless you turn off the power or push the Reset button.

2 Load DMA address and word counters by writing the following data to the addresses shown:

4D0000:	0	
4D4000:	0	Sets starting address at 0
460000:	0	Shows C000 when 0 is written to this address
460000:	8000	

This enables a DMA write to memory. The transfer to memory starts from address 000000 in RAM.

Note

460000 shows C001 after you write 8000 to this address. This indicates that the DMAEN+ bit is set and the first byte has been moved. If you read address 0 at this point, it is different. These counters should be set up before any command that reads or writes information from a track or sector.

3 Issue the Read Track command to read either side 0 by writing E0 or side 1 by writing E2 to address E10000.

This causes track 0, side 0 or side 1, to be moved into the first 8KB of memory. The highest address that contains data from the track is about 187A (hex). The disk controller starts reading from the first index pulse after the command is issued and continues until the next index pulse.

IBM System 34 Format

Number of Bytes	Hex Value of Byte Written
80	4E
12	00
3	F6 (writes C2)
1	FC (index mark)
50	4E
----*	
12	00
3	F5 (writes A1)
1	FE (ID address mark)
1	Track number (0-4C)
1	Side number (0 or 1)
1	Sector number (1-1A)
1	01 (sector length)
1	F7 (2 CRCs written)
22	4E
12	00
3	F5 (writes A1)
1	FB (data address mark)
256	DATA
1	F7 (2 CRCs written)
54	4E
----*	
598	4E

* Bracketed field is written 26 times.

4 Examine memory (see preceding table) starting at address 0 and look for the pattern of 80 4Es and 12 00s that precedes the index address mark of FC.

Note

The 4E pattern may look like 27, 96, or 39 at first, because the disk controller has not yet synchronized its data separator, which tells it which bit a byte starts with.

6 Next find the ID fields for the first sector. Look for an A1A1FB pattern.

The next byte after FB is the first byte of the six bytes in the ID field. The contents of the fields are:

Byte 1: Track address 0-40 (decimal)
Byte 2: Side number 0 or 1
Byte 3: Sector address 1-26 (decimal)
Byte 4: Sector length code
Byte 5: CRC
Byte 6: CRC

Seek Command

If you want to modify the preceding procedure to read a different track, simply load the data register with the track you want and issue the Seek command before issuing the Read Track command. Note that there are two forms of the Seek command: with and without verify. With verify, the command reads the track information from the disk and compares it with its internal track register. The maximum track number is 28 hex.

Read Address Command

If you want only to read the ID field, you can substitute the Read Address command for the Read Track command. With this command, you can read the first ID field after the index pulse. Because this command writes only six bytes, you can repeat the command without reloading the DMA counters. The data is six bytes higher in memory because the address counter advances with each execution.

Summary of addresses and data used in the foregoing procedure:

4A0000:	0	Sets DMA R/W*
4E0000:	0	Resets controller
4E0000:	E0	Selects controller and turns on motor
4D0000:	0	DMA address counter
4D4000:	0	DMA address counter
460000:	8000	DMA word counter, DMAEN+, IDMAR/W*
E10002:		Track register
E10004:		Sector register
E10006:		Data register

Summary of commands used to write to the floppy disk controller command register at address E10000:

1B:	Seek, no verify
1F:	Seek, with verify
E4:	Read Track side 0
E6:	Read Track side 1
C4:	Read Address side 0
C6:	Read Address side 1
8C:	Read Sector side 0
8E:	Read Sector side 1
AC:	Write Sector side 0
AE:	Write Sector side 1

Guidelines for using commands:

All commands:	Set DMA R/W*, reset controller, and select drive and head before executing first command and after any reset.
Seek:	Load data register with desired track number first.
Read Track:	Set up DMA counters first.
Read Address:	Set up DMA counters first.
Read/Write	Sector:Set up DMA counters and load sector number into sector register first.

Sector numbers start at 1. Tracks are 0-40 decimal or 28 hex.

Floppy Write Sector Procedure

This procedure writes data from the lowest 1/2 KB of memory to sector 1, track 0, side 0 of the floppy disk. Enter the following data at the addresses shown:

4A0000:	4000	Sets DMAR/W- high to read memory and write to disk
400000:	0	Resets the floppy disk controller
4E0000:	E0	Selects floppy drive 0 and turns on motor
4D0000:	0	A1-A8 correspond to least significant 8 bits of DMA address
4D4000:	0	Sets starting address to 0; A1-A13 correspond to most significant 13 bits of DMA address
460000:	0	Shows C000
460000:	C000	Sets DMAEN D15 and IDMAR/W high; initializes DMA word count to 0 E10000: AC. Writes sector side 0

Initialize Floppy Program

This machine language program initializes the DMA bus interface and floppy disk controller to read sector 1, track 0, side 0 from a formatted floppy disk. To use the program, perform the following steps:

1 Load the program as shown below. Verify that it is entered correctly using the debugger DI command.

Note

The program is stored in video memory so that it is not overwritten by disk DMA.

```
420008:   33FC
42000A:   0000 Resets disk drive
42000C:   004E
42000E:   0000

420010:   33FC
420012:   00E0 Selects disk drive
420014:   004E
420016:   0000

420018:   33FC
42001A:   0000 Sets DMA address least significant
               bits to 0

42001C:   004D Uses address bits A1-A8
42001E:   0000

420020:   33FC
20022:    0000 Sets starting address most
               significant bits to 0

420024:   004D Uses address bits A1-A13
420026:   4000

420038:   4EF9
42003A:   0080 Returns to debugger
42003C:   0EE0
```

Note

This program starts storing data from sector 1 at RAM address 000000. You can change the starting address by changing the address data shown at 42026. For example, to change the starting address to 70000, change data for address 420026 to 4700. The third digit of this word is the fifth digit of the DMA address.

2 Execute the program by typing go 420000.

The debugger message scrolls on the screen.

3 Read sector 1 by writing 8C (the Read Sector command) to E10000.

4 Examine memory to see if the command has been executed.

Note

If you issue the Read Sector command a second time, the sector is read again but it is stored in the next 1/2 KB word of memory. You can fill as much memory as you like by repeating the Read Sector command. Each time the Read Sector command is repeated, the DMA word counter at 460000 increments by 1/4 KB.

DMA Looping Program

The following machine language program uses a loop instruction to test the DMA bus interface. It produces a continuous disk bus request, grant, and acknowledge waveform. The program fills the 2 MB of memory that can be accessed by DMA with the same data.

```
420100:   33FC
420102:   0000
420104:   004A          Sets DMA R/W bit low to write
                        to memory
420106:   0000

420108:   33FC          Sets DMA enable low
42010A:   0000
42010C:   0046
42010E:   0000

420110:   33FC          Toggles DMA enable high
420112:   8000
420114:   0046
420116:   0000
420038:   60EE          Branch always
```

Hard Disk Registers

This section provides general information that is to be used in the procedures involving the hard disk drive. Addresses 4A0000 and 4E0000 are the addresses of the disk control and miscellaneous control registers. These registers must be set up before any command is written to the disk controller.

