Sun 3

Customer Maintenance

Training Manual

(Revision H)

September 10, 1988

Sun Microsystems, Incorporated
Educational Services Department

Credits and Trademarks

Sun Workstation is a registered trademark of Sun Microsystems, Inc.

SunStation, Sun Microsystems, SunCore, SunWindows, SunOS, SunView, DVMA and the combination of Sun with a numeric suffix are registered trademarks of Sun Microsystems, Inc.

UNIX, UNIX/32V, UNIX System III and UNIX System V are trademarks of AT&T Bell Laboratories.

Intel and Multibus are registered trademarks of Intel Corporation.

REVISION HISTORY

REV LEVEL	DATE	COMMENTS
A	01 June 1985	First edition of this manual
B	28 Feb. 1986	Second edition. Add 3/50 lowcost desktop Greyscale display Added to VME/MB
C	15 Dec. 1986	Third edition. Add 3/110 3/200 FPA Enhanced TroubleShooting section
D	15 Sept. 1987	Fourth edition. Add 3/60 low cost desktop color. New SCSI Enhanced Diagnostic section Enhanced Lab section New peripherals descriptions
E	15 Dec. 1987	Fifth edition. Add 3/150 6 slot chassis New Lab Projects General clean up
F	10 Mar. 1988	Sixth edition. Add ALM-2 Updated Lab Projects Updated Options Section
G	10 June 1988	Seventh edition Updated Lab Projects Updated Options Section Add GP2, CG5 and EXEC
H	10 Sept. 1988	Eighth edition. Updated Lab Projects, Add TAAC-1, Channel Attach, 3E Reference

Sun–3 Customer Maintenance Training

Course Objectives and Overview

This course will prepare technical users and support engineers to perform all aspects of Sun3 maintenance to the Field Replaceable Unit (FRU) level. Students will install, configure and verify functionality of all existing Sun3 workstations. Malfunctions will be inserted into each functional area for troubleshooting using all available Sun3 diagnostics. Sun3 add–ons and upgrades will be installed and diagnosed during the class.

Course Format

The course is 5 days (9Am -5Pm)
Instruction is a combination of lecture (40%) and guided hands–on lab with troubleshooting (60%).

Target Audience and Prerequisites

This course is designed for technical personnel responsible for installation, configuration, maintenance and upgrades of Sun3 workstations.
Sun Installation and Networking is a Prerequisite. Experience with digital computer architecture and system troubleshooting is assumed.

You are overqualified for this course if: You are able to install, configure, upgrade
and troubleshoot the entire product line of Sun3 workstations

You are underqualified for this course if: You have not taken the Sun Installation and Networking course.

Materials

Students receive a course manual with system and architecture block diagrams,
guided lab projects, theory of operations and configuration charts including all user selectable jumpers.

Overview of Topics

Sun3 products product description
Sun3 Architecture (includes Motorola MC68020, MC68881)
Installation and operation
Configuration
Power
VME Bus operation
SCSI Bus and device operation
Video theory and operation
Memory theory and operation
Diagnostics
System Troubleshooting
Upgrades and add-ons

Day 1

Introduction
Course objectives, content and schedule (.5 hours)
Sun3 Overview
Physical identification and capabilities (1.5 hours)
System Block diagram
Device descriptions, Data flow, Control flow
MC68020, MC68881 (1 hour)
VME Bus
Operation and error codes (.5 hours)
SCSI Bus
Operation ,device descriptions and error codes (.5 hours)
Lab
Familiarization and booting procedures, physical identification of systems, verification of working systems (3 hours)

Day 2

Configuration
Board, device and cable placement
jumper and switch settings, power adjustments (3 hours)
Lab
Configure all boards, devices and switch settings. perform power adjustments (4.0 hours)

Day 3

Sun Diagnostics
How to load, run and interpret all Sun
diagnostics and monitor commands
(includes EEPROM) (3.0 hours)

Lab
Set up, run and interpret all Sun diagnostics
and monitor commands (4.0 hours)

Day 4

System troubleshooting Lab
Isolate and repair inserted malfunctions
in each system area (8.0 hours)

Day 5

System add ons and upgrades
Theory and installation procedures for upgrades such as
graphics and Asynchronus Multi-Line
Adapter (ALM) (3.0 hours)

Lab
Install and verify upgrades (4.0 hours)

CHAPTER BREAKDOWN

TABLE OF CONTENTS

CHAPTER SEVEN: SECTION A

CHAPTER SEVEN: SECTION B

CHAPTER SEVEN: SECTION C

CHAPTER SEVEN: SECTION D

CHAPTER EIGHT: SUN 3/50

CHAPTER NINE:

CHAPTER TEN:

SUN 3/200 SERIES PROCESSOR AND MEMORY

CHAPTER ELEVEN

3/60 Desktop Color Workstation

APPENDICIES

CHAPTER ONE

SUN 3 ARCHITECTURAL OVERVIEW

OVERVIEW

This training manual provides a detailed study of Sun 3 architecture as it is implemented on the Sun 3 workstation product line. Sun 3 architecture implements the processing power and capabilities of the 68020 32-bit microprocessor, in a VME bus environment, and adds enhanced floating-point operations extended with the 68881 floating-point coprocessor.

The following sub-sections are an overview of the architecture in functional blocks and will seldom go to component level detail. Occasional references are made to specific component characteristics necessary to describe a Sun 3 operation.

What this manual will contain is a detailed view and introduction to the 68020 microprocessor, how it is used in Sun 3 architecture, and how Sun 3 products comply with the VME bus specification. This manual only covers the current release of SUN 3 products. At the release time of this manual, the Sun 3 product line included: the 3/75M, a general purpose desktop workstation, the 3/140, a 3 slot version with the Sun 3 CPU, and the 3/150, a six slot system using the same technology as the 3/140; the 3/160M/G/HM/C (monochrome, greyscale, hi-res monochrome, and color), a 12-slot pedestal workstation; the 3/180FS, a 12-slot file server; the 3/50, a low cost single board system; the 3/110C, a 3-slot, low cost color workstation; the 3/200 series, a 25 MHz high end workstation with high resolution video display; and the 3/60, a single board desktop color workstation. The following is a summary of the features found on each system.

SUN 3/75, 140 and 150	SUN 3/160M or 3/180FS	SUN 3/160C or G
68020 at 16.67 MHz	68020 at 16.67 MHz	68020 at 16.67 MHz
68881 at 16.67 MHz	68881 at 16.67 MHz	68881 at 16.67 MHz
4 MB memory standard Demand-paged virtual memory at 256 MB addr space/process, total memory expansion option can be up to 16 MB	4 MB memory standard Demand-paged virtual memory at 256 MB addr space/process, total memory expansion option can be up to 16 MB	4 MB memory standard Demand-paged virtual memory at 256 MB addr space/process, total memory expansion option can be up to 16 MB
2, 3 or 6 slot cardcage	12-slot card cage	12-slot card cage
19-inch landscape monochrome monitor at 1152 x 900 x 1 resolution	19-inch landscape monochrome monitor at 1152 x 900 x 1 resolution	19-inch landscape color monitor at 1152 x 900 x 8 resolution with 256 displayed colors, color palette is over 16 M
Refresh rate at 66 Hz, non-interlaced	Refresh rate at 66 Hz, non-interlaced	
32-bit VME bus	32-bit VME bus	Graphics processor and graphics buffer for enhanced high-resolution graphics
Add-on sub-system storage(1/4-inch tape, and/or 71-327 MB drive, formatted.	Communication SunLink option for SNA and BSC environments and routing	Refresh rate at 66 Hz, non-interlaced
SUN 3/50	8 or 16 channel asynch line multiplexer(ALM)	Communication SunLink option for SNA and BSC environments and routing
15 MHZ 68020,optional 68881. Single slot system.	VME/MB IEEE 796 adapter, or 3/2 VME adapter optn	8 or 16 channel asynch line multiplexer(ALM)
The 3/50 has on-board SCSI control logic, and. features VLSI CMOS gate arrays. User can use on board transcievers for coax ethernet or standard ethernet.	Note that the 3/180 is a 19-inch rack mountable system, and can include a 1/2-inch tape subsystem, and a large capacity mass storage subsystem.	VME/MB IEEE 796 adapter, or 3/2 VME adapter optn
		32-bit VME bus

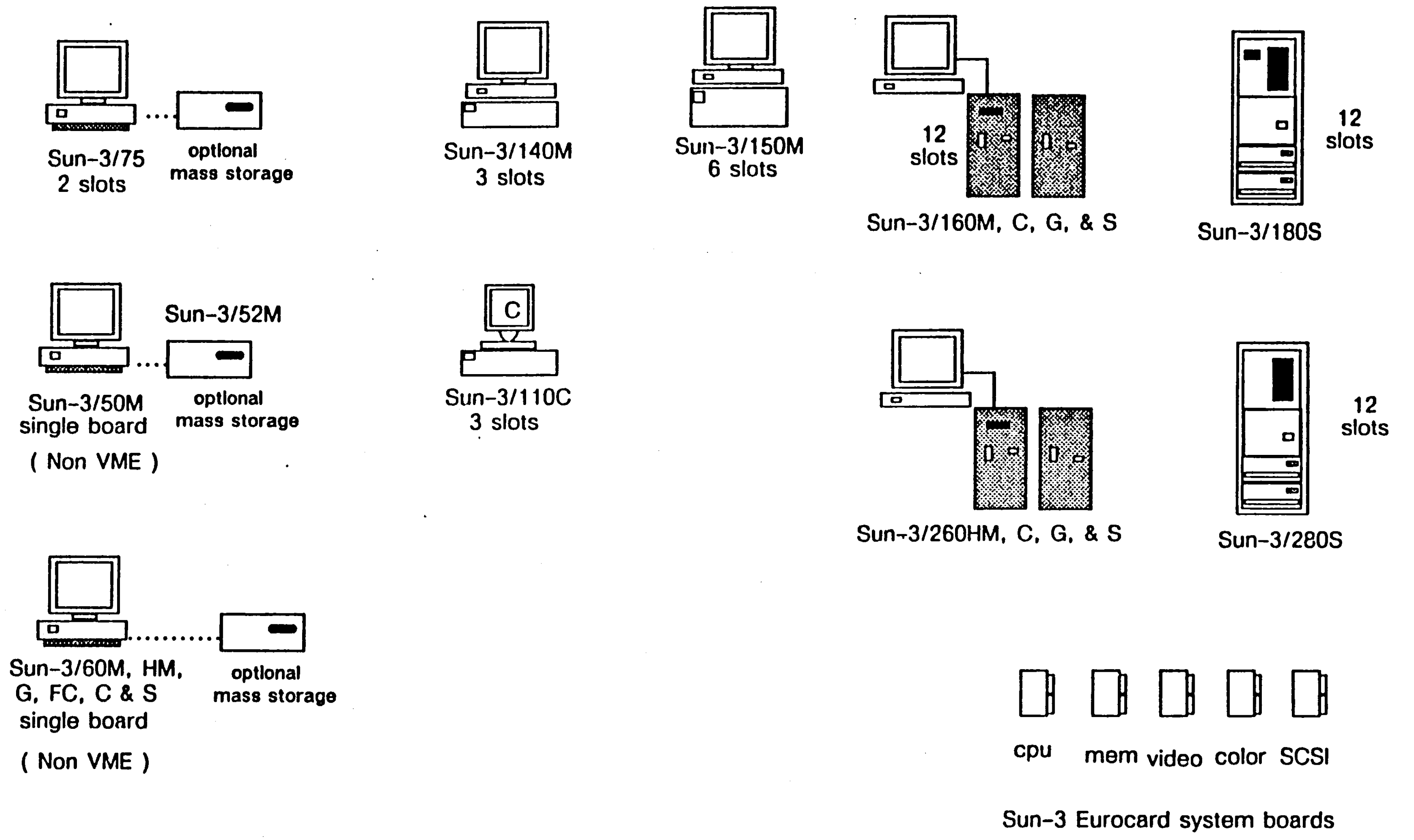
SUN 3/110C		SUN 3/200 SERIES	
68020 at 16.67 MHz	3-slot system support	68020 at 25.00 MHz	12-slot system
68881 at 16.67 MHz		68881 at 20.00 MHz	64-bit data bus
4 MB memory standard(up to 12 MB)		8 MB ECC memory standard(up to 32 MB)	
128 KB frame buffer, and 1 MB FB for on-board color support		256 KB video FB 1600 by 1280 res.	64 KB write back data cache

SUN 3/60	
20 MHZ 68020,68881. Single slot system	4MB Memory Std. Up to 24 MB Opt.
On board Color, Mono or HiRes	Onboard SCSI Controller
User can use onboard transceiver for coax ethernet	P4 "Piggyback" Bus for all video

Figure 1-0: Sun 3 Workstation Features

SUN 3 PRODUCT LINE

68020 **DESKTOP** **DESKTOP/SIDE** **DESKSIDE** **RACK MOUNT**



Sun-3/75
2 slots

optional
mass storage

Sun-3/140M
3 slots

Sun-3/150M
6 slots

12
slots

Sun-3/160M, C, G, & S

12
slots

Sun-3/180S

Sun-3/52M

Sun-3/50M
single board
(Non VME)

optional
mass storage

Sun-3/110C
3 slots

Sun-3/260HM, C, G, & S

12
slots

Sun-3/280S

Sun-3/60M, HM,
G, FC, C & S
single board
(Non VME)

optional
mass storage

cpu mem video color SCSI

Sun-3 Eurocard system boards

1.1 68020 MICROPROCESSOR OVERVIEW

The 68020 microprocessor used in Sun 3 architecture is a 32-bit VLSI device, implemented with an asynchronous bus structure, including 32-bit registers and data paths, 32-bit address registers, and enhanced instruction sets and addressing modes. The 68020 is object code compatible with the 68010 virtual memory microprocessor used in earlier Sun 2 architecture machines.

The 68020 32-bit address and data buses are non-multiplexed bus structures which supports a dynamic bus sizing device that allow the microprocessor to do data transactions with a number of operand size bus devices, including 32-bit, 16-bit and 8-bit data transfer operations.

The 68020 microprocessor on the Sun 3 CPU card runs at 15, 16.67, 20 or 25 Mhz (Model dependent) works with Sun's external virtual memory management unit (MMU) and can interface with other coprocessors such as the 68881 floating point processor.

The 68020 supports the control signal groups (plus some additional control signals) found on earlier Sun 2 68010-based products. Figure 1-1 contains a description of the signal groups.

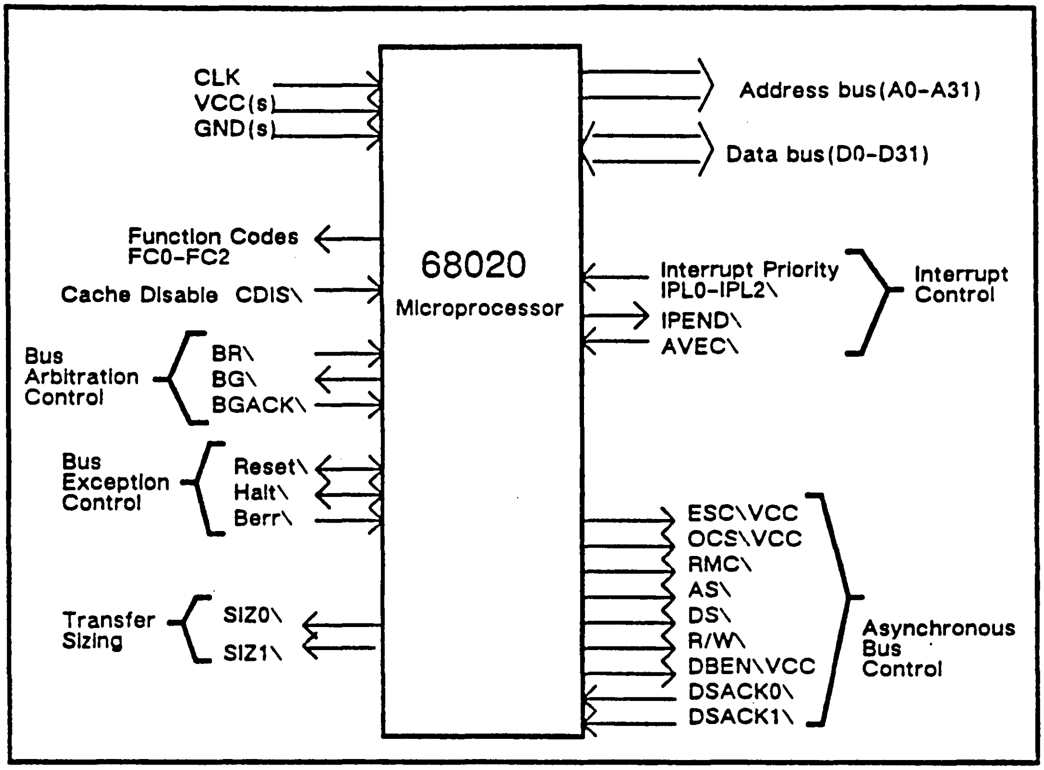


FIGURE 1-1: 68020 SIGNAL DESCRIPTION

1.3 68020 OPERATIONS

During the execution of a 68020 cycle, the bus controller (illustrated in block form in Figure 1-2) loads instructions from the data bus onto the instruction decode block and the on-chip instruction cache. Note that cache can be controlled by software. The sequencer and control unit block provides for overall chip control, helps in management of the internal buses, as well as the internal registers. This block also provides control for the functions of the execution unit.

The 68020 has 16 32-bit general purpose registers, a 32-bit program counter, a 16-bit status register, a 32-bit vector-base register, two 3-bit function control registers (source and destination), and two 32-bit cache (address and control) registers.

The vector base register is used to determine the location of the exception vector table in memory (where the routine for that particular exception or interrupt handler begins). The status register contains the interrupt priority mask

(3-bit), the condition codes (for branching decisions),trace enable bits, a supervisor/user bit and a Master/Interrupt bit.

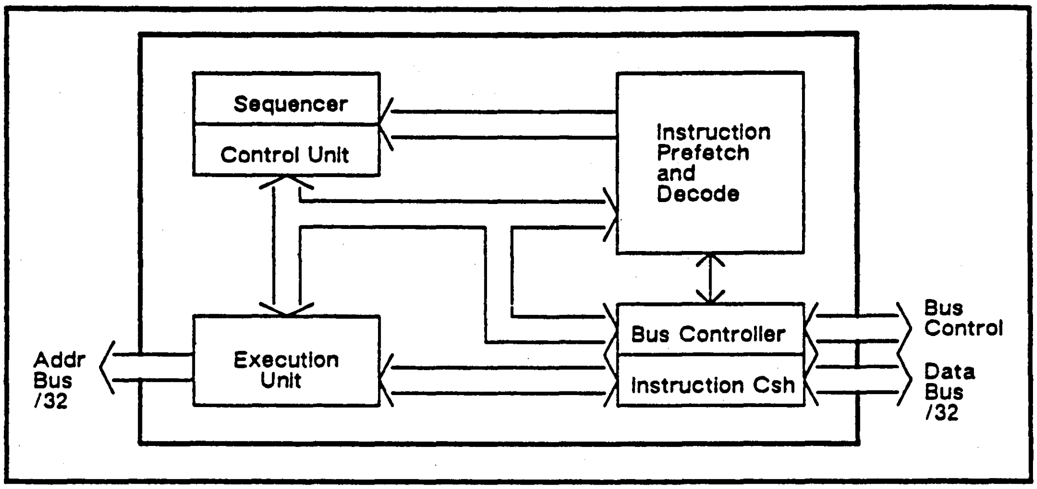


FIGURE 1-2: 68020 INTERNAL BLOCK DIAGRAM

1.4 VIRTUAL MEMORY IMPLEMENTATION

The extended addressing range of the 68020 microprocessor is 4 giga bytes. The actual physical memory found will be smaller. Sun uses a 28 bit virtual address, so the extent of virtual address space is 256 Mbytes per process. The other three bits of the VA range are used to construct a context which allows rapid switching between processes.

Using a technique called virtual memory access, a virtual memory based system support is provided for the limited amount of high-speed physical memory that can be accessed directly by the processor, while maintaining a much larger 'virtual memory' on a secondary storage device (such as a large capacity disk drive). This process is designed to support a 'demand paging' scheme.

As the microprocessor attempts to access a location in the virtual memory map, that is not actually residing in the physical memory (page invalid) its access is temporarily suspended while the actual data is fetched from the secondary storage device and written to physical memory, via the demand paging scheme program pointed to by the invalid page exception. The microprocessor can then restart or continue its access cycle. To restart, the 68020 uses instruction continuation for virtual memory management.

In order to implement instruction continuation, the 68020 stores its internal state on the supervisor stack when a cycle has been suspended. The 68020 will load the program counter with the address of the virtual memory location, and thus, will be able to restart (continue) with program execution at the new address.

This holds true for bus errors which cause exception processing as well. When the microprocessor bus error exception handler routine has completed, an instruction (RTE=return from exception) is executed to reload the 68020 with the internal state stored on the supervisor stack. The processor can then rerun the faulted bus cycle, or continue (restart) the suspended instruction cycle via the virtual memory map. On Sun 3 architecture, an external memory management unit is provided to support virtual memory. Information on this circuit is provided later on in this section.

1.5 SPACE ORGANIZATION ON SUN 3 ARCHITECTURE

Sun 3 architecture is divided into three spaces including the CPU space, the control space, and the device space.

The CPU space is comprised of the 68020, and coprocessors such as the 68881 floating point processor.

The control space is considered the core of Sun 3 architecture and includes the Memory Management Unit (MMU) and extensions of the CPU such as the bus error register, the system enable register, the diagnostic register, and the system ID PROM. These devices will be discussed later on in this chapter. The ID PROM will contain the machine 'type' implemented for each respective architecture level (e.g. Sun 3 architecture).

The device space will define the devices that exist in the architecture and how each will be accessed. The devices space includes the main memory, the video frame buffer, the system bus master interface (VME bus), and the I/O devices.

All levels of Sun 3 space types can be accessed by the CPU via the MMU with the exception of the 68881. The CPU can directly access the floating point processor. The primary purpose of the MMU will be to allow access to the devices in a shared and protected environment. The MMU will also manage address translation (from virtual-to-physical addresses) and the context of the current process

DVMA devices such as Ethernet, SCSI and VME slaves as well as the CPU will arbitrate for (and share) the virtual address bus and will be discussed in the next section. Figure 1-3 illustrates how the CPU, MMU and Devices are connected in Sun 3 architecture.

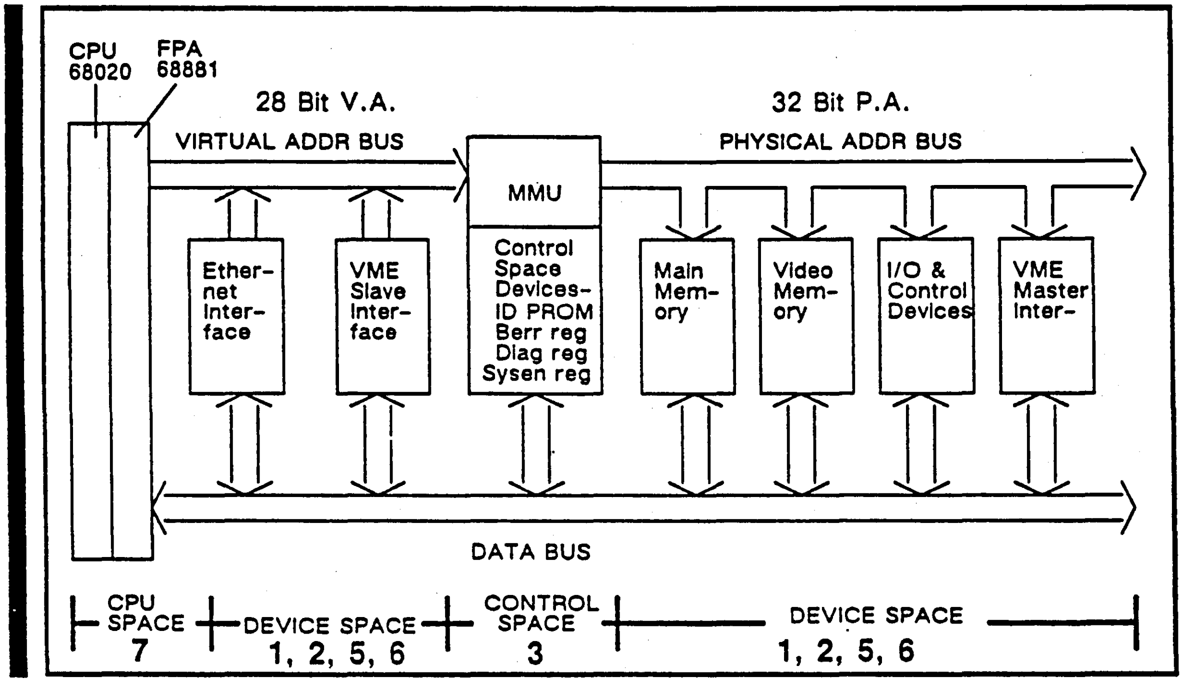


FIGURE 1-3: ARCHITECTURAL BLOCK OVERVIEW

Sun 3 architecture is divided into three spaces: CPU space, Control space, and Device space. An image of this space division is maintained by the processor (aided by the MMU) to map out the spaces. The Function Code table is used to describe how they are mapped out.

The contents of the Function Code table identify the processor state and the address space of the bus cycle currently being executed. The MMU is used to decode the table (see MMU section) so that the memory system can utilize the 4 GB address range for the space divisions. Figure 1-4 illustrates the Function Code table and space assignment. The MMU section of this overview will describe how they are decoded during the execution of a particular cycle. Note while operating under an operating system, the decoding will also provide useful information on a successful or unsuccessful process cycle.

FC2	FC1	FC0	CYCLE TYPE	FC	ADDRESS SPACE
0	0	0	Undefined,Reserved	0	Reserved
0	0	1	User Data Space	1	Device Space(usr data)
0	1	0	User Program Space	2	Device Space(usr progm)
0	1	1	Control devices	3	Control Space
1	0	0	Undefined,Reserved	4	Reserved
1	0	1	Supervisor Data Space	5	Device Space(supr data)
1	1	0	Supervisor Program Space	6	Device Space(supr prog)
1	1	1	CPU Space	7	CPU Space

FIGURE 1-4: ADDRESS SPACE ASSIGNMENTS

Note that the CPU space uses Function Code level 7. Such space cycles will include the processor, coprocessor, interrupt, breakpoint and ring-protection cycles IE: after an exception. Supervisor space Function codes are generated when the supervisor bit in the status register is on, (FC 6 would be on for an instruction fetch) Control space cycles will be level 3 and include access to the memory management unit (MMU), the bus error register, the system enable register, the diagnostic register and the ID PROM. The device space will include all devices that will be accessed by the CPU with data or program instructions (e.g. main memory, VME bus master interface and I/O devices).

1.7 MEMORY MANAGEMENT UNIT (MMU) OVERVIEW

The purpose of the Memory Management Unit (MMU) is to ensure that devices have access to the memory and to each other in a fully protected, shared and managed, multiprocess environment. The MMU will manage three memory states including:

1. The mapped state - where a process is resident and mapped into the page map, making the mapped process runable.
2. The resident state - where the resident process state has a user page and page table in memory.
3. The swapped state - where a swapped process is completely residing on the secondary storage device (usually the disk).

The MMU logic is implemented with a context register, a segment map, and a page map. Virtual addresses from the processor and DVMA devices will be translated into physical addresses by the previously written segment and page maps and previously chosen context.

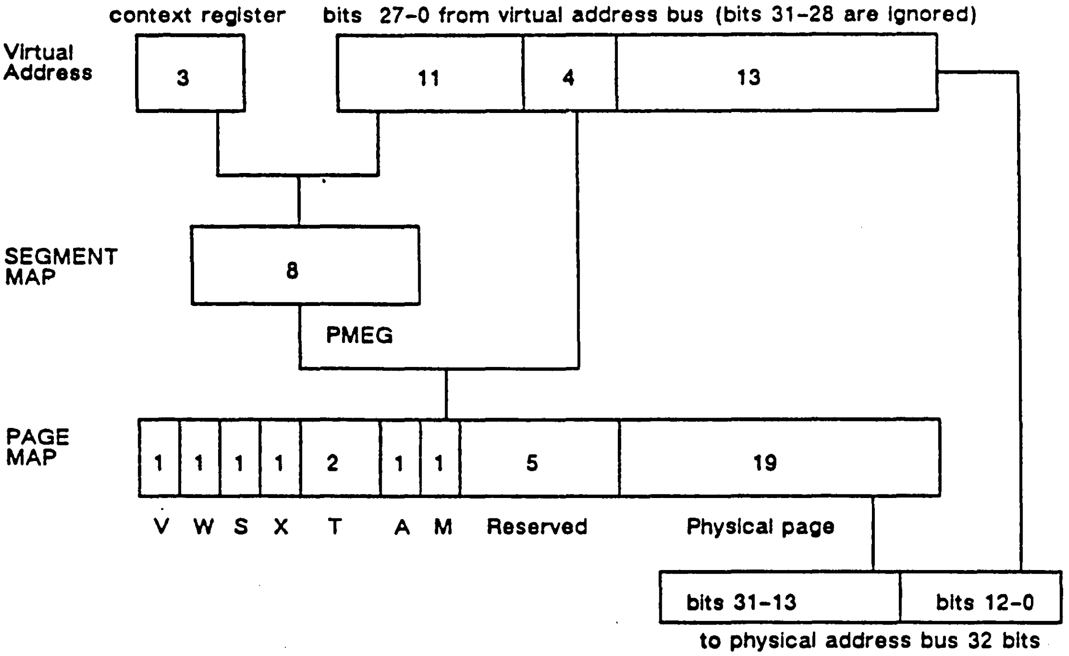
The MMU uses a page size of 8K bytes and a segment size of 128K bytes. There are 8 separate contexts which can be thought of as processes with an address space of 256M bytes for each context.

The current context is selected ("switched") in the 3-bit context register by a single instruction from the 68020. This has proved to be a highly efficient mechanism to switch from one process to another. This way we can give meaning to a supervisor or user 'context'. In other words, define the context of the address space access by reading the bits in the register. When referring to the context of a cycle, access generate by the the KERNEL will typically operate within a supervisor context, while application processes will operate within a user context.

The Segment map has a total of 16384 entries (refer to Figure 1-5 and 1-6). The segment maps are indexed by the current context (3-bits from the context register) and the 11 most significant bits of the virtual address. The Segment map is divided into 8 contexts of 2048 entries each for a total of 16384 entries.

These segment map entries are 8-bits wide and are used to point to a Page Map Entry Group (PMEG). The PMEG is used as a pointer, producing an intermediate address which will be translated to a physical one via the page map.

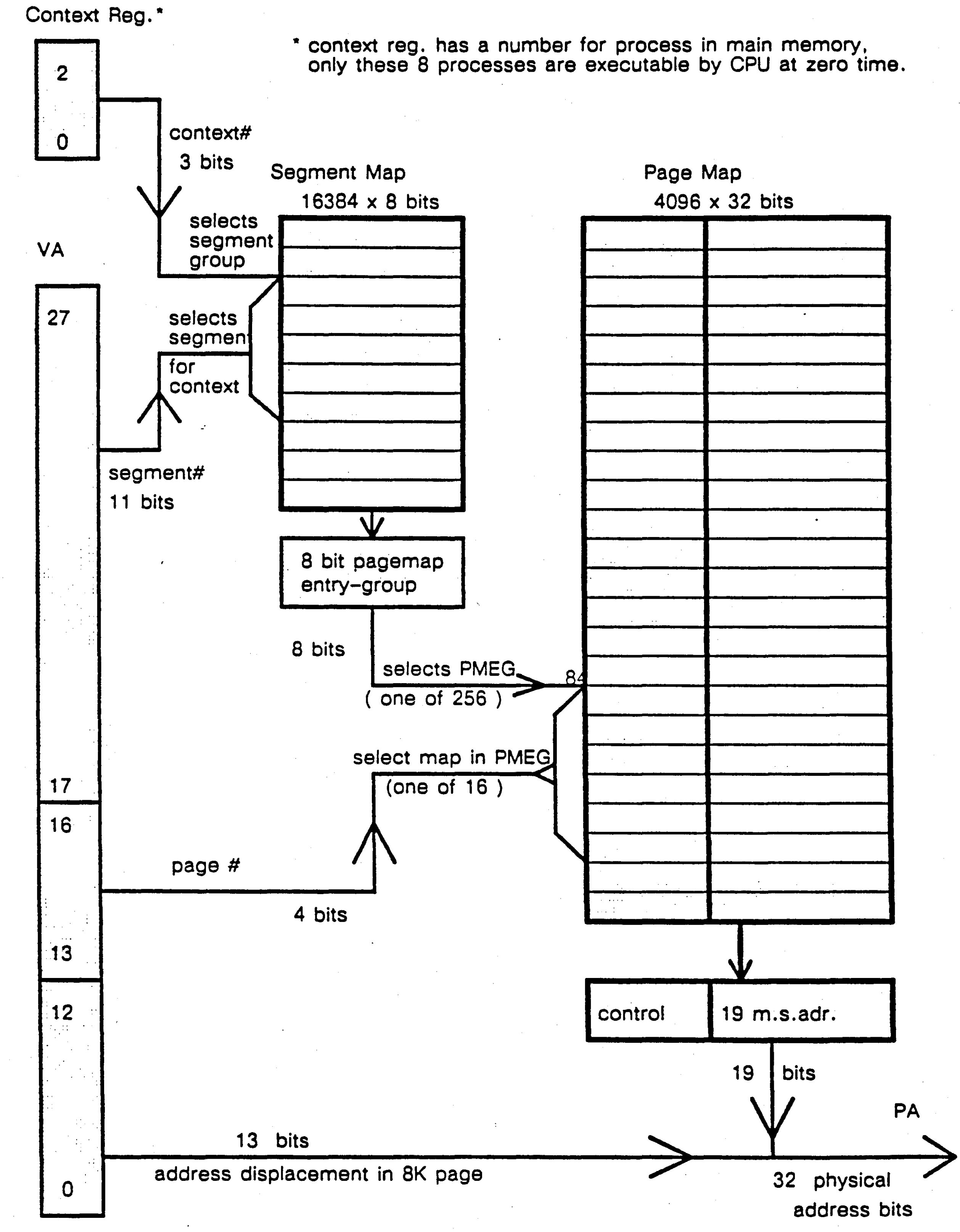
The Page Map contains 4096 page entries with each mapping an 8K byte page. The page map entries will be composed of a valid bit, protection field, do not cache bit, type field, access and modification bits, and a page number. Figure 1-5 illustrates this entry field. The page map itself is divided into 256 sections of 16 entries each. Each section, then, will be pointed to by the segment map entry, or PMEG. Refer to Figure 1-5 and 1-6 on the next 2 pages.



SO THAT:
V=valid bit(implies read access)
W=write access bit
S=system access bit(supervisor)
X=do not cashe bit
T=type access**
A=accessed bit
M=modified bit

TYPE IS:**
type 0=main memory
type 1= on-board I/O
type 2=VME bus(system bus) 16-bit data
type 3=VME bus(system bus) 32-bit data

FIGURE 1-5: MEMORY MANAGEMENT ADDRESS TRANSLATION



* context reg. has a number for process in main memory, only these 8 processes are executable by CPU at zero time.

VA = virtual address

PA = physical address

UNIT DESCRIPTION	SIZE
Page size	8K bytes
Segment size	128K bytes
Process size	256M bytes
Number of contexts	8
Number of segments per context	2048
Number of pages per segment	16
Number of pages	256
Number of pages total	4096
Number of segments total	16384

FIGURE 1-6: MMU SUMMARY ASSIGNMENTS

1.8 PHYSICAL ADDRESS MAP

The physical address map is fixed for all implementations of Sun 3 architecture. The number of physical address bits which define a devices address space is dependent on the MMU 'type field' illustrated in Figure 1-5. Figure 1-7 illustrates the physical address assignments used in Sun 3 architecture.

TYPE	MMU BASE ADDRESS	DEVICE	
0	0x00000000	Main Memory	
1	0x00000000	Keyboard/Mouse	
	0x00020000	Serial Ports	
	0x00040000	EEPROM	
	0x00060000	TOD Clock	
	0x00080000	Memory Error Register	
	0x000A0000	Interrupt Register	
	0x000C0000	Ethernet Interface(if Intel)	
	0x000E0000	Color Map	
	0x00100000	EPROM(Boot PROM)	
	0x00120000	Ethernet Interface(if AMD)	
	0x00140000	SCSI Interface(3/50 or 3/60 only)	
	0x00160000	Reserved	
	0x00180000	Reserved	
	0x001A0000	Reserved	
	0x001C0000	DEP	
	0x001E0000	Undefined,Reserved	
2	0x00000000	32-bit address space	VME bus 16-bit data
	0xFF000000	24-bit address space	
	0xFFFF0000	16-bit address space	
3	0x00000000	32-bit address space	VME bus 32-bit data
	0xFF000000	24-bit address space	
	0xFFFF0000	16-bit address space	

FIGURE 1-7: PHYSICAL ADDRESS ASSIGNMENTS

1.9 DEVICE AND CONTROL DATA PATHS

The following sections will cover the device and control logic physically residing on-board the CPU card and interfaced to the 32-bit data bus as well as the byte bus (8-bit) on the CPU card. This will be an overview of the main data path, the MOS data path, and TTL 8-bit data bus, and their respective interface. The bus paths and component groupings in this section will be related in the accompanying block diagram in Figure 1-8 on the following page.