Summary of addresses and data written to the disk control and miscellaneous registers:

4A0000:	0000	Sets bit 14 low for DMA read disk and write to memory
4A0000:	8000	Sets bit 14 high for DMA read memory and write to disk (see sheet 15 of schematics for circuits)
4E0000:	FF00	Resets hard disk controller (see sheet 11)
4E0000:	FF1F	Selects drive 0, head 0 (see sheet 11) and clears reset

To set the DMA address counter to start a DMA transfer at memory address 0, write 0 to addresses 4D0000 and 4D4000. Do this before each Read or Write Sector command.

To set the DMA word counter and enable DMA write, write the following data to the address shown:

460000:	0	Sets enable low
460000:	8000	Sets DMAEN high and selects DMA write

To initialize the WD1010 internal registers, write the following data to the addresses shown:

E00004:	01	Sector count (number of sectors to be transferred)
E00006:	0000	Sector number (sector to be transferred)
E00008:	FF64	Cylinder, least significant 8 bits
E0000A:	FF02	Cylinder high (only bits 0 and 1 are used)
E0000C:	20	Size drive head (SDH), head 0, drive 1, 512 bytes/sector

WD1010 Command Register (E0000E)

Table 4-5 lists the WD1010 command register bit definitions:

Table 4-5 Bit Number

Command	7	6	5	4	3	2	1	0
Restore	0	0	0	1	R3	R2	R1	R0
Seek	0	1	1	1	R3	R2	R1	R0
Read Sector	0	0	1	0	I	M	0	T
Write Sector	0	0	1	1	0	M	0	T
Scan ID	0	1	0	0	0	0	0	T
Write Format	0	1	0	1	0	0	0	0

R3-R0 = Step rate for head movement
F = 7.5 ms
0 = 35 us

I = Interrupt enable
I = 0 Interrupts at BDRQ time
I = 1 Interrupts at end of command

M = Multiple sector flag
M = 0 Transfers one sector
M = 1 Transfers multiple sectors

T = 0 Enables retries
T = 1 Disables retries

Commands in Hexadecimal

The left or most significant digit selects the command. The right or least significant digit selects the options described in the command register table above and in the following tables.

Restore = 1X, where X is a value from 0 through F depending on step rate
Seek = 7X

Read Sector = 2X

Table 4-6 Read Sector Commands

Read Sector	I	M	T
20	0	0	0
21	0	0	1
24	0	1	0
25	0	1	1
28	1	0	0
29	1	0	1
2C	1	1	0
2D	1	1	1

Requirements to successfully complete the Read Sector command:

- Ready input must be asserted from the disk drive.
- Seek complete must be asserted at the end of a Seek command. A rising edge on this input informs the WD1010 when head settling time has expired.
- Cylinder and head numbers must match.
- No CRC error or bad block must be detected.
- Data address mark must be found.

Write Sector = 3X

Table 4-7 Write Sector Commands

Write Sector	M	T
30	0	0
31	0	1
34	1	0
35	1	1

Scan ID = 40 or 41, where 40 enables retries up to 10 revolutions. 41 aborts after 2 revolutions of drive.

Table 4-8 SDH Byte Register (E0000C, Read/Write)

SDH Byte Register

Bit Number			Head Selected
2	1	0	
0	0	0	Head 0
0	0	1	Head 1
0	1	0	Head 2
0	1	1	Head 3
1	0	0	Head 4
1	0	1	Head 5
1	1	0	Head 6
1	1	1	Head 7

Bit Number			Drive Selected
4	3		
0	0		Drive 1
0	1		Drive 2
1	0		Drive 3
1	1		Drive 4

Bit Number			Size Selected
6	5		
0	0		256 bytes/sector
0	1		512 bytes/sector
1	0		1024 bytes/sector
1	1		128 bytes/sector

Bit 7, the extension bit, extends the data field by seven bytes when ECC codes are used. CRC is not appended to the end of the data field. The SDH byte is written into the ID field during format.

Write Precompensation Register (E00002, Write Only)

The value 00-FF loaded into this register is internally multiplied by 4 to specify the cylinder where RWC is asserted. A value of FF always causes RWC to be low, no matter what cylinder number is used.

Error Register (E00002, Read Only)

Bit 7: Bad block detect, used for bad sector mapping.
Bit 6: CRC error.
Bit 5: Forced to 0.
Bit 4: ID not found. This bit is set to indicate that the correct cylinder, head, sector number, or size parameter could not be found or that CRC error occurred on the ID field. This bit is set on the first failure and remains set even if the error is recovered on a retry. When recovery is unsuccessful, the error status bit is set also.
Bit 3: Forced to 0.
Bit 2: Aborted command. This bit is set if the command is issued while the DRDY is deasserted or the WF is asserted. The aborted command bit is also set if an undefined command code is written into the command register.
Bit 1: Track 0 error. This bit is set only by the Restore command. It indicates that TK000 has not gone active after the issuance of 1KB of stepping pulses.
Bit 0: Data address mark not found. This bit is set during a Read Sector command if the data address mark is not found after the proper sector ID is read.

Status Register (E0000E, Read Only)

Bit 7: Busy. This bit is set whenever the WD1010 is accessing the disk. Commands should not be loaded into the command register while the busy is set.
Bit 6: Ready.
Bit 5: Write fault.
Bit 4: Seek complete.
Bit 3: Data request.
Bit 2: Reserved, forced to 0.
Bit 1: Command in progress.
Bit 0: Error. This bit indicates that a nonrecoverable error has occurred. If the host reads the status and finds this bit set, it must read the error register (E00002) to determine the type of error written into the error register.

Hard Disk Recalibration Procedure

The following procedure uses the debugger program to verify the ability of the hard disk controller to execute the Restore and Seek commands and the ability of the data separator to produce a proper clock and data stream.

1 Write the following data in the sequence listed using the debugger program:

```
Disk control register, reset:          4E0000: FF00
Disk control register, select drive:   4E0000: FF1F
```

2 After writing this data, verify the following signal conditions:

- o At chip 17K (pin 9), a logic low indicates that drive 0 has been selected.
- o At chip 17K (pin 3,5 & 7), a logic low indicates that hard disk drive head 0 has been selected.
- o At chip 18K (pin 5), index pulses are present.
- o At chip 18K (pin 11), a logic low indicates that the drive is ready.
- o At chip 16M (pin 9), separated data from the data separator is present.
- o At chip 13K (pin 3), data from the hard disk drive is present.