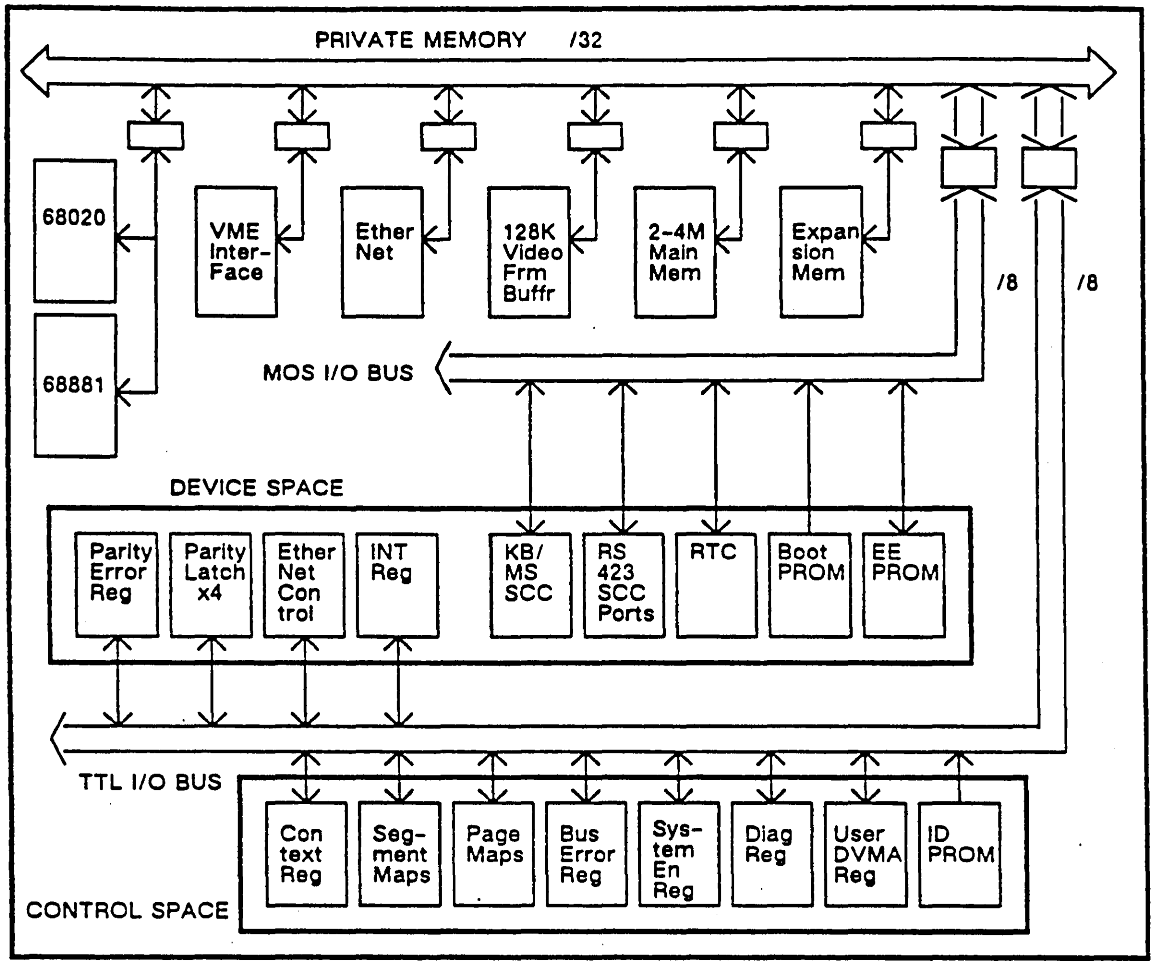


FIGURE 1-8: SUN 3 DATA BUSSING

There are two bus sizes used on the CPU card including the 32-bit and the 8-bit buses. The 32-bit bus provides the high bandwidth data path between the processor, the DVMA devices, the VME bus and the main memory. The 8-bit bus is used to reduce board routing problems during data transaction cycles. This is due primarily to the different driving capabilities between MOS and TTL devices. The MOS devices have weaker bus drivers and are sensitive to undershoot. The TTL devices, on the other hand, have the opposite drive characteristics, and are therefore prone to overshoot.

For devices on the 8-bit bus that will require a larger interface bandwidth, such as the Page Maps, the dynamic bus sizing capability of the 68020 is used so that long word transactions can be accomplished. This holds true for the Parity Latch as well (x4 latch).

1.11 CONTROL DEVICES ON THE TTL BUS

1.12 ID PROM

The ID PROM, located on the 8-bit TTL bus, is a read only device which provides the system bus with key information about the machine. The data fields included on this device are the format of the PROM, the machine type-specifying the type of architecture (i.e. Sun 3), the Ethernet address-a unique address assigned to the Sun Workstation, the date the PROM was generated, the serial # of the idprom (not the system serial #), a checksum-specifying the PROMS checksum yield, and a reserved field for future PROM revisions. The ID PROM is read during the BOOT stage, and can be read by initiating a 'K2' reset to the system, or by invoking 'KB'. Figure 1-9 shows the contents of the PROM.

ENTRY FIELD	SIZE
(1) Format	1 byte
(2) Machine Type	1 byte
(3) Ethernet Address	6 byte
(4) Date	4 byte
(5) Serial Number	3 byte
(6) Checksum	1 byte
(7) Reserved	16 byte

FIGURE 1-9: ID PROM CONTENTS

Machine Type is user by the Kernel in determining the CPU type on boot.

Ethernet number is used when diskless clients request a server and for low level file transfer.

Unix hostid = Machine Type followed by Serial Number (of I.D. Prom)

Bad or wrong ID PROMs can cause watchdog resets, invalid frame buffers, SCSI boot timeouts, client boot problems and spurious interrupts.

Hostid is used by some companies to protect their products against unauthorized duplication and distribution. Hostid is a SunOs command.

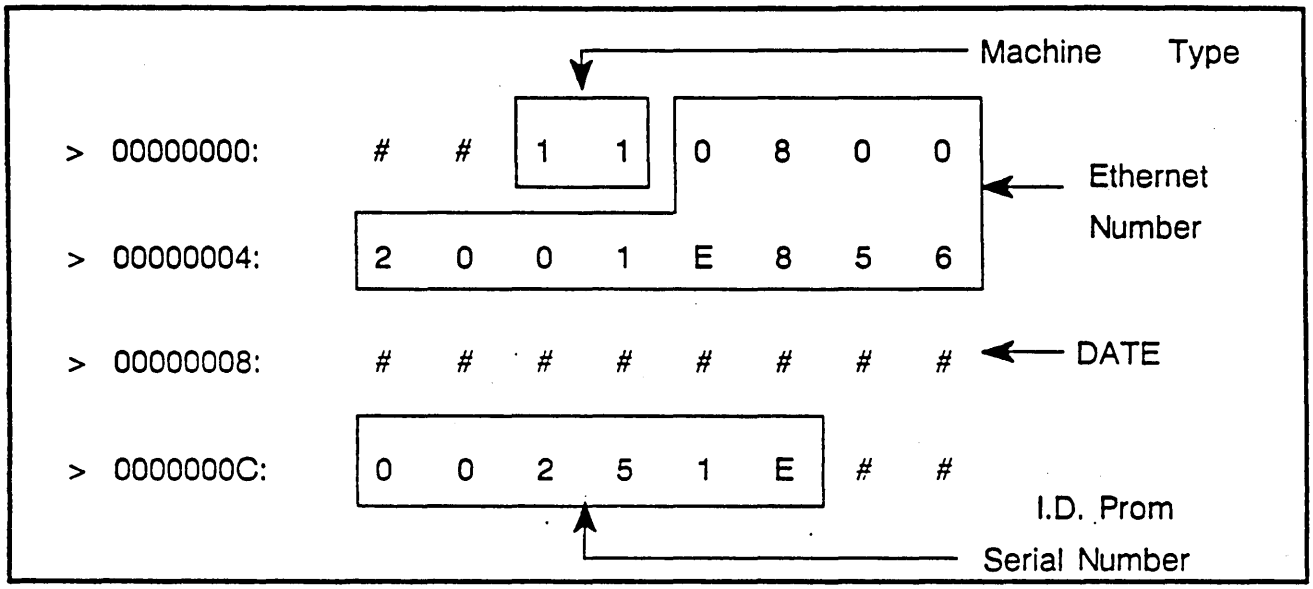
EXAMPLE:

```
#hostid  <cr>
1100251E      (11=carrera, 00251E=SN# of ID PROM)
```

Dumping the I.D. Prom

Address 0x00000000

```
>s3
>l 0  (lowercase "L", "zero", and <cr> 4 times)
```



```
> 00000000:   #  #  1  1  0  8  0  0
> 00000004:   2  0  0  1  E  8  5  6
> 00000008:   #  #  #  #  #  #  #  #
> 0000000C:   0  0  2  5  1  E  #  #
```

Code	Name	Family
11	Carrerra	3/75, 3/140, 3/150, 3/160, 3/180
12	Model 25	3/50
13	Sirius	3/260, 3/280
14	Prism	3/110
17		3/60
21	Sunrise	4/260, 4/280
22	Cobra	4/110
02		Sun2 VME
01		Sun2 Multibus

1.13 SYSTEM ENABLE REGISTER

The System Enable register enables the system facilities and allows the BOOT state. The register can be read or written to under software control. During a Power-On-Reset (POR) or Watchdog reset (see reset section), the system enable register is cleared. That is, when it is cleared all 8-bits are set to '0'.

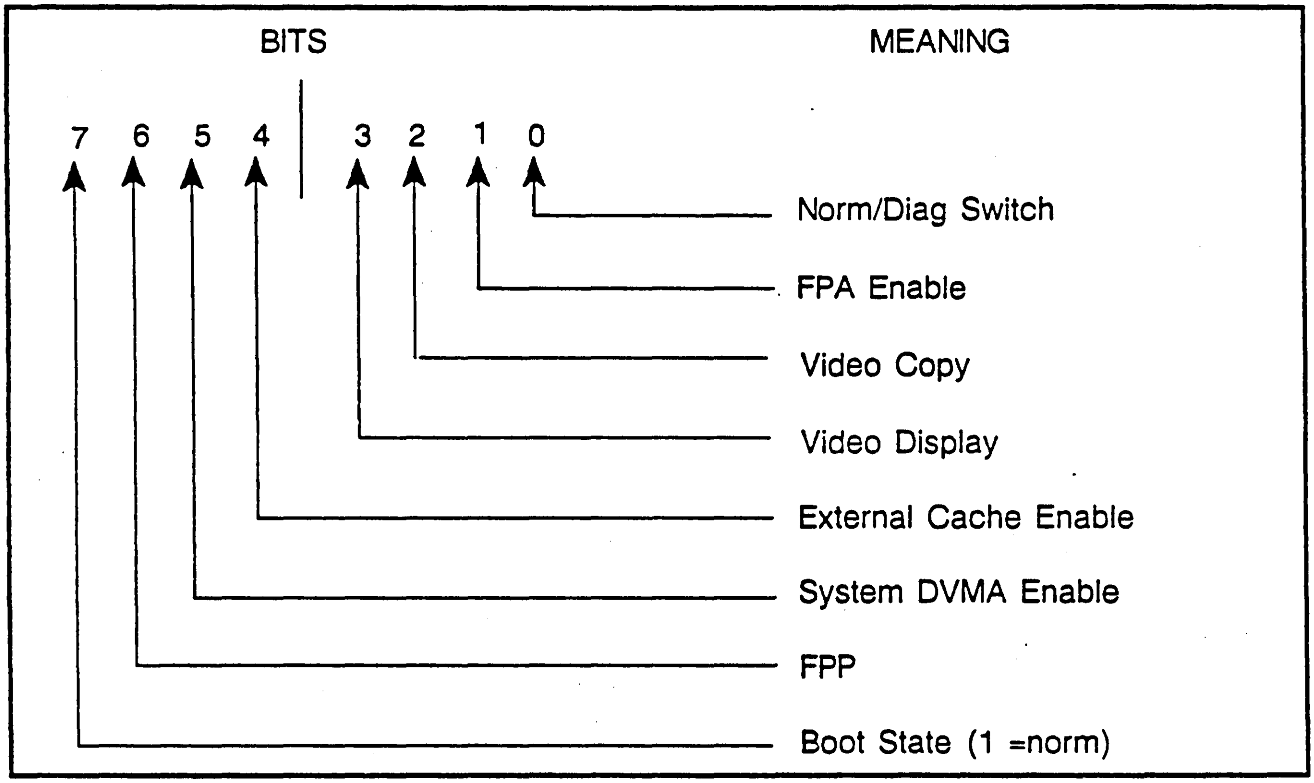
The system enable register sets up the bits that will monitor the state of reading back the diagnostic switch(see switch section), enabling the floating point coprocessor (68881) or FPA, enable the copy mode to video memory and the video display, and will also enable a system DVMA (when present). During a BOOT state, the system enable register is read to the state of the forced BOOT FETCH (i.e a normal state vs a boot state).

System Enable Register:

8 bit r/w data register

Address 0x40000000

```
>s3
>o 40000000
```



Data displayed is in the form of a two digit hex byte. To interpret the data, you must break each digit down into it's binary bits and identify which bits are set (1) and which are reset (0)

EXAMPLE: Register contains A9 = 10101001 binary Bits 7, 5, 3, and 0 set. This is a normal indication for a 3/160 with it's Norm/Diag Switch in Diag.

1.14 USER DVMA ENABLE REGISTER

On cycles which allow user DVMA this register controls which contexts will have DVMA access. Each context will have a separate enable bit provided. During a POR or Watchdog reset the User DVMA enable register bits are set to '0'. While in a reset state, all user DVMA will be disabled.

1.15 BUS ERROR REGISTER

The Bus Error register is used to latch a bus error which occurs synchronously to a CPU cycle. By latching the error as it occurs, the software can be used to identify the source of the bus error. The synchronous errors can be divided into two groups, those which occur and are identified by the register (when read), and those identified by the CPU itself- by the type of cycle it occurred on.

Examples of bus errors which occur during the cycle, and that are identified by the CPU include spurious interrupts (an error occurring during an Interrupt Acknowledge cycle) or an line instruction interrupt to the floating point processor. Both errors are identified solely by the type of cycle in which they occurred.

The Bus Error register will latch the cause of the pending bus error for the most recent case. If there are stacked bus errors, the data bit relating to the earlier error is lost. The Bus Error register is a read-only device which will monitor floating point enable errors or FPA bus error response, VME bus errors (a CPU cycle to the VME bus was acknowledged with a bus error), timeout errors (resulting from accessing non-existing or non-responding devices on the bus), protection errors(indicating that the page protection bits did not allow the kind of operation attempted during a CPU bus cycle to a valid page) and invalid page errors (indicating that a valid bit in the page map was not set).

1.16 DIAGNOSTIC REGISTER

The diagnostic register is used to drive the 8-bit LED display for monitoring error messages. During a cycle error, the insuing error will correspond to the LED display pattern. The diagnostic register will drive the LEDs to all ON during a power-on-reset condition and is a range of addresses beginning at 70,000,000. Refer to the diagnostic section of the manual for a description of the LED table.

Diagnostic Status Register

8 bit write only data register

Address 0x70000000

```
>s3
>o 70000000 5a (flashes 10100101)
>o 70000000 00 (flashes 11111111)
>f  70000000 7fffffff 00 (flashes 11111111
                            for extended period)
```

**NOTE* Register does not latch*

1.17 DEVICE SPACE ON THE MOS BUS

1.18 TIMER CHIP LOGIC

The timer device on the CPU card is an Intersil 7170 time-of-day clock with a battery back-up circuit. The timer crystal oscillates at 32.768KHz. The clock interrupt output causes an interrupt request at level 7 and level 5, via the interrupt register, when the respective levels are enabled.

1.19 INTERRUPT REGISTER

The interrupt register provides for the generation of software interrupts (from interval OS dependent cycles), controls the device hardware interrupts, and the clock interrupts on the CPU board. The register will read in interrupt enable levels for on-board devices.

On-board interrupts will be autovectored on all levels except for level 6. For level 6 (SCC's) the 8530 SCC devices will provide the vector. VME vectored interrupts will be vectored at a lower priority than On-board interrupts, and are set by jumpers on the CPU board.

The chart on the following page lists the interrupts by level and divides them into on board and off board groups for easy reference. Note that a level can have numerous devices, all of which may attempt to interrupt at the same time.

The processor uses the level and the AVEC signal to control interrupting devices and respond to the most urgent request.

CPU Interrupt Levels

Off Board	Level	On Board
	7	NMI & RTC
	6	SCC
	5	RTC (Unix)
ALM	4	Video Vertical Int.
B/W Graphics	4	
Color Graphics	4	
Graphics Processor	4	
1/2 inch tape	3	Ethernet
Second Ethernet	3	Sys enable (4)
Xylogics Disk	2	
SCSI	2	Sys enable (2)
Versatek	2	
	1	Sys enable (1)

FIGURE 1-10: Off board and On Board Interrupt Levels

By removing the jumpers on the CPU board you can isolate off board interrupt problems.

1.20 EEPROM

A EEPROM, implemented with a 2816 device, provides the CPU with 2K bytes of electrically erasable storage parameters. These parameters are primarily set up to provide information about the system, the display (which frame buffer will be defaulted to) and SIO ports (console information), and buffer information describing how the self-test will be accomplished (exclude memory test default). Chapter Five will detail the use and set-up of the EEPROM.

1.21 BOOT PROM

The Boot PROM is a 27512 EPROM providing around 64K bytes of bootable code to initialize the system. This read-only device resides on the MOS I/O bus, and is addressed directly with virtual addresses from the processor. The BOOT PROM is accessed during the FORCED BOOT state. While in this state, all supervisor program fetches are forced to fetch from this device, independent of the setting of the MMU. A NORM/DIAG switch on the boards rear edge panel is used to select the normal boot code, or diagnostic code, from the BOOT PROM device while at the MONITOR level.

1.22 SERIAL COMMUNICATION CONTROLLERS

The CPU card provides two Serial Communication Controllers (SCC) for SIO port and keyboard/mouse control (1200 baud). These controllers behave like UARTS, but are programmable. The SCC's are implemented with Zilog 8530's, featuring two high speed, fully symmetrical and programmable serial channels. Both SCC's on the CPU card have built-in baud rate generators.

The SCC controlling the SIO ports uses its channel A for the UART A port, and channel B for UART B port. Both ports are set up for EIA RS 423 specifications, and operate from an independent CPU clock rate at 4.9152MHz. The SCC controlling the keyboard/mouse uses its channel A for the keyboard and channel B for the mouse. The RS 423 devices are signal compatible with RS 232 C, CCITV.24 but limit voltage to +/- 12VDC (refer to EIA specification).

1.23 ETHERNET INTERFACE

The Ethernet interface and control is implemented with an Intel 82586 Ethernet Data Link Controller. Ethernet accesses the top 16 M bytes of the current virtual address space, with the supervisor data function code. The Ethernet interface uses the TYPE 0 space (memory) only, as a DVMA master.

Bus errors can occur during an Ethernet transfer as a result of a protection error or a memory error (parity) during a read operation. If an Ethernet bus cycle concludes with a bus error, the error bit in the Ethernet control register is set and further activity is stopped. Ethernet will reset during an aborted DVMA cycle when a bus error is pending.

Like Sun 2 architecture, the 82586 is physically byte swapped on-board the CPU card. This ensures that the Ethernet data is stored in memory in 68020 byte-order.

1.24 VME BUS INTERFACE

The VME bus interface is dual-ported. The VME bus master interface provides access from the CPU to the VME bus, and the VME slave interface provides access from the VME bus to the CPU.

A VME master is a device which can initiate a transaction over the VME bus. Such as the CPU or the SMD disk controller when they want to store data directly into memory. A VME slave simply responds to such transactions. The CPU board will act as the arbitor and default master.

To meet the VME specification, Sun 3 architecture supports a 32-bit master or slave address bus, (This is the Physical address bus out of the MMU) a 32-bit master or slave data bus, has a minimum timeout period of 739 usec (excluding bus acquisition), supports single level arbitration and a 'release-on-request (ROR)' option, as well as providing interrupt handling.

The VME master interface uses two page map type codes including one for 16-bit data (TYPE 2), and the other for 32-bit data (TYPE 3).

1.13 CPU RESETS

There are 5 possible reset sources for Sun 3 architecture. These include POR (active for >100 msec after the power reaches a 4.5 VDC level), VME bus reset (jumperable, when the CPU board is not the arbitor), Watchdog reset (when the CPU halts due to double bus faults), a user reset (initiated via the external reset switch), and a CPU reset (resulting from the CPU executing a RESET instruction).