3 Issue the Restore command by writing FF1F to address E0000E.

This command causes the stepping motor on the drive to bring the head back to cylinder 0 if it is not already there.

After this data is written to the WD1010 command register, the data you read back on the screen is FF50. This is the contents of the disk controller's status register, and it indicates that D6, the ready bit, and D4, the seek complete bit, are both high. When D6 is high, it indicates that the disk controller is receiving a logic high on its RDY pin 28. When D4 is high, it indicates that a seek initiated by the last command has been completed successfully.

4 Write the following reset and drive select data to the disk-control register to reset the controller:

```
Disk control register, reset:          4E0000: FF00
Disk control register, select drive:   4E0000: FF19 (head 1)
```

5 Load the cylinder register as shown below (these are read/write registers, so you should read back the same data you write):

Cylinder number low (614): E00008: FF64

This register holds the eight least significant bits of the cylinder number.

Cylinder number high (614): E0000A: FF02

Bits 0 and 1 are the two most significant bits of the cylinder number.

6 Issue the Seek command, which causes the stepping motor to rotate until the head is at cylinder 614.

Seek command: E0000E: FF7F

Procedures for Reading/Writing Data to Any Sector

Using the debugger program, the next two procedures make it possible to write any desired data to any selected sector on the hard disk and read it back to verify that it was written properly. These procedures use the Read Sector, Write Sector, Seek, and Scan ID commands of the WD1010 controller. The error register bits generated by the WD1010 are listed as well as the signals to check when a board failure prevents these procedures from executing properly.

Read Data to Any Sector

1 Set the DMA address counter:

```
4D0000: 0
4D4000: 0
```

2 Set DMA write, select the drive, and restore the head to track 0:

```
4A0000: 0    Sets bit 14 low, DMA write to memory from disk
4E0000: 0    Resets the disk controller
4E0000: 19   Selects drive 0, head 1
E0000E: 1F   Restore command, returns the head to track 0
```

3 Set the DMA word counter and toggle DMA enable:

```
460000: 0       Reads C000
460000: 8000    Reads C001
```

4 Clear memory by putting 0s in the first 16 words of memory at addresses 0h-2h and 1F0-1FF.

5 Load the hard disk controller registers by writing the following data to the addresses shown:

E00002:	0	Write compensation register
E00004:	1	Sector count register (selects the number of sectors to read)
E00006:	0	Sector number register (selects the sector to read)
E00008:	FF	Least significant byte of cylinder number (cylinder 255 selected as an example)
E0000A:	0	D0 and D1 are the most significant bits of the cylinder number

6 Issue the Seek command by writing 7F to address E0000E.

You read back D2 and the stepping motor rotates. D2 indicates that the command has executed with no errors.

7 Issue the Scan ID command to update the SHD register by writing 41 to address E0000E.

You read back D2, indicating successful completion.

8 Check the SDH register by reading address E0000C.

It contains FF21, which indicates that 512 bytes per sector and head 1 have been selected.

9 Write 28 (Read Sector command) to address E0000E.

It reads back D2. Check the contents of memory to see that the data has been moved from disk. If you do not read D2 after issuing this command, read E00002, the error register, and determine the error. If you write 28 to E0000E again, the command is executed again, but the data is transferred to the next 1/2 KB of memory from address 000200 to 000300. Each time the command is repeated, the DMA count register at 460000 shows another 1/2 KB of data moved. After the first execution, it shows C101; after the second execution, it shows C201, and so on.

Write Data to Any Sector

This procedure can be used to write data to any selected sector of the hard disk drive from the first 1/2 KB of memory. To execute the procedure, use the debugger program to write the data to the addresses shown.

1 Set the DMA write bit, select the drive, and restore the head to track 0:

4A0000:	4000	Sets bit 14 high, DMA write to disk from memory
4E0000:	0	Resets disk controller
4E0000:	19	Selects drive 0, head 1
E0000E:	1F	Restore command

2 Set the DMA address counter to 0:

4D0000:	0
4D4000:	0

3 Set the DMA word counter and toggle DMA enable:

460000:	0	Reads C000
460000:	C000	Reads C001, bit 15 is DMAEN, bit 14 is R/W*

4 Load an arbitrary data pattern in the first 16 words of memory at addresses 0h-2h.

5 Load the hard disk controller registers:

E00002:	0	Write compensation register
E00004:	1	Sector count register (selects the number of sectors to read)
E00006:	0	Sector number register (selects the sector to read)
E00008:	FF	Least significant byte of the cylinder number (cylinder 255 selected as an example)
E0000A:	0	D0 and D1 are most significant bits of the cylinder number

6 Issue the Seek command:

E0000E:	7F

It reads back D2 and the stepping motor rotates. D2 indicates that the command has executed with no errors.

7 Issue the Scan ID command to update the SDH register:

E0000E: 41

It reads back D2, indicating successful completion.

8 Check the SDH register by reading E0000C.

It contains FF21, which indicates that 512 bytes per sector and head 1 have been selected.

9 Issue the Write Sector command by writing 35 to address E0000E, and then use the read sector procedure above to verify that the data has been moved to disk.

After the Write Sector command has been issued, address E0000E shows DA if the command executes without error. If it does not execute properly, read address E00002, the error register, to determine the error.

Write fault and DRDY lines are checked by the controller during command execution.

Signals to Check if Procedures do not Execute

DRDY (IC21H, pin 28) must be asserted after the drive is selected in step 1.

SC (IC21H, pin 32) must be asserted from the drive when a Seek command is completed.

Unseparated data from the drive must appear at IC13K (pin 3) and separated data must appear at IC16M (pin 9) during a Read Sector command.

WFAULT (IC21H, pin 30) must not be asserted from the disk drive.

INTRQ (IC21H, pin 3) must be asserted at the end of a command and must be latched into the line printer status register at IC15P (pin 15).

TK000 (IC21H, pin 31) must be asserted after the Restore command is executed.

INDEX pulses (IC21H, pin 29) from the drive must appear after the drive is selected.

Disk RE (IC21H, pin 6) and WE (IC21H, pin 7) must be asserted during command execution.

TFER (IC24H, pin 18) must occur for each byte transferred. For every other TFER, there must be a DKBG (IC25B, pin 17) and a DKBGA (IC26F, pin 8).

Bit D0 of the WD1010 status register at address E0000E must be low after any command is issued. If it is not, read E00002, the error register, to determine the error.

Hard Disk Data Separator Test

This test verifies proper operation of the hard disk data separator and its component circuits. The test requires a running system with debugger PROMs installed, a serial terminal connected to the serial port, an oscilloscope, and a frequency counter. The hard disk data separator is shown on sheet 12 of the schematics.

The circuits that make up the data separator are:

- Voltage-controlled oscillator (output at IC14N, pin 10)
- Phase detector (outputs at IC18M, pins 11 and 8)
- Loop filter (output at IC17N, pin 1)
- Data run one shot (output at IC13N, pin 13)
- Data separation PAL (outputs at IC14M, pins 19 and 12)

To perform this test:

1 Select the hard disk drive.

Before the data separator can be tested, the hard disk drive must be selected to provide 5-MHz data and clock input to data separation PAL IC14M (pin 3). To select the drive and cause the drive to send data, push the Reset button and then use the debugger program to write 19 hex to address 4E0000. This causes pins 9 and 3 of IC17K (shown on sheet 11) to go low. These signals are sent to the hard disk drive, which responds by turning on its LED and generating the following signals (shown on sheet 11):

- DRDY from the drive goes low at IC18K (pin 11)
- INDEX pulses appear at IC18K (pin 5)
- TK000 goes low at IC18K (pin 3)
- SC goes low at IC18K (pin 13)

2 Check the data separator input and output.

Use a oscilloscope to check the input to the data separation PAL on sheet 12 at IC14M (pin 3). It contains 5-MHz data and clock pulses coming from the hard disk drive. Then check the output of the data separator at IC16M (pin 9), which shows 5 MHz. The DRUN signal at IC13N (pin 13) is high. RCLK at IC14M (pin 12) shows 5 MHz.

If these results are not obtained, test the individual circuits that make up the data separator using the following procedures (steps 3-5):

3 Check the voltage-controlled oscillator.

The voltage-controlled oscillator is IC14N. First check pin 11, the enable input; it is low to enable the oscillator. Next ground pin 1, the voltage control input, forcing the output frequency to a minimum of 3-4 MHz. Then connect pin 1 to 5 volts, forcing the output frequency to a maximum of 17-18 MHz.

4 Check the pullup and pulldown circuits and the loop filter.

To test pullup, ground pin 11 of IC18M, the pullup circuit. This saturates Q7. Its collector goes to +5 volts, causing the inverting input of IC17N (pin 6) to be +5 volts and the noninverting input to be +3.75 volts. This +3.75 volts results from the voltage drop across R67 to the voltage divider made up of R96 and R97. Thus the output at pin 7 drops to -12 volts. The section of 17N with output at pin 1 is connected as an inverting amplifier with a gain equal to the ratio of R65 to RP24, or about -2, and its input is fed from IC17N (pin 7). With an input of -12 volts, the output is driven to the positive supply voltage of +5 volts. This drives the output of the VCO IC14N (pin 10) to a maximum of about 17 MHz. Enable pin 11 must be low for the VCO to work.

To test pulldown, ground the output of the pulldown circuit at IC18M (pin 8). This causes the inverting input at IC17N (pin 6) to drop to 0 and the noninverting input to be +1.25. Thus the output at pin 7 is +5 volts, causing the output of the inverting amplifier to be -12 volts. This output results in -0.66 volts at IC14N (pin 1) of the VCO. The -0.66 volts is the output of voltage divider R64 and RP24. This voltage divider applies 1/3 of the -12 volts from the inverting amplifier output at IC17N (pin 1) plus 2/3 of the +5 volts connected to RP24. The total is -0.66 volts. The result is that the VCO outputs a minimum frequency of 3-4 MHz.

During normal operation, the capacitor C253 provides an integrating action. To check this, connect an oscilloscope on the output at IC17N (pin 7) when the VCO is locked on a 10-MHz output. The input to the VCO is about +2 volts. To obtain this, the output of the integrator at IC17N (pin 7) is about -4 volts. Each time a pullup pulse is generated, the voltage across C253 starts to swing negative at a rate calculated by dividing the current through R68 by the value of C253, or about 2.5 mA/0.01 uFd = 250,000 volts per second. Since typical pullup or pulldown pulses last only 10-20 ns, the output can change by about 5 mV for each pullup or pulldown pulse.

5 Check the phase detector.

The phase detector has a 5-MHz input from IC14M (pin 1) of the data separator PAL and a 5-MHz input from IC14M (pin 19). The 50-ns delay circuit together with the data flipflop output IC15K (pin 9) generates a 25-ns pulse at IC17M (pin 11). Pulses of 10 MHz are applied to IC16M (pin 11).

Dialer Test

The following addresses and data turn on the tone generator in the dialer chip and send tones to the speaker in the monitor. To turn the tone off, write the second address twice.

4B0560:	0	
4B0800:	0	Turns on maximum volume ringing
4B0500:	0	
4B0802:	0	Dual tone to speaker
4B0540:	0	
4B0802:	0	Single tone to speaker
4B0548:	0	
4B0802:	0	Single tone to speaker

Waveform Timing Analysis Procedure

This procedure checks signal generation and signal timing during various bus cycles of the logic board.

Special tools required include:

- HP1630D logic analyzer
- Piggyback extender cables
- Floppy diagnostics disk

Logic Analyzer Setup

1 Disassemble the **UNIX** PC to gain access to the logic board.

2 Install external cables to the piggyback board (if present).

3 Set up the HP1630D logic analyzer as follows:

- Apply power by depressing the line switch in the upper-right corner. The logic analyzer comes up in the system mode.
- In system mode, select timing mode by moving the down-arrow cursor key to the second line (- 16).
- Depress the format key and create the following labels:

LABEL	POL	POD1	POD0
1PCK	+	....	.......*
LDS	+	....	......*.
UDS	+	....	.....*..
DTACK	+	....	....*...
AS	+	....	...*....
R/W	+	....	..*.....
ADDEN	+	....	.*......
DBEN	+	....	*.......
RAMEN	+	.......*	....

- Go to trace mode and set the trace on the AS signal by putting a 0 under AS.

- Connect the signal probes from POD1 and POD0 to the following locations on the logic board:

Signal	Chip	Pin	Schematic
1PCK	14E	15	Sheet 5
LDS	14E	8	Sheet 5
UDS	14E	7	Sheet 5
DTACK	14E	10	Sheet 5
AS	14E	6	Sheet 5
R/W	14E	9	Sheet 5
ADDEN	17F	19	Sheet 5
DBEN	13D	9	Sheet 5
RAMEN	12C	19	Sheet 20

4 When the connections are made and the trace is set, insert the floppy diagnostics disk and power up the system.

5 When the disk has booted, depress the Run button on the logic analyzer.

This causes a waveform to be stored.

Typical Waveforms

The following waveforms show examples of a fast-cycle read operation. The first example shows the timing of AS-, which is approximately 130 ns after the beginning of the machine cycle.