The power-on-reset (POR) is illustrated below to indicate to the reader what conditions must be met internally to the 68020, before it is stable enough to execute its first instruction fetch

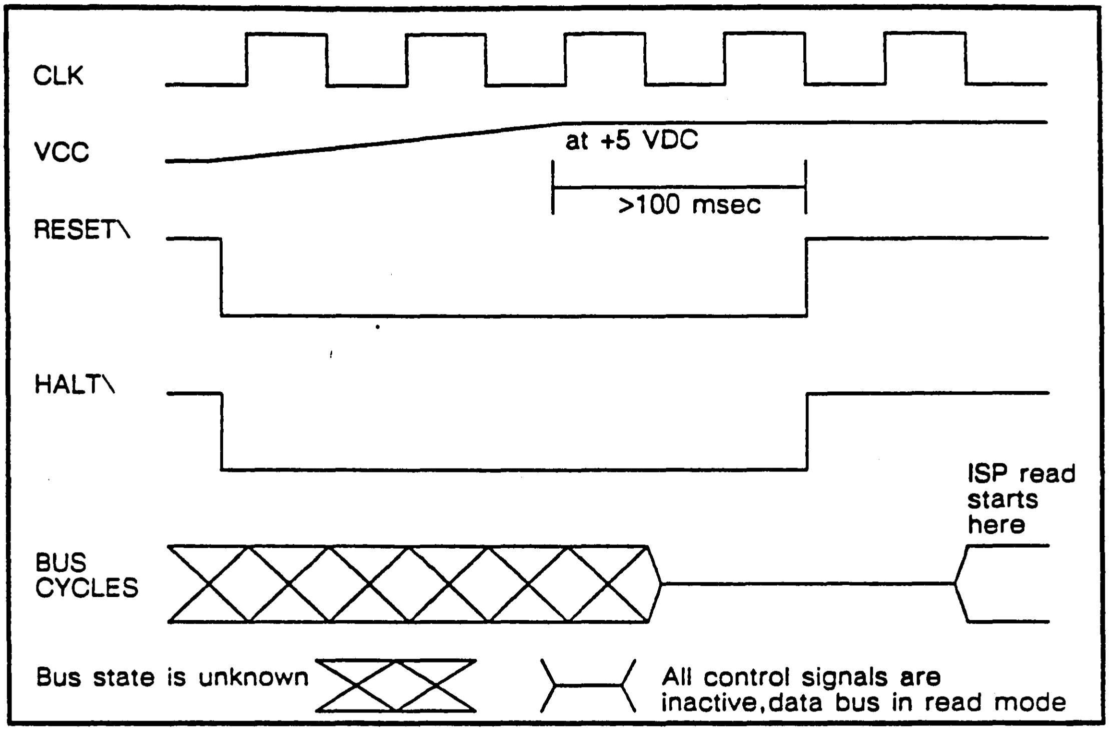


FIGURE 1-11: POWER-ON-RESET(POR) ACTIVITY

When the processor is reset by the external reset circuit (POR or user reset switch) the reset is recognized as a system reset. The processor will attempt to complete the current bus cycle, and will then reset the vector table entry (two words) and load it into the Interrupt Stack Pointer (ISP). This will set the status register to a level 7 (NMI) interrupt mask.

1.27 SYSTEM SUMMARY SPECIFICATIONS

The following table.will give the user a quick reference guide to the system summary specification. Current versions of the Sun 3 product covered include the Sun 3/75M 'diskless node' workstation, the Sun 3/160M monochromatic-based pedestal, and the Sun 3/160C color workstation, the 3/180 file server, and the low cost 3/50- single board system, the Sun 3/110 low cost color workstation, the Sun 3/140 -3 slot version with the Sun 3 CPU, and the Sun 3/200 high end 25 MHz workstation with high resolution display video. See next page.

DEVICE DESCRIPTION		DEVICE SPECIFICATION		
Main Processor		68020 operating at 15 Mhz to 25 Mhz		
Floating Point Processor		68881 operating at 12.5 to 20.00 MHz		
Memory Management				
	Type	Sun 3 MMU		
	Virtual Memory	256 M Bytes per process		
	Contexts	8		
	I/O Interface	DVMA		
Main Memory				
	DRAM	4 or 8 MB on CPU configurations		
	Expansion DRAM	4 or 8 MB on mem exp configurations		
	Maximum DRAM	4 up to 32 MB(depending on system type)		
	Cycle Time	270 nsec		
	Data Bus Bandwidth	32-bit		
	Error Detection	Byte Parity or ECC(ECC on 3/200 only)		
Ethernet Interface				
	Media	Coax Cable		
	Data Rate	10 Mbits/sec		
Display				
	Format	19-inch Landscape		
	Aspect Ratio(monitor)	4:3		
	Resolution	1152 x 900 or 1600 x 1280(1 or 8-bit deep)		
	Pixel Aspect Ratio	1:1		
	If Color(colors displayed)	256		
	If Color(display palette)	16.7 million		
	Refresh Rate	66 Hz non-interlaced		
	Frame Buffer-Color	1 or 2 MB(double buffer option)		
	Frame Buffer-Monochrome	128 KB or 256 KB(256 KB on 3/200 or 3/60 only)		
	Bandwidth	Color=93 MHz Mono=100 or 200 MHz		
	Controls	Both have brightness,color has degaussing		
Keyboard		94-key, low profile		
Mouse		3-button Optical		
SIO'S		2 RS 423 serial Options include 8/16 channel control or Options include 4 channel high-speed for SunLink software		
Disk Options		SCSI	SMD	ESMD
	Formatted Capacity	71 MB	130 MB	380 MB
	Unformatted Capacity	85 MB	168 MB	474 MB
	Average Time Access	35 msec	20 msec	18 msec
Disk Options	Data Rate	.5 MB/sec	2.4 MB/s	1.85MB/sec
	Formatted Capacity	141MB	280MB	575MB
	Unformatted Capacity	170MB	337MB	689MB
	Average Time Access	23msec	20msec	18msec
Disk Options	Data Rate	1.2MB/sec	2.4MBsec	2.45MB/sec
	Formatted Capacity	327MB		892 MB
	Unformatted Capacity	380MB		1067 MB
	Average Tims Access	18msec		15 msec
	Data Rate	1.2MB/sec		2.45 MB/sec
Tape Options				
	1/4-inch	60 MB capacity		
	1/2-inch	6250 bpi, >100 MB capacity		
Electrical				
	AC Voltage	90-132 VAC or 180-264 VAC		
	AC Frequency	47.5-66 Hz		
	Power	150-325 W(desktop) 850 W pedestal/server		

FIGURE 1-12: SYSTEM SUMMARY SPECIFICATIONS

CHAPTER TWO

SYSTEM BUS OPERATIONS

OVERVIEW

This section will describe how bus operations transact on the Sun 3 system between the 68020, the 68881 floating point coprocessor, the on-board I/O control logic, and the VME bus extensions to memory and other peripheral controller boards operating within the Sun 3 environment (while operating under the Monitor, under standalone diagnostic programs, and under system software control).

The first half of this chapter will describe board cycles from an execution standpoint with the 68020, and the second half will describe how the system executes a specified cycle over the VME bus.

The purpose of this section is to give the engineer and technician an overall picture of how the CPU will accomplish transactions during successful and unsuccessful operations. Understanding normal, or rather typical operations (as specified by 68020 and system software operations), the system user should be able to determine how and when things are operating successfully under typical cycle execution, and will be able to use this knowledge of the systems architecture to follow a logical path to troubleshooting the Sun workstation, if a specified operation is not successful.

The next chapter, Chapter THREE, will cover the configuration elements of the system and will be followed by Chapter FOUR, covering the diagnosis and troubleshooting of system problems.

2.1 68020 BUS OPERATIONS AND TRANSFER MECHANISMS

In this section we will discuss transactions and cycle executions at the 68020 level. This overview of processor cycle handling is meant to give the user an idea of how the host processor deals with control and bus operations during a data transaction, and how the 68020 will interface with other devices during bus arbitration, bus error handling and halt conditions, and during the various types of reset operations.

The 68020 will be operating at system clock speed (15-25Mhz). The operand size of the 68020 will include byte (8bits), word (16), long word (32) and extended word transfers (64bits Model 200's). To control for this, the 68020 supports a dynamic sizing and transfer mechanism which allows the processor to monitor the size of the current data cycle it is executing with a mapped device.

If the 68020 is attempting to execute an instruction which involves a read of a long word (32-bit) for example, the processor will attempt to read all 32-bits within the first bus cycle. For normal read cycles with on-board devices, or with the VME bus, this mechanism handles fixed size assignments with the devices that responded to the transaction with the 68020.

This dynamic bus sizing capability allows the host processor to respond to the bus divisions pointed out in Chapter ONE. The main bus, as discussed in Chapter ONE, has a bus width of 32 bits that interfaces the 68020 to the 68881, the VME interface, the Ethernet Data Link Controller, the Video Frame buffer and the main memory and expansion memory. (Model 200's have 64bit bus)

All data cycles are handled via a transfer handshake with the 68020 and MMU. The Data Sizing and Transfer Acknowledge (DSACK) signal is used to acknowledge the completion of a data transfer. All data transactions must terminate with DSACK. During a data transfer, if a device on-board or on the VME bus specified a longword transfer, the 68020 latches all 32-bits, the DSACK signal will be monitored, and the system will allow the next operation. If the cycle does not complete successfully, a bus error is generated, and the system software can read back the source of the error from the 'bus error register'. This register will be read back synchronously to the cycle that the error occurred on.

There are also two on-board byte buses supported under Sun 3 architecture. The MOS and TTL buses are 8-bit buses. All transactions with these devices will follow the same rules for sizing as the 32-bit bus, but will be specified to the 68020 as to the size of their transfer in bytes.

2.2 READ BUS OPERATIONS

During a read cycle with the 68020, the processor will determine the operand size via the operand transfer mechanism. The address bus, data bus and control bus will be used to implement the cycle, and the host processor will be responsible for de-skewing the transfer acknowledge and data signals from the slave device. The following flowchart details how a successful read cycle (without a bus error) would execute with the 68020 and an on-board I/O device, or any bus master and memory.

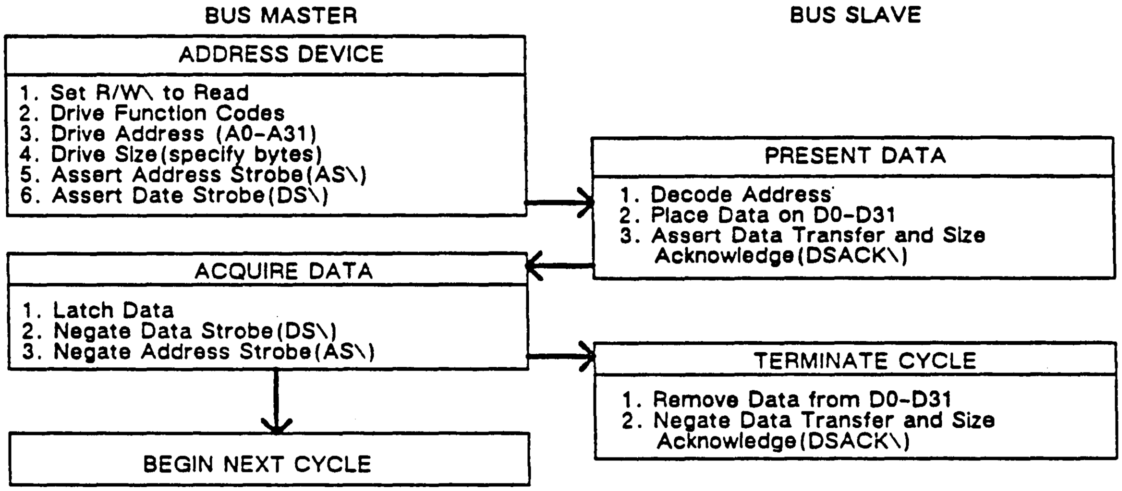

BUS MASTER
BUS SLAVE
ADDRESS DEVICE
1. Set R/W\ to Read
2. Drive Function Codes
3. Drive Address (A0-A31)
4. Drive Size(specify bytes)
5. Assert Address Strobe(AS\)
6. Assert Date Strobe(DS\)
PRESENT DATA
1. Decode Address
2. Place Data on D0-D31
3. Assert Data Transfer and Size Acknowledge(DSACK\)
ACQUIRE DATA
1. Latch Data
2. Negate Data Strobe(DS\)
3. Negate Address Strobe(AS\)
TERMINATE CYCLE
1. Remove Data from D0-D31
2. Negate Data Transfer and Size Acknowledge(DSACK\)
BEGIN NEXT CYCLE

FIGURE 2-1: READ CYCLE FLOWCHART

2.3 WRITE BUS OPERATION

When the 68020 executes a write operation, the processor sends data to memory or the chosen device (on-board device, or VME device) for the transaction. To specify the size of the transfer, the 68020 utilizes the operand transfer mechanism the same way as it did in the read cycle. The following flowchart is used to illustrate this point.

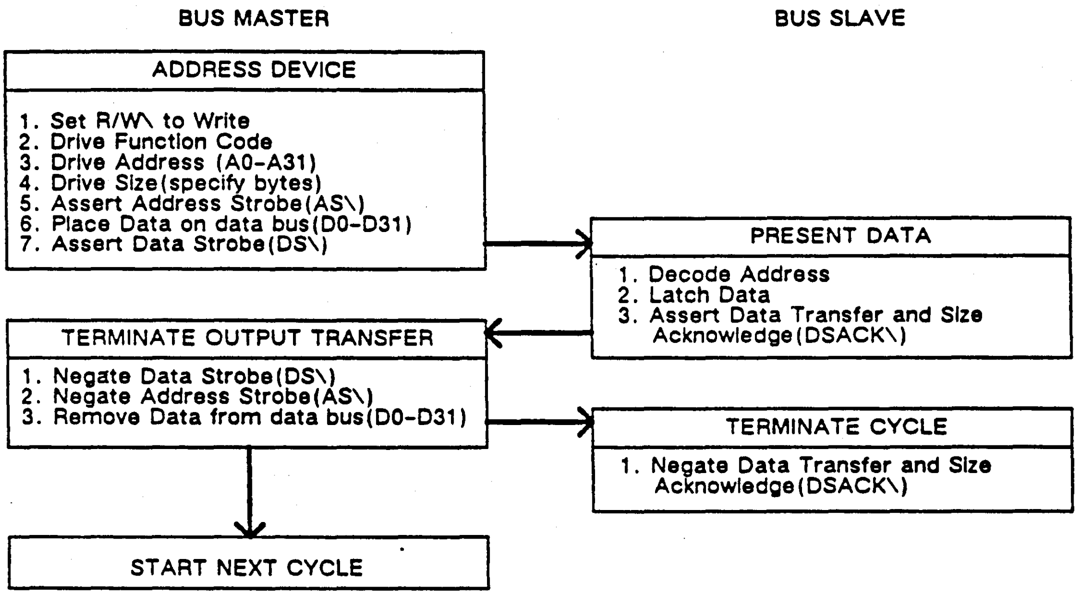

BUS MASTER
BUS SLAVE
ADDRESS DEVICE
1. Set R/W\ to Write
2. Drive Function Code
3. Drive Address (A0-A31)
4. Drive Size(specify bytes)
5. Assert Address Strobe(AS\)
6. Place Data on data bus(D0-D31)
7. Assert Data Strobe(DS\)
PRESENT DATA
1. Decode Address
2. Latch Data
3. Assert Data Transfer and Size Acknowledge(DSACK\)
TERMINATE OUTPUT TRANSFER
1. Negate Data Strobe(DS\)
2. Negate Address Strobe(AS\)
3. Remove Data from data bus(D0-D31)
TERMINATE CYCLE
1. Negate Data Transfer and Size Acknowledge(DSACK\)
START NEXT CYCLE

FIGURE 2-2: WRITE CYCLE FLOWCHART

2.4 BUS ERROR OPERATIONS AND BER REGISTER

When a Read or Write cycle does not execute successfully (when a bus error occurs synchronously to a CPU bus cycle), Sun 3 architecture implements a mechanism to report the cause of the bus error. The BER register will latch the cause of the unsuccessful cycle attempt, to allow software to identify the source of the bus error.

Thus, the operating system or a stand-alone-diagnostic utility has a mechanism that can be used to report a pending bus error to the system user. The error is usually reported on the output device. In most cases, this would be the Sun display or available ASCII terminal.

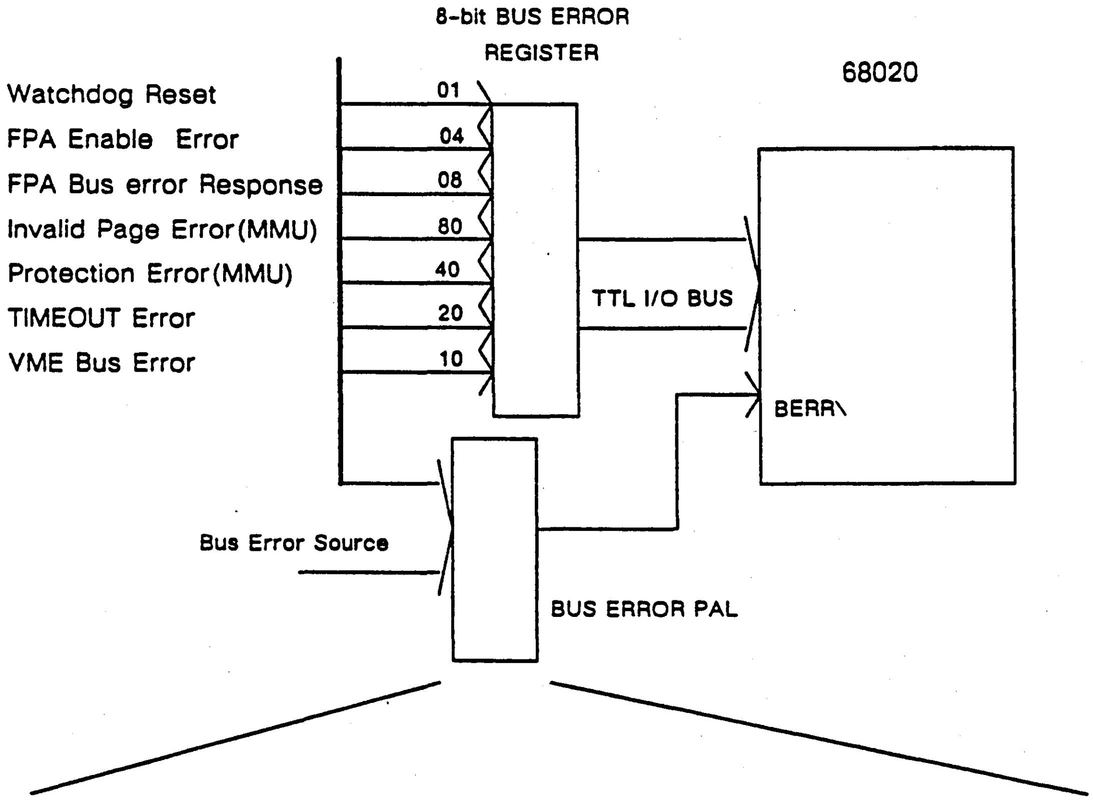
As indicated in Chapter ONE, there were two types of error report. The first was an error reported via the BER register, and the second specified by the type of cycle currently being executed by the processor.

Sun 3 architecture requires a handshake from the responding device in order to successfully complete a cycle. In addition, due to the fact that some of these devices will use different response times, a mechanism has been included to monitor the duration of the handshake-that is, the timeout logic. The timeout of a current cycle will also report a bus error to the BER register, as well as accessing a non-existing device on both the on-board and VME bus.