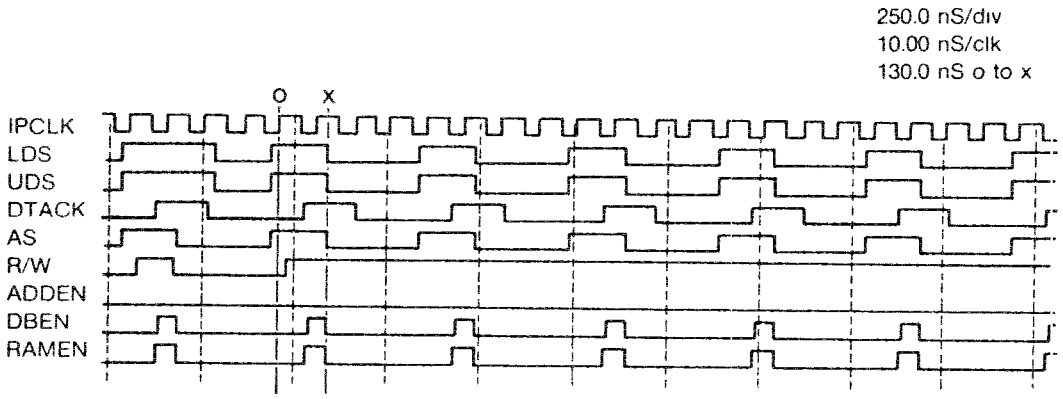


Figure 4-1 Timing of AS- (130 ns)

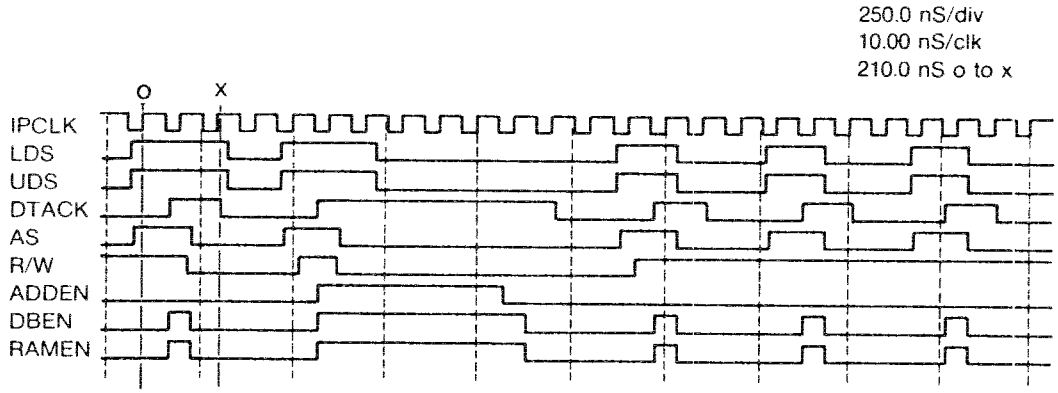


Figure 4-2 Timing of DTACK (210 ns)

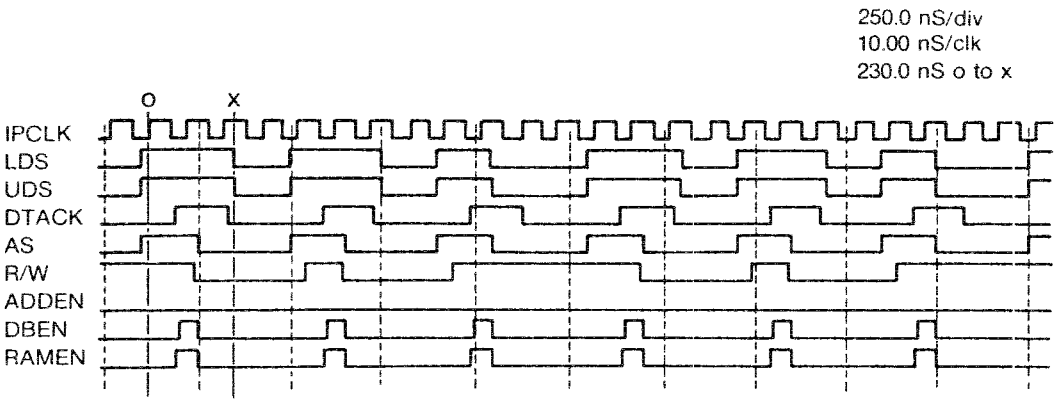


Figure 4-3 Timing of LDS and UDS (230 ns)

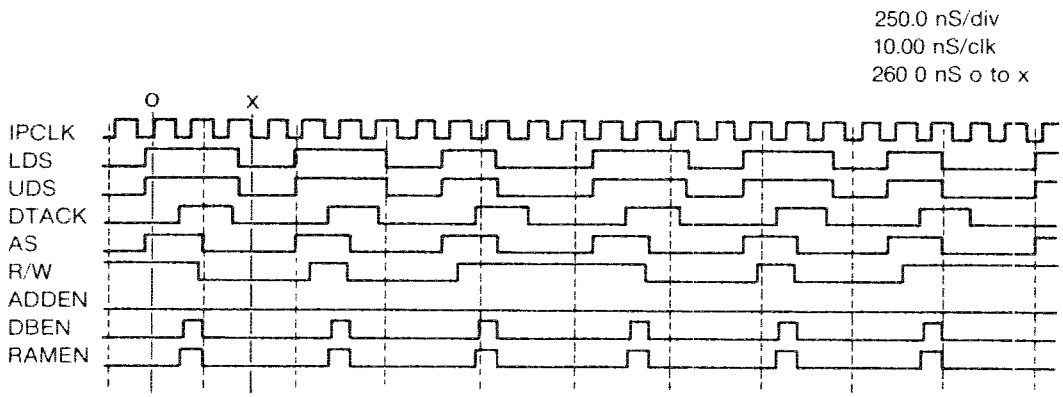


Figure 4-4 DTACK Latched by Processor (260 ns)

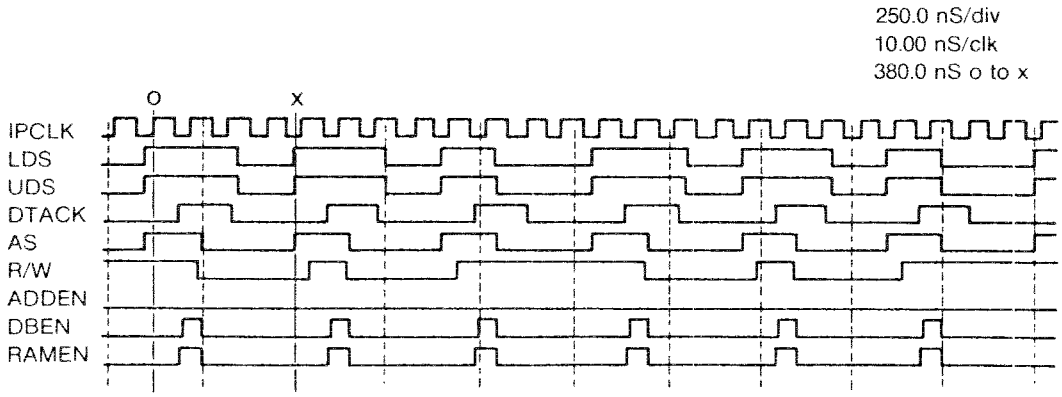


Figure 4-5 AS, LDS, and UDS Deasserted by Trailing Edge of Next 1PCK (380 ns)