When a bus error is recognized, the current cycle is terminated, and (in most cases) the error reports on the display screen. The following chart illustrates how the 68020 handles and reports pending bus errors. Note that Chapter FOUR will detail how the system user might resolve bus errors that would necessitate further user interaction.

Dumping the BER

```
>s3
>o 60000000    { address of BER }
```



Hex bit in
Register

01 - An exception occured durring exception processing such as a double bus fault or the user reset switch was pressed.

04 - An access was attempted to the Floating Point Accelerator when it was disabled.

08 - The FPA detected a P2 bus error durring access. The FPA does not have interupt capabilities and must report ANY errors he detects via this line.

80 - The virtual memory address sent to the MMU pointed to a Page entry that did not have the valid bit set.

40 - The type access to memory attempted is denied. (Write Protected or Supervisor only).

20 - An access was attempted to a non-existing device or the device did not respond in time.

10 - A user of the VME bus has detected an error on the bus.

Figure 2-3: Bus Error Report

When a bus error is asserted to the processor by the bus error PAL from one of the above listed sources, the 68020 will save its supervisor state and image of the instruction of the current cycle that the error occurred on (address, function code,size), and halt the cycle. Then, if a bus error and halt is pending during the bus cycle, the processor usually terminates the cycle, but can enter a retry sequence. The actual timing sequence will go from the detection of a READ bus error, to an internal processing state, where an exception is taken and the stack is written. It should be noted, however, that when both a bus error and a halt is asserted the processor will typically terminate the bus cycle, while placing the control signals into their inactive state. The negation of these two control lines will then have to be accomplished by an external logic source under software control.

The condition of the bus error will now be written to a buffer (under software control) and in most cases will be copied or echoed to the display (via the video frame buffer pathway). If the bus error is set and the cycle retry fails, this may cause the 68020 to abort, causing a reboot of the system or a return to the monitor mode (if the reboot fails). The system software will typically try to clean up the faulted cycle, but in some cases will not be able to.

It should also be noted, that if a halt is pending by itself, the 68020 will typically run a halt-run-single step function. If the halt is asserted when the processor begins another cycle, it will remain asserted and that cycle will be allowed to complete but new cycles will not be allowed to start. The halt will most likely be generated via the Bus Error PAL. Bus halting will not effect arbitration, so that when a processor halt is pending, bus arbitration can proceed as normal.

2.5 READ-MODIFY-WRITE BUS CYCLES

The 68020, under Sun 3 architecture, supports Read-Modify-Write (RMC) cycles. Under this condition, the 68020 performs a READ, modifies the data, and WRITES back to the same address. During the entire RMC, a processor read-modify-write signal will be asserted to the arbitration state control PAL, and will cause the VME bus to be locked out. The lock-out is controlled by not asserting a bus grant out to any board arbitrating for the VME bus, until the RMC cycle has completed. This way the processor has a means of implementing a secure cycle. Read-modify-write cycles (RMC) provide support for intertask or interprocessor transactions. The following flowchart illustrates how this is accomplished.

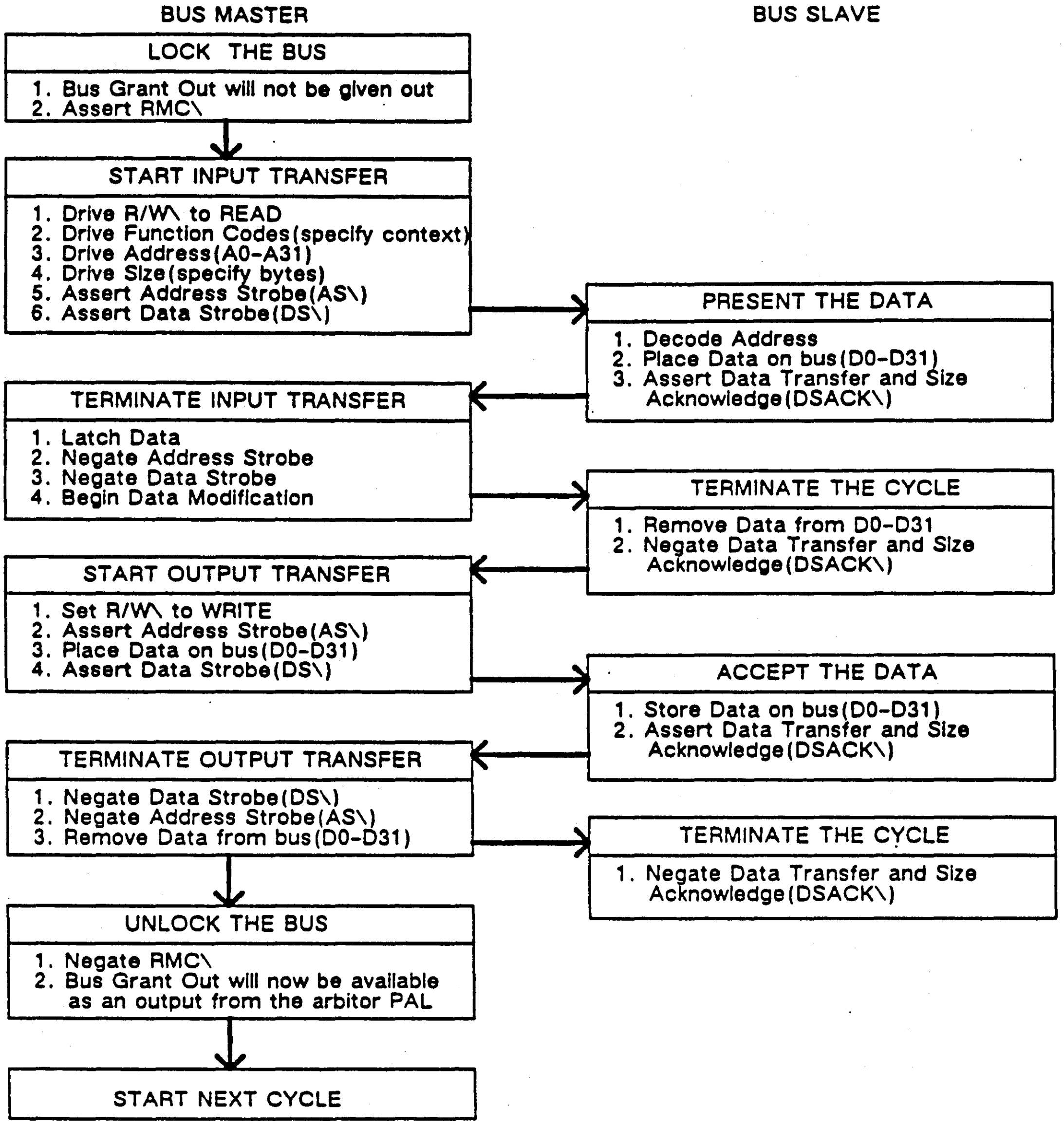

BUS MASTER
BUS SLAVE
LOCK THE BUS
1. Bus Grant Out will not be given out
2. Assert RMC\
START INPUT TRANSFER
1. Drive R/W\ to READ
2. Drive Function Codes(specify context)
3. Drive Address(A0-A31)
4. Drive Size(specify bytes)
5. Assert Address Strobe(AS\)
6. Assert Data Strobe(DS\)
PRESENT THE DATA
1. Decode Address
2. Place Data on bus(D0-D31)
3. Assert Data Transfer and Size Acknowledge(DSACK\)
TERMINATE INPUT TRANSFER
1. Latch Data
2. Negate Address Strobe
3. Negate Data Strobe
4. Begin Data Modification
TERMINATE THE CYCLE
1. Remove Data from D0-D31
2. Negate Data Transfer and Size Acknowledge(DSACK\)
START OUTPUT TRANSFER
1. Set R/W\ to WRITE
2. Assert Address Strobe(AS\)
3. Place Data on bus(D0-D31)
4. Assert Data Strobe(DS\)
ACCEPT THE DATA
1. Store Data on bus(D0-D31)
2. Assert Data Transfer and Size Acknowledge(DSACK\)
TERMINATE OUTPUT TRANSFER
1. Negate Data Strobe(DS\)
2. Negate Address Strobe(AS\)
3. Remove Data from bus(D0-D31)
TERMINATE THE CYCLE
1. Negate Data Transfer and Size Acknowledge(DSACK\)
UNLOCK THE BUS
1. Negate RMC\
2. Bus Grant Out will now be available as an output from the arbitor PAL
START NEXT CYCLE

FIGURE 2-4: READ-MODIFY-WRITE BUS CYCLE FLOWCHART

RMC cycles provide a means for compare and swap (CAS) operations with the data registers, as well as testing and setting of an operand (TAS). An instruction field (data addressing) will specify the location of the tested operand.

2.6 CPU SPACE CYCLES

Within Sun 3 Architecture the CPU Space Cycles will include all cycles that execute with Function Code 7 level (indicated to the MMU). These cycles will include coprocessor (68881) and interrupt cycle handling.

Coprocessor cycles will be indicated when specified address lines are decoded and reference is directed to the 68881 floating point coprocessor. The coprocessor can terminate with data transfer and size acknowledge, or can terminate with a pending bus error reported synchronously to the cycle.

Interrupt acknowledge cycles will complete with data transfer and size acknowledge, and will terminate with the assertion of an 'autovectored' strobe for most onboard interrupts. The interrupt cycle will complete normally unless the acquisition is aborted with a pending bus error report. The following sections details how coprocessor and interrupt cycles are handled by the CPU.

2.7 68881 FLOATING POINT COPROCESSOR CYCLES

The implementation of the 68881 floating point coprocessor on the Sun 3 board is designed to follow the protocol of the 68020 host processor. The 68020/68881 interface is designed as such so that any instruction instituting a transaction between the two processors, is made to appear to be residing on the internal stack of the 68020. The 68881 interface conforms to the IEEE 754 interface specification.

Sun 3 architecture bases this interface on asynchronous cycles, and therefore the 68881 can operate at 12.5 MHz. This way, transfers can be accomplished without special control signals since both processors reside on a common 32-bit bus outside of MMU control.

The 68881 floating point coprocessor uses binary floating point arithmetic, and features eight floating point data registers which support 80-bit precision real-data format (64-bit mantissa plus sign bit and 15-bit biased exponent). The coprocessor will also support byte, word, and long word data transfers.

Additional features includes a 32-bit control register (for modes), 32-bit status register (containing condition codes), and a 32-bit instruction address register containing the host processor memory address of the last floating point instruction that was executed. This register can be accessed for handling and locating the instruction that caused the exception on a current cycle.

The following blocks illustrate the 68881 functional signal groups, and its interface with the 68020 host processor.

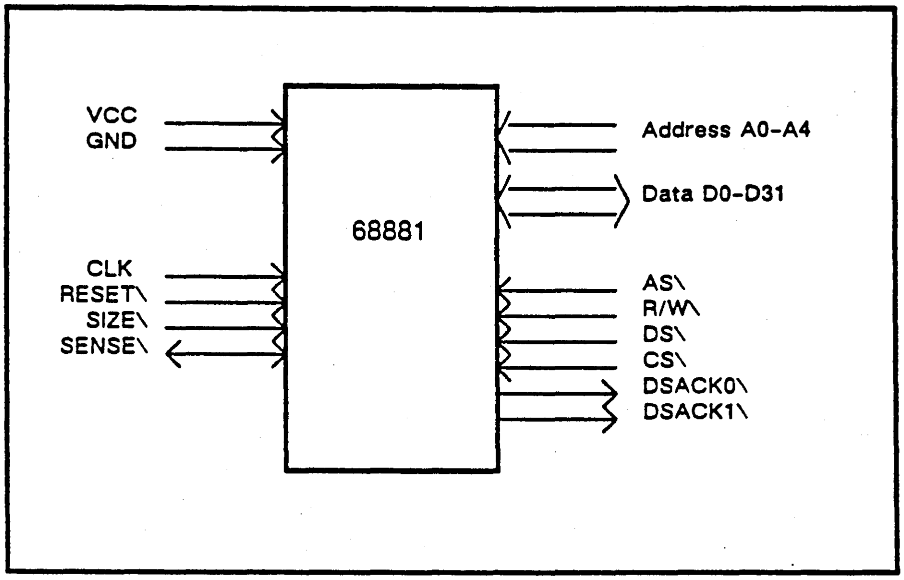


FIGURE 2-5: 68881 SIGNAL GROUPS

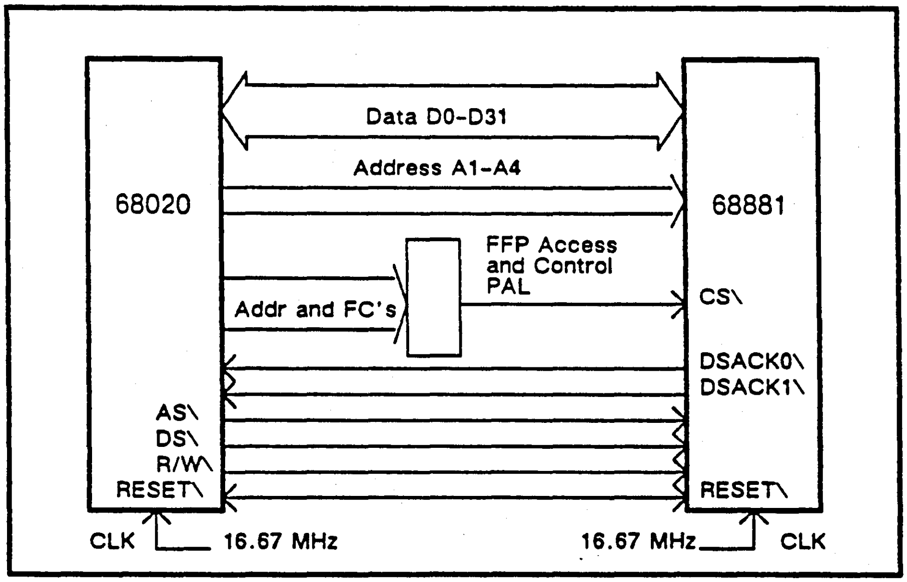


FIGURE 2-6: 68020/68881 INTERFACE

The interface protocol for general category instructions between the 68020 and the 68881 are defined by the 68020. To execute a general instruction, the host processor initiates the transaction with the floating point coprocessor by writing an instruction command to the coprocessors interface register (CIR). The coprocessor, in turn, will decode the command and execute it with the next instruction. The following flowchart illustrates this interface.

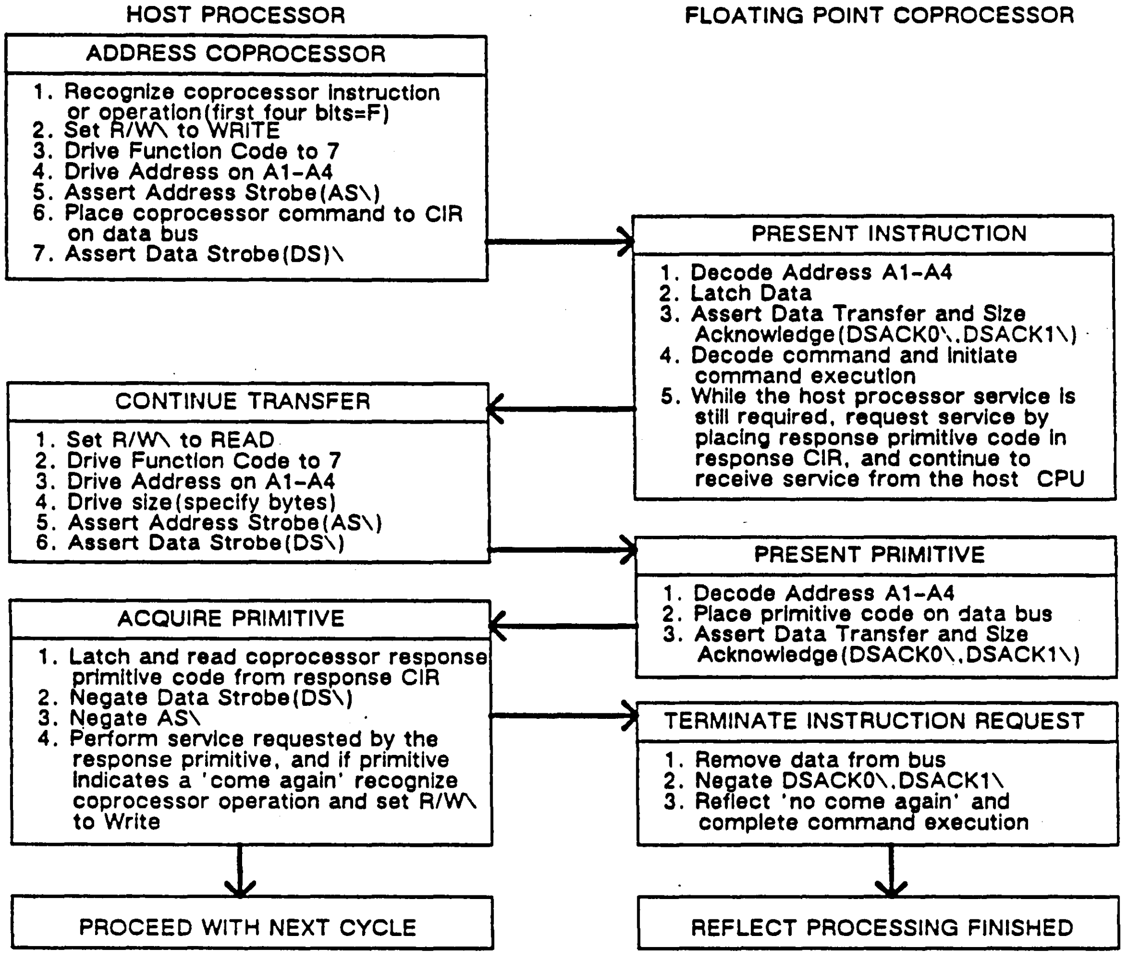

HOST PROCESSOR
FLOATING POINT COPROCESSOR
ADDRESS COPROCESSOR
1. Recognize coprocessor instruction or operation(first four bits=F)
2. Set R/W\ to WRITE
3. Drive Function Code to 7
4. Drive Address on A1-A4
5. Assert Address Strobe(AS\)
6. Place coprocessor command to CIR on data bus
7. Assert Data Strobe(DS)\
PRESENT INSTRUCTION
1. Decode Address A1-A4
2. Latch Data
3. Assert Data Transfer and Size Acknowledge(DSACK0\.DSACK1\)
4. Decode command and initiate command execution
5. While the host processor service is still required, request service by placing response primitive code in response CIR, and continue to receive service from the host CPU
CONTINUE TRANSFER
1. Set R/W\ to READ
2. Drive Function Code to 7
3. Drive Address on A1-A4
4. Drive size(specify bytes)
5. Assert Address Strobe(AS\)
6. Assert Data Strobe(DS\)
PRESENT PRIMITIVE
1. Decode Address A1-A4
2. Place primitive code on data bus
3. Assert Data Transfer and Size Acknowledge(DSACK0\.DSACK1\)
ACQUIRE PRIMITIVE
1. Latch and read coprocessor response primitive code from response CIR
2. Negate Data Strobe(DS\)
3. Negate AS\
4. Perform service requested by the response primitive, and if primitive indicates a 'come again' recognize coprocessor operation and set R/W\ to Write
TERMINATE INSTRUCTION REQUEST
1. Remove data from bus
2. Negate DSACK0\.DSACK1\
3. Reflect 'no come again' and complete command execution
PROCEED WITH NEXT CYCLE
REFLECT PROCESSING FINISHED

FIGURE 2-7: COPROCESSOR PROTOCOL FOR GENERAL INSTRUCTIONS

2.8 INTERRUPT BUS OPERATIONS

Sun 3 architecture supports 7 levels of prioritized interrupts. Figure 1-10 illustrated the level priority of most interrupts.

Interrupted recognition on the Sun 3 CPU board is processed by the state of the Interrupt Priority Level lines (IPL0-2) input to the 68020 from the interrupt priority encoder and interrupt register, and is set up in the interrupt priority mask of the 68020 status register. The lowest level mask is level zero (indicating no pending interrupts), and the highest level is seven(indicating a non-maskable interrupt is pending). On the VME bus, level 7 would handle the Time-of-Day clock and any pending parity errors during start-up.

When level one through six is requested, the processor will compare the interrupt level to the current interrupt mask. The processor will then be able to determine if the new interrupt will be serviced. If the new request is lower to the current interrupt mask, the processor will ignore it. If the incoming interrupt level is recognized, however, as being higher than the current mask, then the pending interrupt will be serviced at the completion of the current cycle. It should be noted, that the interrupt priority level lines must be held at their present level until the processor acknowledges their presence.

2.9 VME INTERRUPT CYCLES

VME interrupt acknowledge cycles differ from normal VME read/write cycles. During a VME interrupt acknowledge cycle the master drives P1 addresses A1 through A3, while asserting an interrupt acknowledge and deasserting P1 write and the address and data strobes. Starting at card cage slot #1, each VME card determines if they are the source of the interrupt at the level specified by addresses A1 through A3. If the VME card is the source, an 8-bit vector address is driven onto the data bus along with the assertion of P1 data transfer acknowledge. If the card is not the source of the interrupt, P1 interrupt acknowledge out will pass to the next physical slot. This VME card will accept the signal as P1 interrupt acknowledge in, and continue the chain.

2.10 INTERRUPT ACKNOWLEDGE AND PROCESSING

Interrupt acknowledge processing is initiated when there is a pending interrupt (at an instruction boundary) and the host processor begins its respective cycle. When initiating the interrupt acknowledge sequence (providing there are no higher level exceptions pending), the host processor will first check the starting location of the interrupt service. This location must correspond to the requested service location.

There are two types of interrupt or exception processing cycles the 68020 can follow. The first comes from the on-board I/O devices (higher priority level), and the second type are established from the VME bus (lower priority).

The host processor also supports the acquisition of data during the interrupt acknowledge cycle. This data will come from an internal vector table, or will be provided over the on-board bus or the VME bus from the requesting device.

The table comes from the vector base register, which points to the base of the 1K byte exception vector table. This table will contain the 256 exception vectors which are used as memory pointers by the 68020 to acquire the address routine that will generate the various exceptions. These exception vectors will be one word in length. Note that the reset vector is 2 words (see RESET section).

The 68020 must be in the supervisor privilege state in order to execute the exception sequence. Once in this state (supervisor data), the 68020 starts its interrupt acknowledge daisy-chain (also see the section on BACKPLANE CONFIGURATION), the processor will set and drive the appropriate Function Codes (see MMU section), set and drive specified addresses, echo the interrupt level being acknowledged on the lower order address lines (A1-A3) and pass the Interrupt Acknowledge IN/OUT daisy chain out to the requesting device. The following flowchart illustrates this process.

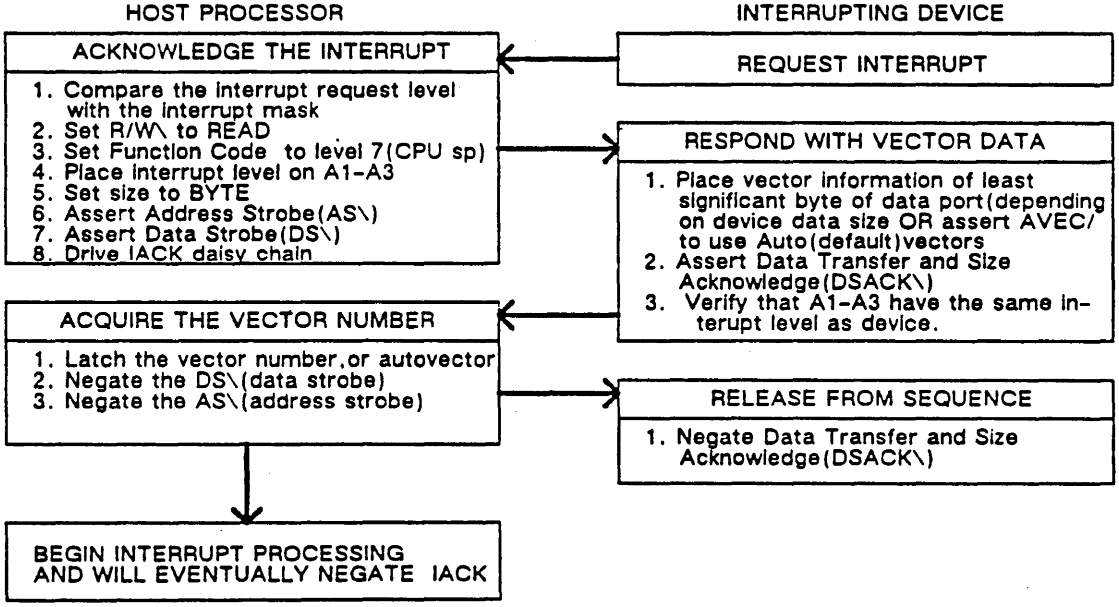

HOST PROCESSOR
INTERRUPTING DEVICE
ACKNOWLEDGE THE INTERRUPT
1. Compare the interrupt request level with the interrupt mask
2. Set R/W\ to READ
3. Set Function Code to level 7(CPU sp)
4. Place interrupt level on A1-A3
5. Set size to BYTE
6. Assert Address Strobe(AS\)
7. Assert Data Strobe(DS\)
8. Drive IACK daisy chain
REQUEST INTERRUPT
RESPOND WITH VECTOR DATA
1. Place vector information of least significant byte of data port(depending on device data size OR assert AVEC/ to use Auto(default)vectors
2. Assert Data Transfer and Size Acknowledge(DSACK\)
3. Verify that A1-A3 have the same interupt level as device.
ACQUIRE THE VECTOR NUMBER
1. Latch the vector number.or autovector
2. Negate the DS\(data strobe)
3. Negate the AS\(address strobe)
RELEASE FROM SEQUENCE
1. Negate Data Transfer and Size Acknowledge(DSACK\)
BEGIN INTERRUPT PROCESSING AND WILL EVENTUALLY NEGATE IACK

FIGURE 2-8: INTERRUPT ACKNOWLEDGE FLOWCHART

In order for the 68020 to drive an off-board Interrupt Acknowledge daisy-chain (on the VME bus), jumpers must be set on the backplane. These jumpers pass Interrupt Acknowledge IN to Interrupt Acknowledge OUT (IACK) across the backplane to the requesting board. Refer to Figure 2-9 for an illustration of the interrupt logic.

As a rule of thumb, any slot which remains EMPTY during the card configuration phase of setting-up your system-that is, there is an empty slot between the CPU card and the card it is doing the transaction with, jumper shunts must be installed in Jxx3 and Jxx4.

Note that Jxx3 establishes the Bus Grant3 In-to-Bus Grant3 Out daisy chain, and Jxx4 is for the Interrupt Acknowledge In-to-Interrupt Acknowledge-Out daisy chain. These jumpers are on the front of the backplane, on the solder side and are accessible by dropping the power supply and front panel. Also refer to the configuration section of this manual (*Chp THREE*).

It is very important to daisy-chain the IACK signals across the backplane. Any miss jumpering will result in a spurious interrupt, as explained in the next section.

IACK and BG3 jumpers must be in place for empty slots, the FPA and memory boards. All other slots that have boards in them should have the jumpers removed.

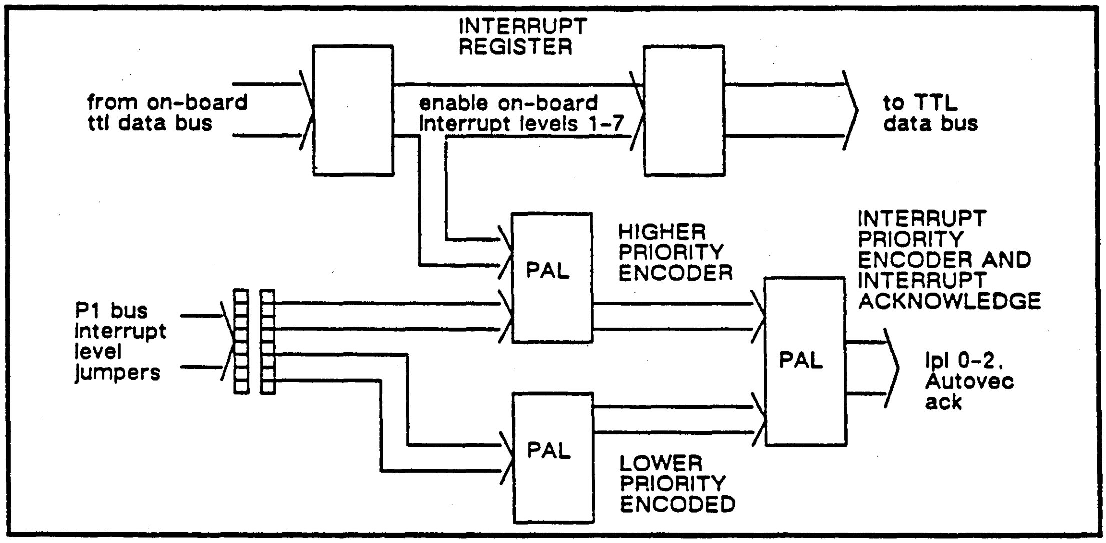

INTERRUPT REGISTER
from on-board ttl data bus
enable on-board interrupt levels 1-7
to TTL data bus
PAL
HIGHER PRIORITY ENCODER
INTERRUPT PRIORITY ENCODER AND INTERRUPT ACKNOWLEDGE
P1 bus interrupt level jumpers
PAL
Ipl 0-2, Autovec ack
PAL
LOWER PRIORITY ENCODED

FIGURE 2-9: INTERRUPT PRIORITY LOGIC BLOCKS

2.10 SPURIOUS INTERRUPT ERROR PROCESSING

A spurious interrupt results during an interrupt cycle when no device on the on-board or VME bus responds to the IACK sequence. In the case of a spurious interrupt, the host processor will separate the process of this error from the resulting bus error, and in some cases will try to rerun the bus cycle (return from exception processing) to clear up the problem.

A common example of this kind of exception processing would be indicated if the IACK daisy-chain did not find the SCSI card on the backplane during an interrupt request for a disc transfer from SCSI. If the jumpers were not installed properly, a spurious interrupt would result if the CPU board interrupt logic could not pass the IACK to SCSI.

In the example above, the user is notified by a software generated (echoed) message to the display screen. This message tells the user that SCSI is not responding to the IACK, a spurious interrupt results, and the system will attempt to sync the discs (dumping to the available device), and system software will attempt to reboot.

It is often handy to be able to isolate spurious interrupts between onboard and offboard interrupts. Pull the jumper for the spurious level you are experiencing (Chap. 3) and If the error changes then the problem was caused by an offboard device that is continuously interrupting, if not problem is caused by a device on the CPU board

2.11 BUS ARBITRATION ON THE SUN 3 SYSTEM

For bus arbitration on the Sun 3, the system architecture will support multiple arbiters which are jumperable on the CPU card (also refer to the configuration section of this manual). The arbitration for the VME bus is monitored at a Bus Request Level 3, and the system will release the bus on request (ROR) at the completion of the current cycle.

To further meet with VME compliancy, Sun 3 architecture will provide for bus timeouts. At the first level timeout, the CPU will support a 3.36 usec RETRY period which will cause the bus to FREEZE (no other VME board will be granted the bus during a bus freeze). While the VME bus is in a freeze state, devices that use the local (on-board) bus as a DVMA channel-such as the refresh logic and Ethernet-can execute their cycles.

The second level of timeout supported in this architecture will try to rerun the bus cycle 256 times before the process times-out. This timeout operation is provided only when the CPU is configured as the VME bus master (see configuration section). Timeouts for the VME slave or user modes will not be provided for, since it is the responsibility of each bus master to provide their own timeouts.

If another bus master tries to access the VME bus simultaneously to the CPU access cycle, while accessing the P2 bus, the CPU cycle will typically be re-run.

During bus arbitration, a CPU cycle to VME is accomplished via the VME master interface, and VME to CPU cycles are accomplished via the VME slave interface. The following flowchart illustrates how bus arbitration takes place from a system hardware standpoint.

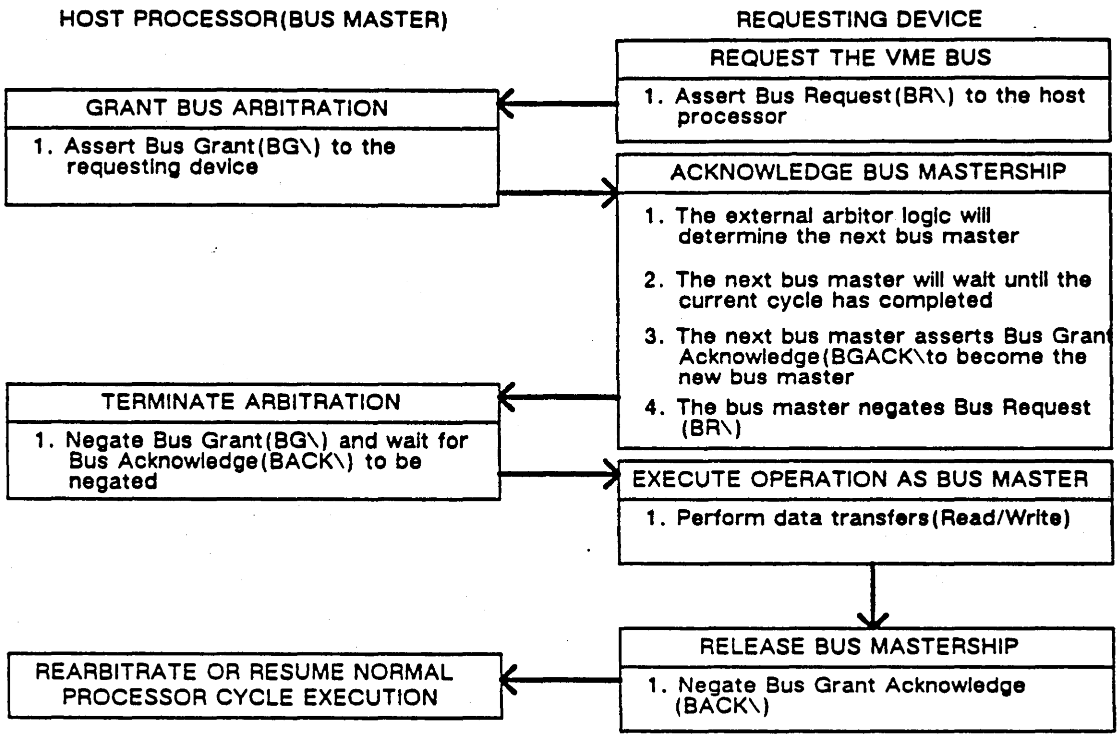

HOST PROCESSOR(BUS MASTER)
REQUESTING DEVICE
REQUEST THE VME BUS
1. Assert Bus Request(BR\) to the host processor
GRANT BUS ARBITRATION
1. Assert Bus Grant(BG\) to the requesting device
ACKNOWLEDGE BUS MASTERSHIP
1. The external arbitor logic will determine the next bus master
2. The next bus master will wait until the current cycle has completed
3. The next bus master asserts Bus Gran Acknowledge(BGACK\to become the new bus master
4. The bus master negates Bus Request (BR\)
TERMINATE ARBITRATION
1. Negate Bus Grant(BG\) and wait for Bus Acknowledge(BACK\) to be negated
EXECUTE OPERATION AS BUS MASTER
1. Perform data transfers(Read/Write)
RELEASE BUS MASTERSHIP
1. Negate Bus Grant Acknowledge (BACK\)
REARBITRATE OR RESUME NORMAL PROCESSOR CYCLE EXECUTION

FIGURE 2-10: BUS ARBITRATION FLOWCHART

2.12 DVMA OPERATIONS

Sun 3 architecture supports three types of DVMA access including the Refresh cycle to memory, the Ethernet Interface cycle to memory, and the VME Slave Interface cycle to the CPU virtual address space and memory. The SCSI interface is also included as an on-board DVMA device on the 3/50,3/60.

Devices accessing memory data such as the Ethernet Interface or the refresh cycle, will access on-board and expansion memory while the supervisor data (FC=5) and must have TYPE 0 set in the page table entries in the Memory Management Unit (MMU) to complete successfully. The VME bus Slave Interface will support both system and user DVMA access into the CPU virtual address space. These accesses can operate as byte, word and longword transfers.

Accessing the memory data (as an Ethernet or refresh DVMA cycle) under any other condition than a TYPE 0 access, or trying to access devices other than memory, will result in a bus error. To further ensure proper or successful execution of a memory access cycle, the DVMA devices are prioritized as to when they will have a legal pathway to the memory data or CPU space.
The DVMA device priorities are set by the DVMA controller logic. The priority levels start with the refresh cycle(nothing can stop a refresh cycle in terms of priority level), next go to the Ethernet Interface (which can issue a hold to lock out lesser priority levels including a dynamic hold feature to lock out the next level), next the VME slave accesses, and finally the 68020 to 68881 interface. There are a number of DVMA features which need to be pointed out at this time. They are as follows:

1. Concerning address space- DVMA address space accesses are performed as data read/write operations in the supervisor and user data context.
2. A far as protection is concerned- protection applies the same way as it does with the CPU. That is, read/write capabilities in the page map has to be enabled to allow corresponding types of access.
3. Concerning parity errors- DVMA cycles that cause parity errors will be reported back to the master, and will result in a bus error.
4. About statistics bits- access and modification bits are set on successful DVMA cycles, and are reported the same way as CPU cycles via the statistics control PAL.
5. During deadlocks- DVMA devices that cause a deadlock with the CPU(system bus conflict) will be resolved by rerunning the CPU cycle.
6. Self reference- DVMA cycles that reference themselves(such as the system bus trying to reference the system bus) will result in a bus error report.
7. As far as error handling is concerned- when a DVMA cycle results in a bus error, the error is sent to the controlling master, which will typically stop transferring.

FIGURE 2-11: DVMA POINTERS

2.13 DVMA CONTROL LOGIC

The DVMA control logic includes a number of request flip-flops, which are used to generate the level DVMA enable request to be latched and be asserted to the DVMA arbitration PAL. The arbitration PAL is a synchronous state machine which controls the interaction of Direct Virtual Memory Access (DVMA) with the host processor and the P2 bus.

The primary function of this state logic is to reflects the prioritized requests from the legal DVMA devices, and in turn, controls the bus request, bus grant and bus grant acknowledge handshake with the 68020. The output from the DVMA controller will include the channel enables and address strobes for the DVMA cycles requested. The following block diagram illustrates the flow through of requests from the DVMA devices for the bus.