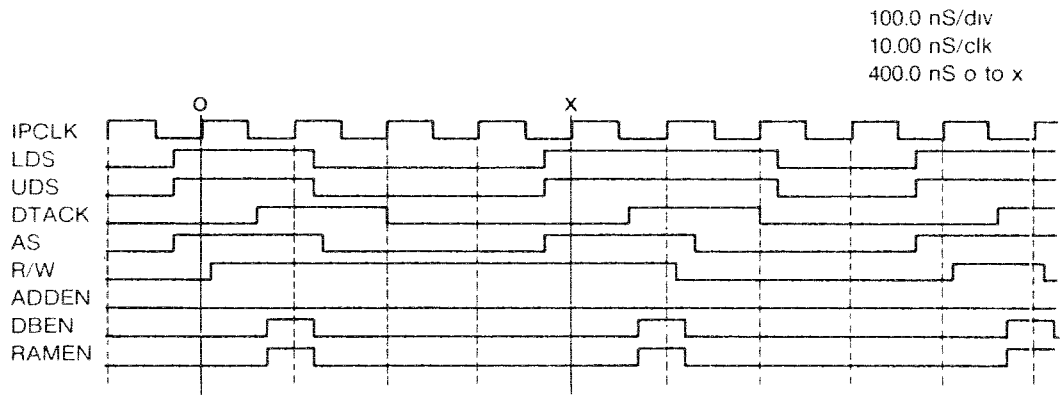


Figure 4-6 Length of Machine Cycle (400 ns)

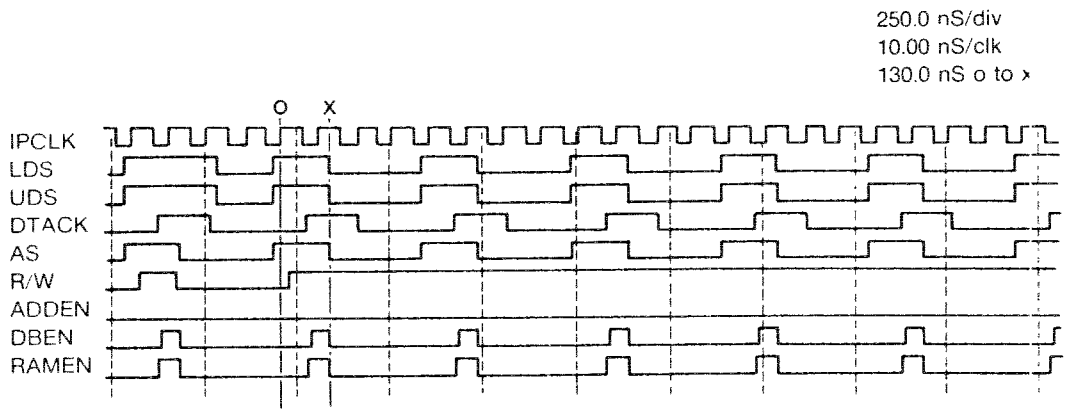


Figure 4-7 Timing of LDS and UDS During a Read Operation (130 ns)

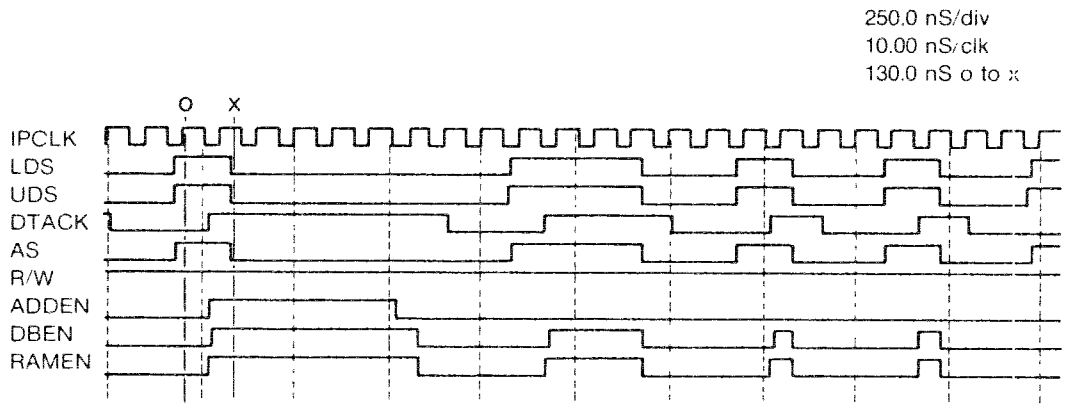


Figure 4-8 Slow Cycle Timing of AS (130 ns)

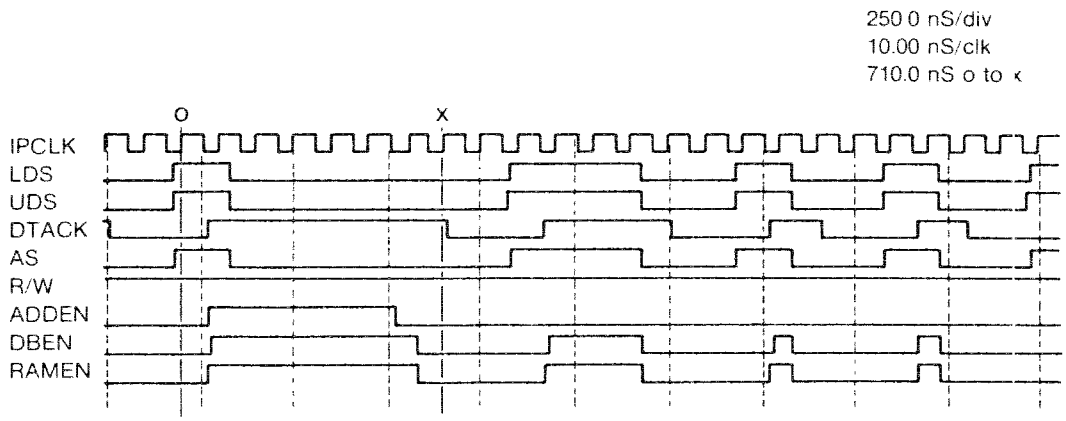


Figure 4-9 Timing of DTACK (710 ns)

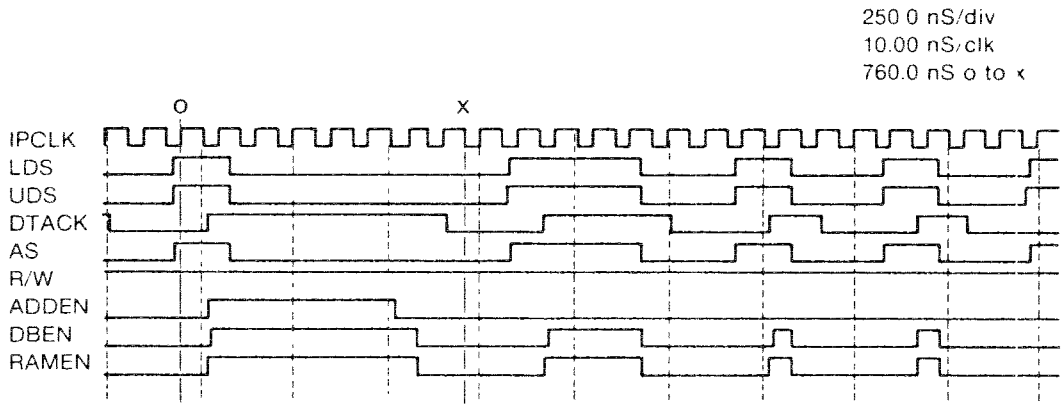


Figure 4-10 DTACK Latched by Processor (760 ns)

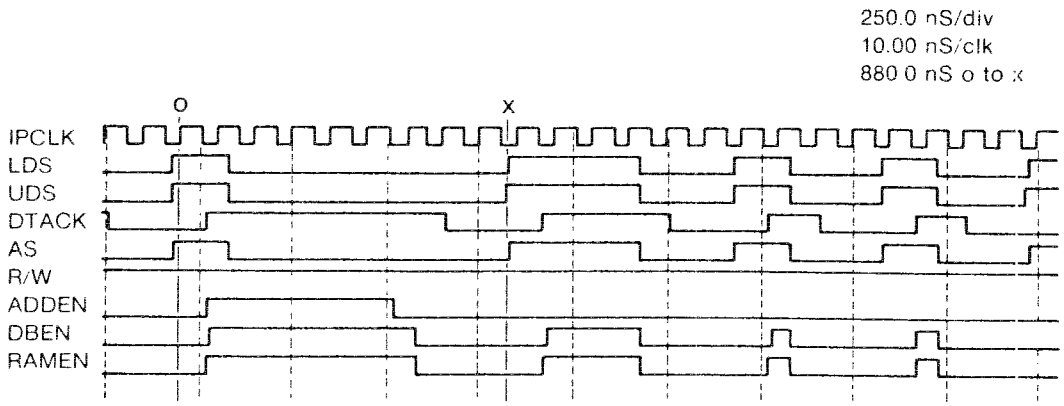


Figure 4-11 AS, LDS, and UDS Deasserted by Trailing Edge of Next 1PCK (880 ns)

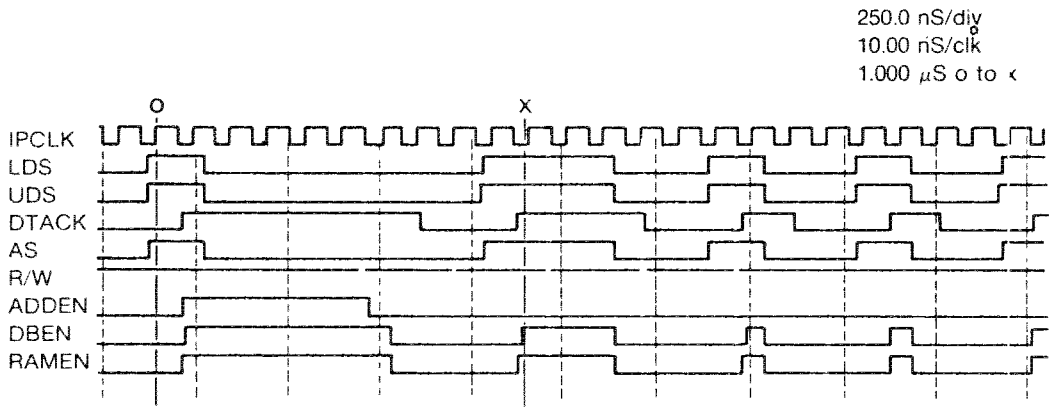


Figure 4-12 Slow Cycle Ends (Complete Cycle = 1.0 us)

68010 Timing Summary

The following waveforms show examples of a 68010 write cycle and a 68010 read-cycle summary.

The 68010 to DRAM write-cycle timing summary as shown in Figure 4-13 is as follows:

1 DTACK goes low before the trailing edge of state 4. This is a high-to-low clock pulse. From this reference point, the rest of the timing states are identified. The complete cycle takes place in 4 clock cycles of 100 ns each. Address line A23 goes low.

2 Address is tristate during S0 and becomes valid during state 1. A23 is valid at the end of state 1.

3 During state 2, the address strobe goes low and r/w (read-write) goes low.

4 When A23 goes low, it causes the following timing events to occur.

- o On sheet 3 of the schematic, 19K pin 6 goes low at next 2 PCK+ going hi. This is the start of state 2.
- o 21E pin 5 goes low generating ENRAS*. The propogation delay 19K and 21E causes ENRAS* going low to be delayed up to 12 ns after the rising edge of S1.
- o 28 to 32 ns after ENRAS* goes low T30* goes low.

5 On sheet 17 of the schematic, when T30* goes low, 3 to 8 ns later the RAS input to the DRAMs goes low. RAS going low strobes the row address from A3 to A11 into the DRAMs. 57 to 63 ns after ENRAS* goes low T60* goes low.

6 On sheet 18 of the schematic, when T60* goes low, the select inputs to the 74F258 multiplexers switch the address input to DRAMs from row address to column address. 4 to 11 ns after T60* goes low, the output of the multiplexers will be valid. Column address comes from the MAP RAMs.

The MAP RAMs receive valid address 2.5 to 6 ns after the start of state 2. The access time of the MAP RAMs is 35 ns.

The MA12-MA21 address output of the MAP RAM is valid at the input to the multiplexers 37.5 to 41 ns after the start of state 2.

7 On sheet 3 of the schematic, 87.5 to 100.5 ns after ENRAS* goes low, 19F9 goes hi. This causes 27F8 to go hi after 27N5 goes low generating DTACK* to the 68010. The propogation delay of the gates generating DTACK and variations in delay adds about 13 to 31 ns to the 90 ns output of the delay line. Thus, DTACK goes low at the 68010 input pin 10 between 100 and 126.5 ns after the start of state 2.