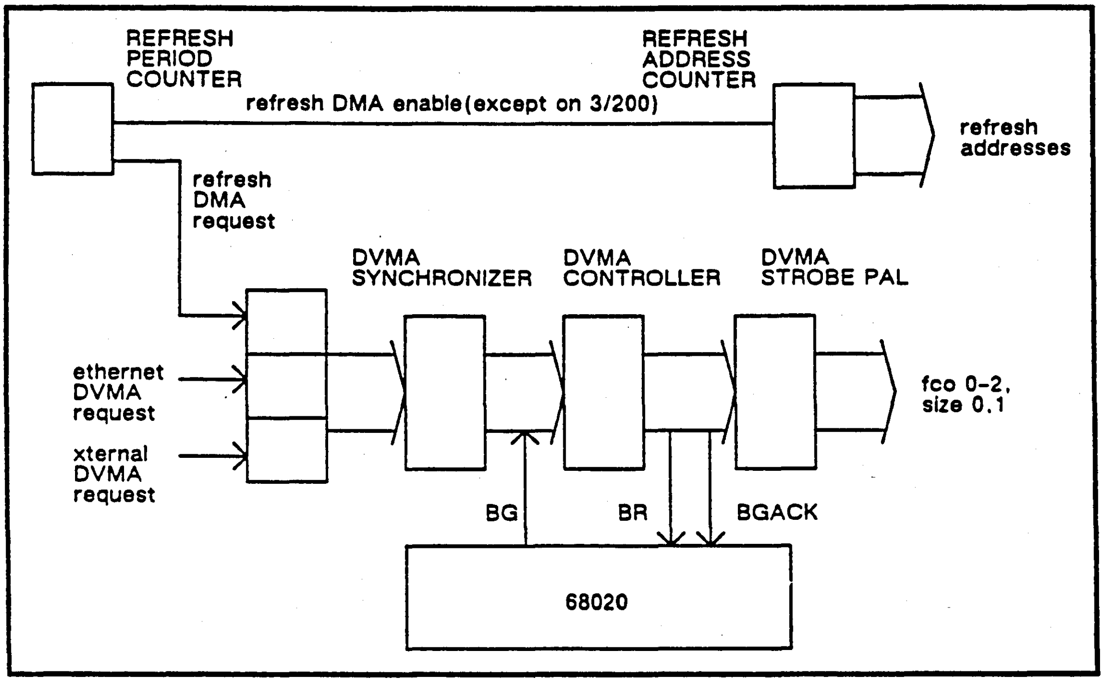


FIGURE 2-12: DVMA CONTROLLER LOGIC PATHWAY

The inputs to the DVMA control logic include the synchronized refresh request which is asserted every 15.7 usec to accomplish the dynamic RAM (DRAM) refresh cycle. Another input to the state logic will synchronize the Ethernet requests for the DVMA channel to memory data. In addition, the Ethernet hold signal to the controller will assure that Ethernet is able to maintain the bus for an extended cycle.

To control the actual DVMA cycles, data transfer and size acknowledge, as well as bus error signals, are presented to the controller at a specified time for CPU recognition. This recognition will be presented to the CPU to end the current cycle.

The controller logic must also ensure that the bus is locked while doing extended (fast back-to-back) accesses with VME devices. These faster cycles increase the burst rate available to the requesting VME device (refer to Figure 2-10).
An external VME device requiring the channel , will assert an external request to the controller to validate its access. To complete the handshake with the

CPU, a bus grant will be presented to the controller. Thus, the output of the controller will generate the bus grant in/out daisy-chain across the back plane(to the requesting board) and complete the handshake for a valid arbitration cycle to begin executing with the VME device. The next section will describe how the VME bus interface will control for VME bus accesses.

2.14 THE VME BUS INTERFACE

The interface to the VME bus is dual ported for the VME bus master and VME bus slave cycles. The VME bus interface includes the VME arbitor and request logic, the master and slave interfaces, and the VME data buffers.

As stated in Chapter ONE, the VME bus master interface controls access from the host processor to the VME bus. While the VME bus slave interface controls the VME to host processor cycles.

Under Sun 3 architecture (to comply with VME bus specifications), interface support includes control for 32-bit address and data for both master and slave, timeout control (-100 usec minimum), bus arbitration at level 3, bus release-on-request (after the completion of the current cycle), and control for VME bus interrupt handling (explained in the interrupt section of this manual).

During a read-modify-write cycle over the VME bus interface the cycle is executed between the CPU and the VME device when the CPU is the bus master. Normally, the CPU will be configured as the bus arbitor (note, however, that a jumper is provided on the CPU card to disable arbitration and provide it by another bus master).

The following diagram illustrates the master and slave cycle, and will include the VME arbitor and request logic.

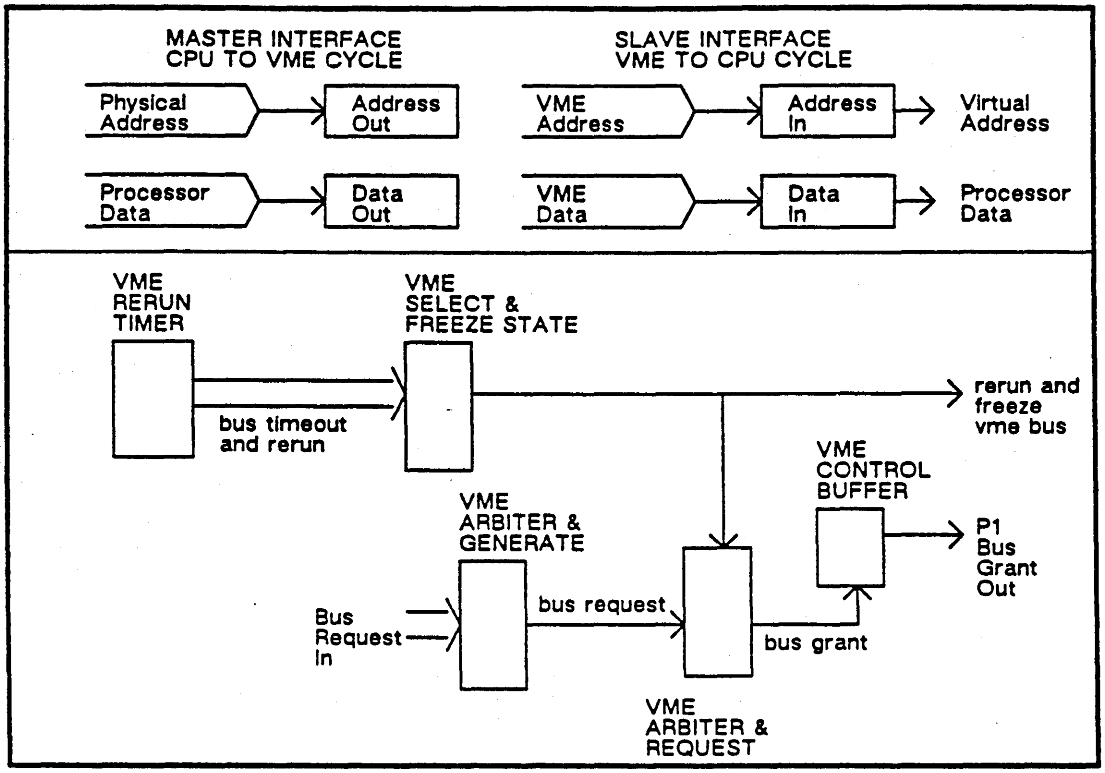


FIGURE 2-13: VME BUS CYCLE CONTROL

2.15 ETHERNET INTERFACE AND DATA TRANSFER CONTROL

The Ethernet interface implemented in Sun 3 architecture provides a communication channel for high-speed data transactions between other Sun systems and peripherals on a local-area-network (LAN).

The primary purpose for using an Ethernet interface in Sun architecture is to provide a low-cost local area net architecture, that will achieve data transactions between general-purpose workstations (known as nodes) on a common net. Each node installed on the net will have a unique identifying number (48-bit), that will correspond to the Ethernet number provided by the ID PROM.

During transactions over the net, this identifier will be distinguished from node to node to ensure the communications transaction involves only the appropriate devices. Within each network, each system is given a net number, and each system is given a local address. These network configuration parameters are monitored by the system software, and are recorded via specified files such as RC.LOCAL, the HOSTS file, the HOSTS.EQUIV file, the ETHERS and the .RHOSTS file. These files are found under the /etc directory with the exception of the .RHOSTS file, it is in the home directory of each user.

During a system boot-up, these files are read ensuring that the individual networks and nodes are bound to their specified addresses (divisions). The basic structure of these divisions is found in the system architecture layers.

Within the Ethernet structure, there are major divisions between the Ethernet layers. These layers include the client layer (including system management), the data link layer, and the physical layer.

The client layer works with the system software (NFS), to manage the network, workstation access, file transactions between general-purpose workstations and peripherals, and network control who gets the file, and when do they get it.

The physical layer involves the network link which includes the BACK END (82586 Ethernet Data Link Controller) of the architecture, the Ethernet control register, and the FRONT END (82501 Ethernet Serial Interface) of the architecture. The data link layer will provide for the communication facility which gives the workstation access to the physical channel including CSMA-CD to meet 802.3 specifications. Figure 2-13 illustrates the architectural layering of the Ethernet, and Figure 2-14 illustrates a typical Ethernet layout.

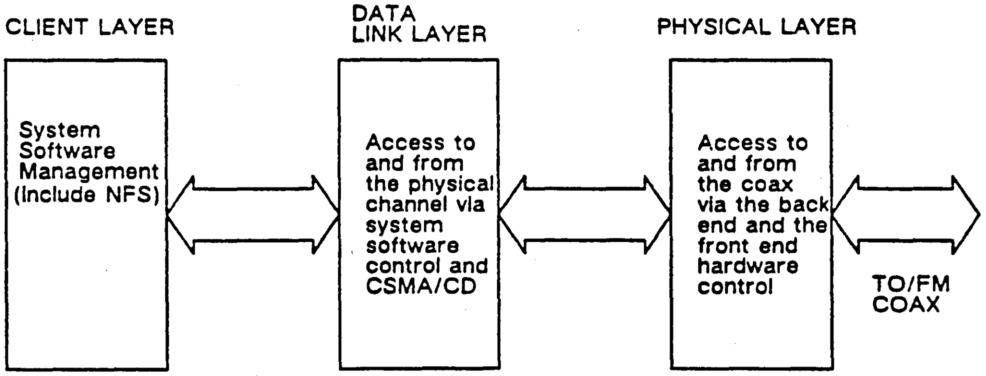

CLIENT LAYER
DATA LINK LAYER
PHYSICAL LAYER
System Software Management (Include NFS)
Access to and from the physical channel via system software control and CSMA/CD
Access to and from the coax via the back end and the front end hardware control
TO/FM COAX

FIGURE 2-14: ETHERNET ARCHITECTURAL LAYERS

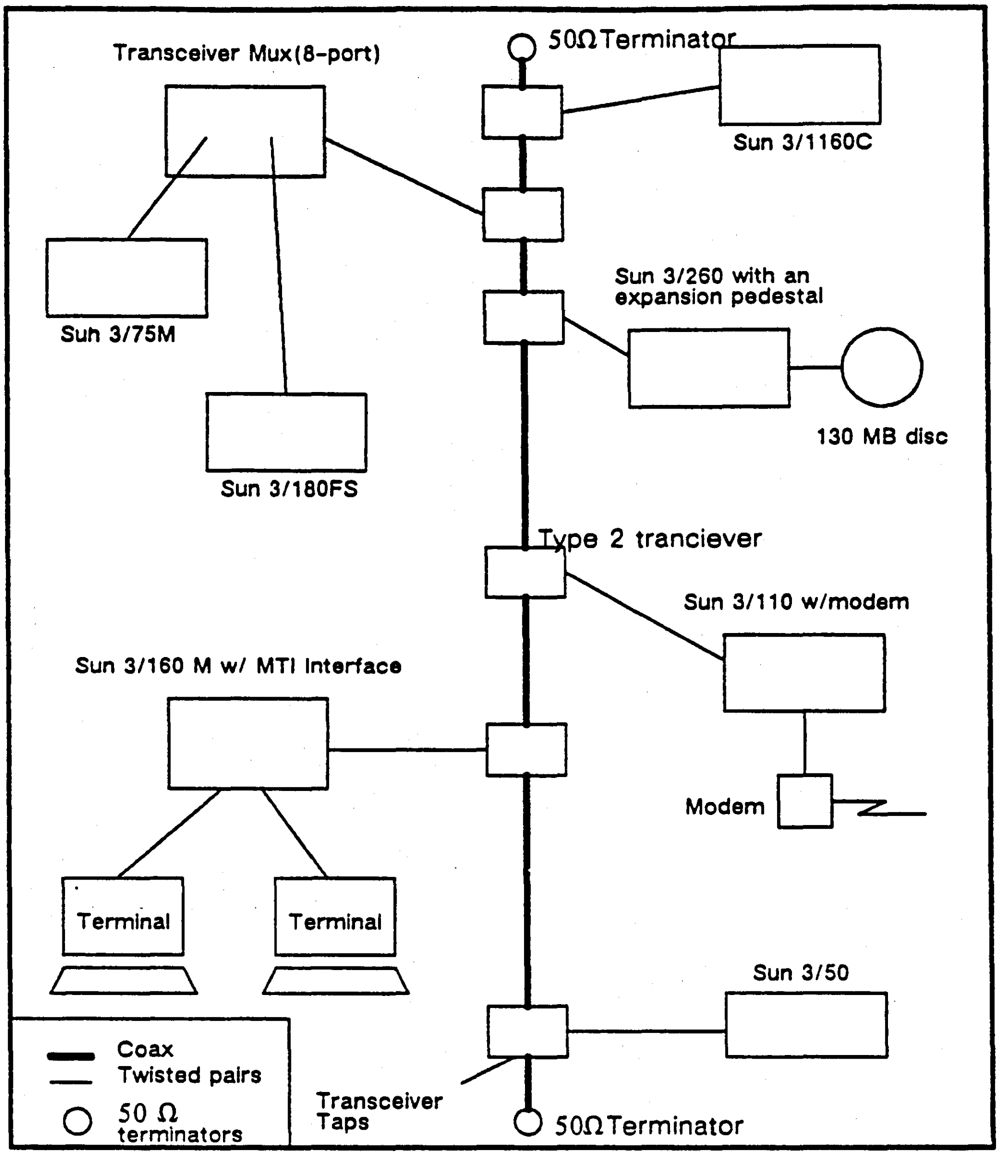


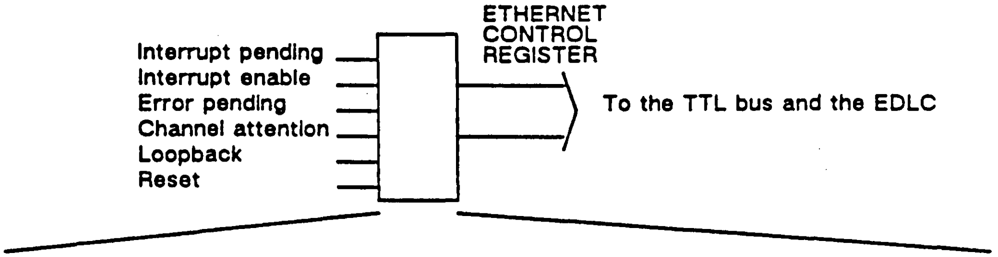
FIGURE 2-15: SUN NETWORK EXAMPLE

Note: 3/50, 3/60 can be hooked up to standard 50 ohm coax using internal transcievers.

2.16 ETHERNET TRANSACTIONS

The Ethernet interface, including the 82586 (EDLC), the 82501 (SIA), accesses the top 16 MB of the current virtual address space with the supervisor data context. All access must be to main memory (TYPE 0), or the cycle will result in a bus error.

If an Ethernet cycle concludes with a bus error, the error is set in the Ethernet control register, and further activity with the EDLC is suspended. The following conditions indicate the successful or unsuccessful data transfer between the memory and the Ethernet interface.



1. The interrupt pending signals an interrupt from the EDLC or an error pending
2. The interrupt enable notifies the CPU that the EDLC wants to interrupt
3. The error pending indicates an error condition to the bus error register. This condition indicates that the error happened during a EDLC operation, thus, inhibiting any further channel activity. To clear this condition, the reset condition to the EDLC must be active
4. The channel attention condition indicates that the CPU wants to do a transaction with the Ethernet Data Link Controller
5. The loop back condition controls when the front end(ESI) encoder/decoder is configured in a loop back mode(if connected or not connected to the transceiver cable)
6. The reset condition initializes the EDLC when active and allows normal activity when inactive

FIGURE 2-16: ETHERNET CONTROL REGISTER ACTIVITY

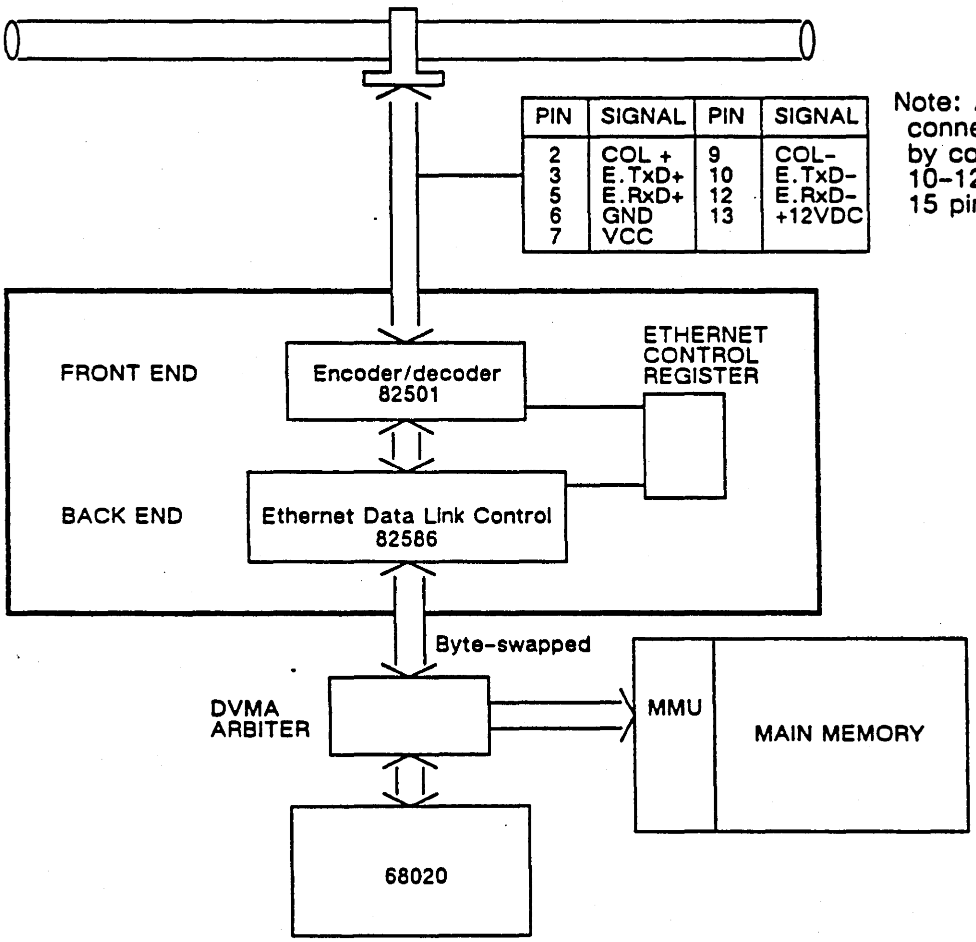
The EDLC 82586 is a high-performance local area network processor which utilizes the Carrier Sense Multiple Access with Collision Detection link access method (CSMA/CD). When the Ethernet controller is used for a transaction under system software control, the following steps are maintained by the layers mentioned earlier:

1. When a workstation user runs a file transfer program (NFS implementation) and specifies a file to be shared or transferred between a sending and receiving device, system software takes control.
2. System software maps the file characters into device-dependent virtual characters to comply with protocol specifications.
3. The mapped character stream is routed to a virtual circuit, which is set up between the two devices.
4. The virtual circuit software breaks the character stream into packets for transmission.
5. The packets are then passed to the Ethernet driver software.
6. The Ethernet driver then copies the packet into a packet buffer, and instructs the controller to transmit the packet.
7. The controller waits until the coax is not in use, and then transmits the packet.
8. The Ethernet transceiver receives the packet bit stream and injects it into the coax.
9. The receiving station recognizes its address and reverses the above procedure- bits are received by the transceiver, fed to the controller, passed to the system software that reassembles the packets, maps the characters, and stores the data

FIGURE 2-17: FILE TRANSFERS OVER THE ETHERNET

The front end of the Ethernet is implemented with the 82501 Ethernet Serial Interface (ESI). The major feature of the front end is to perform Manchester encoding and decoding of the transmitted and received frames, respectively. To implement this feature, the ESI uses an analog phase-lock-loop technique (PLL) to extract the received clock from the data.