8 On sheet 5 of the schematic, DTACK* arrives at the processor about 24.5 to 50 ns before the falling edge of S4. DTACK* must go low before the falling edge of S4 to prevent the 68010 from inserting wait states. Output data from the processor is valid before the end of S3.

9 On sheet 16 of the schematic, T120* goes low 114 to 126 ns after NRAS* goes low. A low T120* causes 25K pin 6 to go hi which in turn makes 26M pin 11 hi and further 25F18 hi causing ENCAS+ to be hi. The delay of these gates, makes ENCAS+ hi between 13 and 37 ns after T120* goes low. Thus, the CAS input to the DRAMs goes low between 136 and 176 ns after the start of state 2.

Data from the 68010 is stable at the latest by the rising edge of state 4. It takes a maximum of 14 ns to pass data through the 74F245 data buffers. Data from the processor arrives at least 29 ns before it is strobed into the DRAMs by CAS.

10 On sheet 3 of the schematic, T120 going low causes 19F pin 7 to go hi and 28F pin 6 goes hi, k18G pin 8 goes low and 16K pin 13 goes hi. The rising edge of state 6 clocks MMUWREN+ hi. Variation in propogation time causes 16K pin 13 to go hi 122 to 145 ns after the rising edge of state 1.

16K pin 13 is hi a minimum of 55 ns before the rising edge of S6. MMUWREN+ going hi sets 18H4 hi. The falling edge of S7 clocks 18H3 hi resetting MMUWREND+.

11 On sheet 2 of the schematic, during the time that MMUWREN+ is hi, 25C pin 6 is low. This generates MMUWR*. MMUWR* enables tristate buffers 24F which output the new page status for the MAP RAMs.

12 On sheet 16 of the schematic, MMUWR* puts a low on the W* input to the 19C and 20C MAP RAMs. This causes the new page status to be written to the MAP RAMs during S6.

13 On sheet 3 of the schematic, MMUWREN+ hi causes 16K pin 13 to go low. The rising edge of S7 then clocks 18H pin 3 hi setting 21E pin 4 hi. If A23 is low on the next rising clock pulse, the memory cycle can start over.

14 On sheet 2 of the schematic, during a 68010 access to I/O address space, PA22 is hi and CASEN+ is not hi.

68010 to DRAM write cycle summary:

A23 before the leading edge of S2
ENRAS* low 6 to 13 ns after S2s leading edge
T30* goes low 37 to 51 ns after S2s leading edge
DRAM RAS 37-51 ns after S2s leading edge
DRAM ROW ADDRESS 5 to 12 ns after S2s leading edge
DRAM ROW ADDRESS worst case setup time 23 ns
T60* 63 to 75 ns after S2s leading edge
T90* 91.5 to 106.5 ns after S2s leading edge
DTACK* 100 to 126.5 ns after S2s leading edge
T120* 120 to 139 ns after S2s leading edge
DRAM CAS 136 to 176 ns after S2s leading edge
DRAM COLUMN ADDRESS to MUX input 37.5 to 41 ns after S2s leading edge
COLUMN ADDRESS set up before select input to MUX 22
COLUMN ADDRESS TO DRAM 67 to 86 ns after S2s leading edge
COLUMN ADDRESS SET UP before CAS 55 ns (write cycle)
DATA to DRAM 205 to 213 ns after S2s leading edge
DRAM DATA SET UP 29 ns.

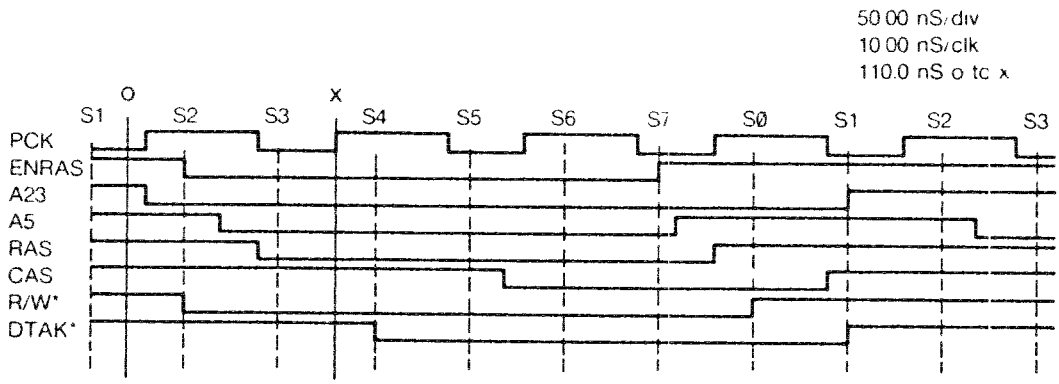


Figure 4-13 68010 to DRAM Write Cycle

The 68010 read cycle timing summary as shown in Figure 4-14 is as follows:

1 During a read cycle CAS to the DRAMs is generated earlier than during a write cycle because data from the DRAM chips is not valid until 60 ns after CAS or 120 ns after RAS.

2 CAS must go low before DTACK to allow time for data from DRAM to be stable at the input of the 68010. During a write, the opposite is true. CAS must not go low until data from the 68010 data bus is stable at the input to the DRAMs.

Read Cycle Summary:

CAS to DRAM 94.5 to 115.5 ns after S2 leading edge.
DRAM read column address set up 14.5 ns.
Data from DRAM valid 156.5 to 189.5 ns.
68010 data read set up time 60 ns.
Minimum row address hold 21 ns - 15 ns chip = 6ns.

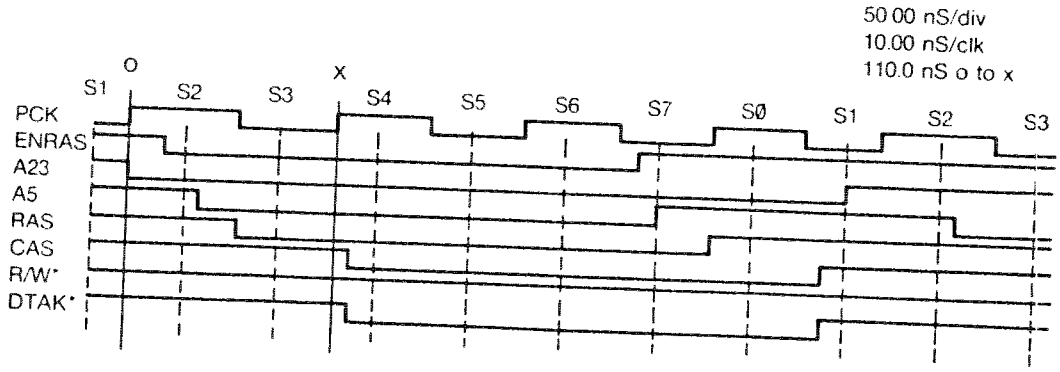


Figure 4-14 68010 Read Cycle Summary