The front end is the interface to the Ethernet connection on the rear panel of the CPU card. This connection drives and receives the packets, as well as providing a noise filter to prevent spurious signals from interfering with the receiver circuitry. The ESI will also provide for the collision presence during a transmission. By tying this sense capability to the EDLC, the front end can signal the back end to retransmit the data packet. The following figure illustrates the Ethernet interface, including the back end, the front end, the Ethernet control register, and the DVMA channel.



PIN	SIGNAL	PIN	SIGNAL
2	COL +	9	COL-
3	E.TxD+	10	E.TxD-
5	E.RxD+	12	E.RxD-
6	GND	13	+12VDC
7	VCC		

Note: A handy loopback connector can be made by connecting 3-5 and 10-12 together on a 15 pin DIN plug.

FIGURE 2-18: ETHERNET SYSTEM LINK

From an electronic standpoint, a number of sequences must occur to successfully transfer data between the host processor and the Ethernet Data Link Controller. The steps are as follows:

1. When Ethernet wants access to main memory, it will assert read/write control. For the CPU to gain access to the Ethernet, it asserts channel attention.
2. The Ethernet will present its request to the DVMA arbiter and request logic.
3. The arbiter level for Ethernet is at a lower level than refresh cycles, and therefore the Ethernet will wait until the current cycle completes. The arbiter logic will continuously request the bus from the bus master until the Ethernet controller drops its request.
4. Once the Ethernet has the bus it may hold it until its cycle completes. The Ethernet controller will produce its own 24-bit address for cycles to memory.
5. The Ethernet will then assert read/write control, and an Ethernet DVMA cycle will result.
6. At this point, a write cycle (Ethernet to memory) is executed, and the DVMA controller will enable the Ethernet 8-bit bus front loading(latch) transceivers.
7. On a read cycle (memory to Ethernet) the DVMA controller will latch the data read from memory into the Ethernet 8-bit front loading(latch) transceivers. These transceivers offer read/write data routing to and from memory.
8. There are four 8-bit bus front loading(latch) transceivers for long word transfers to and from memory over the P2 bus.
9. The EDLC is physicall byte-swapped to the processor data bus so that data is stored into memory in 68020 byte order.

FIGURE 2-19: TRANSACTIONS BETWEEN CPU AND EDLC

The data link layer of the Ethernet present a means of resolving bus transfer conflict. The following procedure details how link control works:

Carrier Sense Multiple Access with Collision Detection(CSMA/CD) is a generic term for a class of link management, used by the Ethernet interface and system software. This procedure is so-called because it:

1. Allows multiple stations to access the broadcast channel at will.
2. It avoids contention via carrier sense and deference. This is a process by which a data link controller delays its transmission when the channel is busy, so as to avoid contention with on-going transmission over a network between workstations.
3. The link control procedure will resolve contention by collision detection and retransmission control via the ESI and EDLC, respectively.

FIGURE 2-20: DATA LINK CONTROL PROCEDURE

2-17 ON-BOARD VIDEO DISPLAY FRAME BUFFER CYCLES

The on-board video frame buffer is a 128 K Byte block of memory implemented with dynamic RAMS (DRAMS). The video frame buffer is accessed over the 32-bit P2 bus via the P2 bus interface state PAL control, and the video control decoder.

When a copy mode(the copy mode condition will be enabled via the video select decoder PAL to the video control decoder PAL. When enabled, a condition will exist on-board the CPU card, which will cause any write operation to a 128 K Byte block of memory (starting at a specified address) to also be written into the video frame buffer.

The copy mode cycle will take place over the P2 bus, and the data will be input to the video memory. To write to memory,. the write strobe decoder will generate write enable signals to the video memory data latches. These latches control the P2 data input to the video frame buffer over a 64-bit path.

While a display enable signal is in its active state, the display data will be driven from memory to the output shift latches. The shifted data is presented to the inputs of the TTL-to-ECL video data converters, and shifted out as differential video outputs to the monitor. The video clock, oscillating at 100 MHz, provides the clock output for the video shift registers.

The video output will be at a 10 usec, 100 MHz video rate, and the video display format will be either 1152 x 900, or 1024 x 1024. The pixel rate will be 10 nsec per pixel.

A horizontal and vertical state machine will provide for the display enable on signal (display on), the vertical blanking signal (display off), as well as the horizontal and vertical sync signals. The video data will be in ECL form, and the sync signals will be in positive TTL pulse form (sync on the rising edge).

The video frame buffer is mapped to the display screen as follows:

1. Data bit 15 of word 0 of the frame buffer is the first pixel in the upper left corner of the display.
2. Consecutive words are displayed along the horizontal scanline from left to right.
3. After the display width(number of pixels) has been displayed, the next word is displayed at the biginning of the next horizontal line, up to the displayed hight specified.
4. The display width and the display height are implemented as constants via the system software(such as font type and size).
5. The display bit polarity is '1' bits for pixel black(pixel off), and '0' bits for pixel white (pixel on), depending on the display mode(normal video, or reverse video).

FIGURE 2-21: VIDEO DISPLAY MAPPING

The video block diagram below illustrates the data pathway, via the P2 bus, to and from the video frame buffer. Note that video refresh is controlled by the video refresh counters.

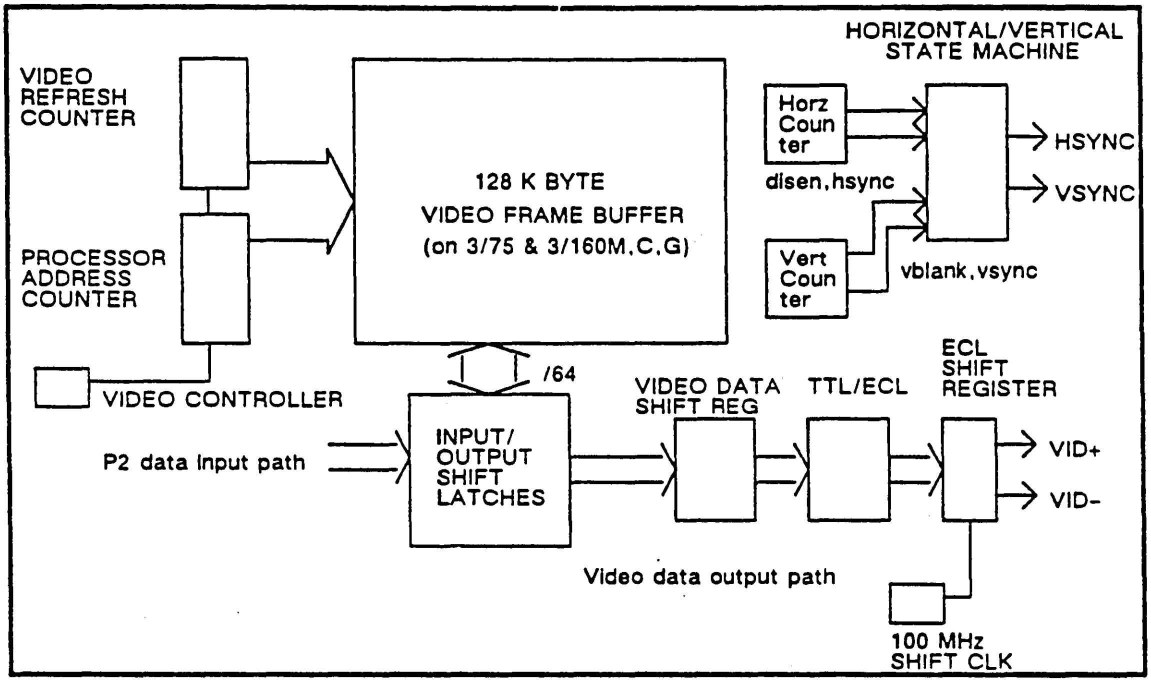


FIGURE 2-22: VIDEO DATA INPUT/OUTPUT PATHWAY

VIDEO CHARACTERISTICS	
FEATURE	DESCRIPTION
Visual display	900 horizontal lines with 1152 pixels per line(version A) 1024 horizontal lines with 1024 pixels per line(version B)
Video clock	10 nsec pixel rate at 100 MHz video rate
Horizontal cycle	16.00 usec at 62.5 KHz
Vertical cycle	15000 usec at 66.67 Hz
Horizontal retrace	4.48 usec
Vertical retrace	600 usec
Video outputs	differential outputs VID+, VID-(ECL levels) HSYNC-positive TTL pulse(sync on rising edge) VSYNC-positive TTL pulse(sync on rising edge)

FIGURE 2-23: VIDEO SIGNAL CHARACTERISTICS

2.18 Sun 3 Technical Overview

Figure 2-24 shows a block diagram of the Sun 3 Carrera family product line.

Points of interest are:

CPU has 4 Mbytes of memory on board.

MMU controls transfer of data between virtual and physical memory.

B&W video is on board and does not require an additional VME card.

ETHERNET is also on board and is in CPU space with 68020.

PRIVATE MEMORY busses are **not** connected to each other.

MEMORY EXPANSIONS and **FPA** are on the same Private Memory bus.

GRAPHICS PROCESSOR and **COLOR CONTROLLER** talk on VME bus.

GRAPHICS PROCESSOR and **GRAPHICS BUFFER** talk on Private Memory bus.

SCSI HOST ADAPTER is a VME card that talks to one or more SCSI controllers.

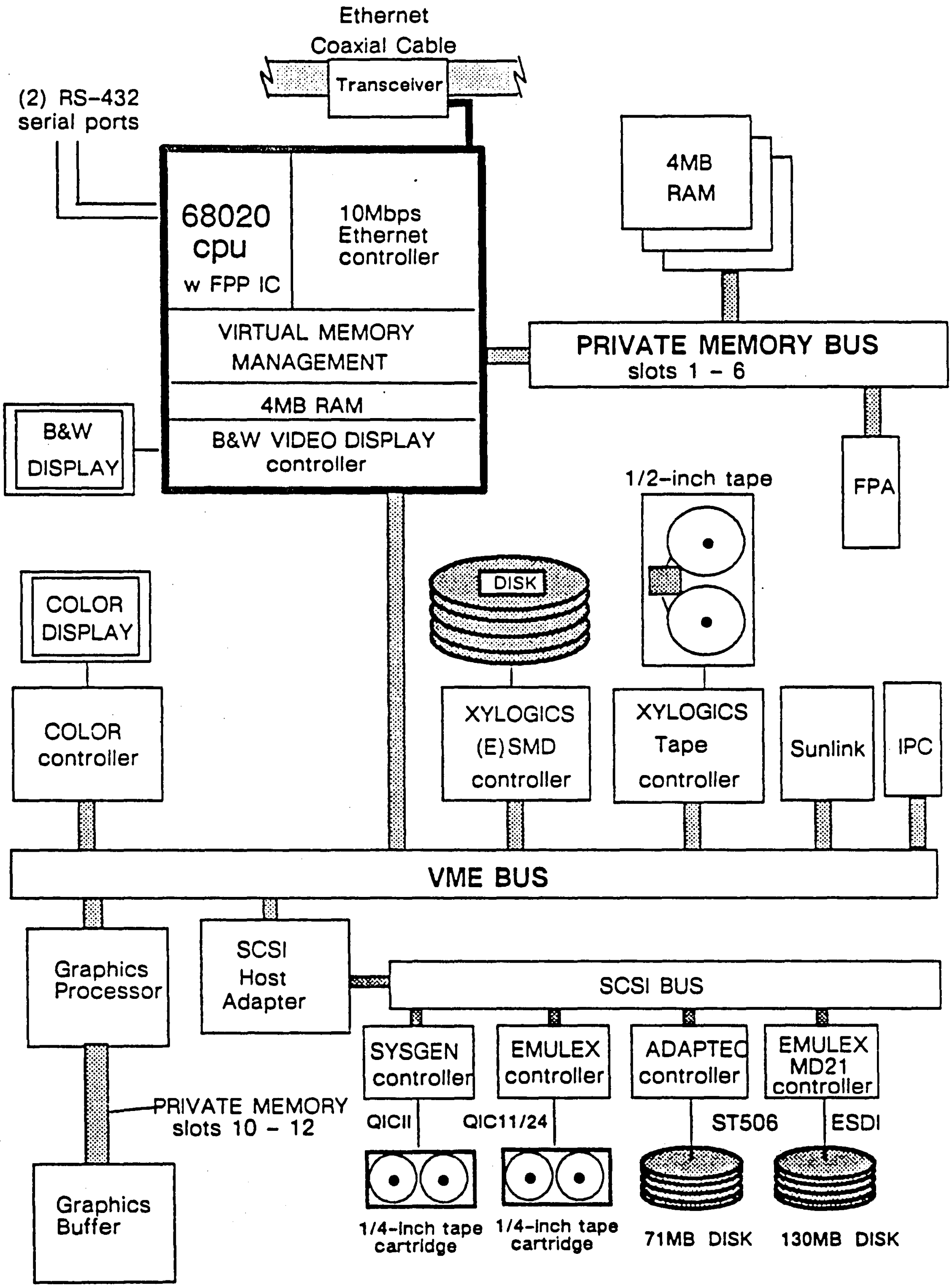


FIGURE 2-24: Sun 3 Technical Overview Block Diagram

This page left blank.

CHAPTER THREE

SYSTEM CONFIGURATION

OVERVIEW

This section of the Sun 3 training manual will cover system configuration, to the board and module level, and will include coverage on the Sun 3/75M, the Sun 3/160M and Sun 3/160C, 3/140, 3/150 and the 3/180FS (Carrera) products.

3/50	configuration is in Chapter 8	Model 25
3/110	configuration is in Chapter 9	Prism
3/200	configuration is in Chapter 10	Sirius
3/60	configuration is in Chapter 11	

The system configuration section will be divided into 8 parts including a section on the pedestal layout, card cage configuration, backplane configuration, sub-system layout, EEPROM settings, power supply options, and environmental factors. These sections will be followed with a complete set of assembly drawings of the CPU card, expansion memory and the SCSI card, as well as the layout of the Sun desktop, pedestal, and rack-mounted system. The monitors (monochrome and color) will be covered in Chapter Six.

3.1 SUN 3/75M SYSTEM LAYOUT

The Sun 3/75M is the general-purpose workstation which can be configured as a 'diskless node', or as a 'stand alone' system when configured with the Sun mass-storage subsystem (to be covered in chapter Seven-Sun system options for the workstation).

The Sun 3/75M diskless node is configured with the 68020/68881 CPU card and has 4 MB of on-board memory. The system comes with a two slot card base. The card base is mounted horizontally, and can accommodate the CPU card, an expansion memory board (4 MB) or an optional SCSI board (allowing the addition of the mass-storage subsystem). A dual function board is also available that provides both SCSI and 4 MB memory expansion.

Figures 3-1 and 3-2 illustrate the possible configurations that can be added to the Sun 3/75M. Note that this system is designed to be used as a monochrome display-based workstation, supporting the high-resolution 19-inch landscape monitor. The resolution of the monitor is factory set to 1152 by 900 by 1.

3.2 SYSTEM USER SWITCHES

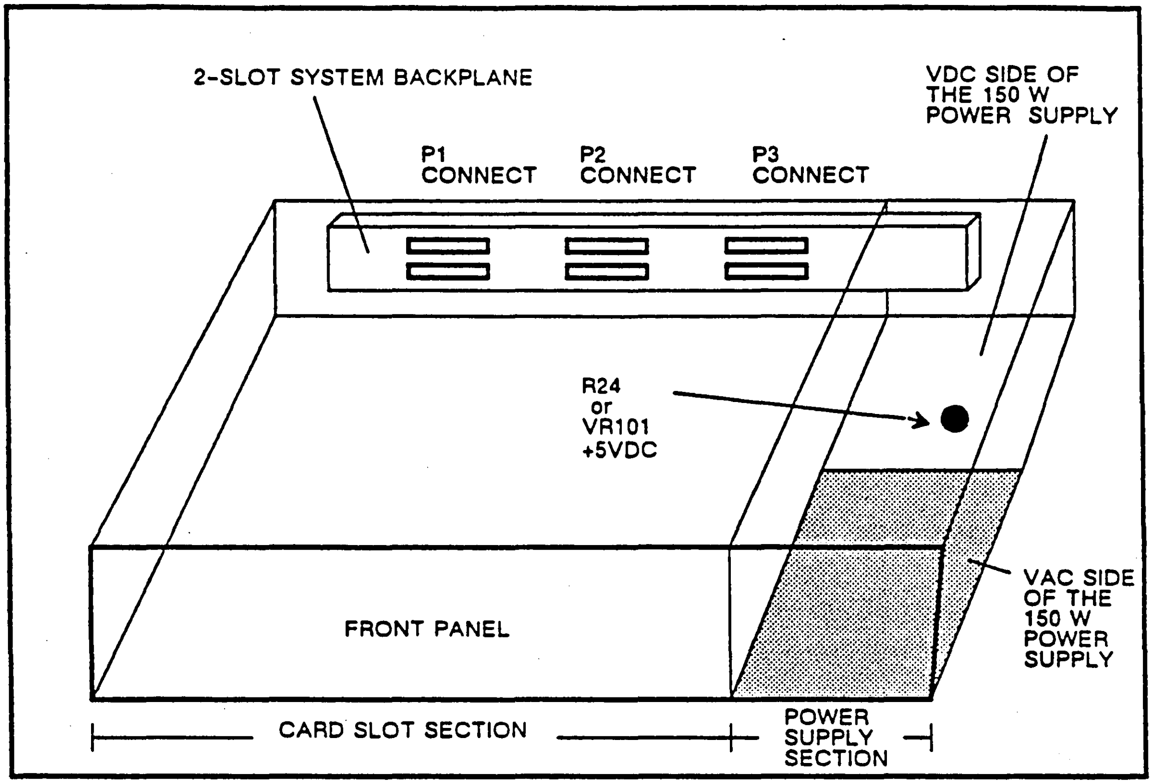
The 68020/68881 CPU card has two user switches on the rear panel of the card. These two switches are labeled DIAG and RESET. The DIAG switch allows the user to select the normal BOOT condition (switch to the right for 3/75M, switch down for 3/160M and 3/160C), or the user may elect to run the resident PROM-based system diagnostic (3/75M switch to the left, 3/160M and 3/160C switch up). If your switch is three position, the center position is typically not connected. This translates to a logic high and appears as though you selected DIAG mode.

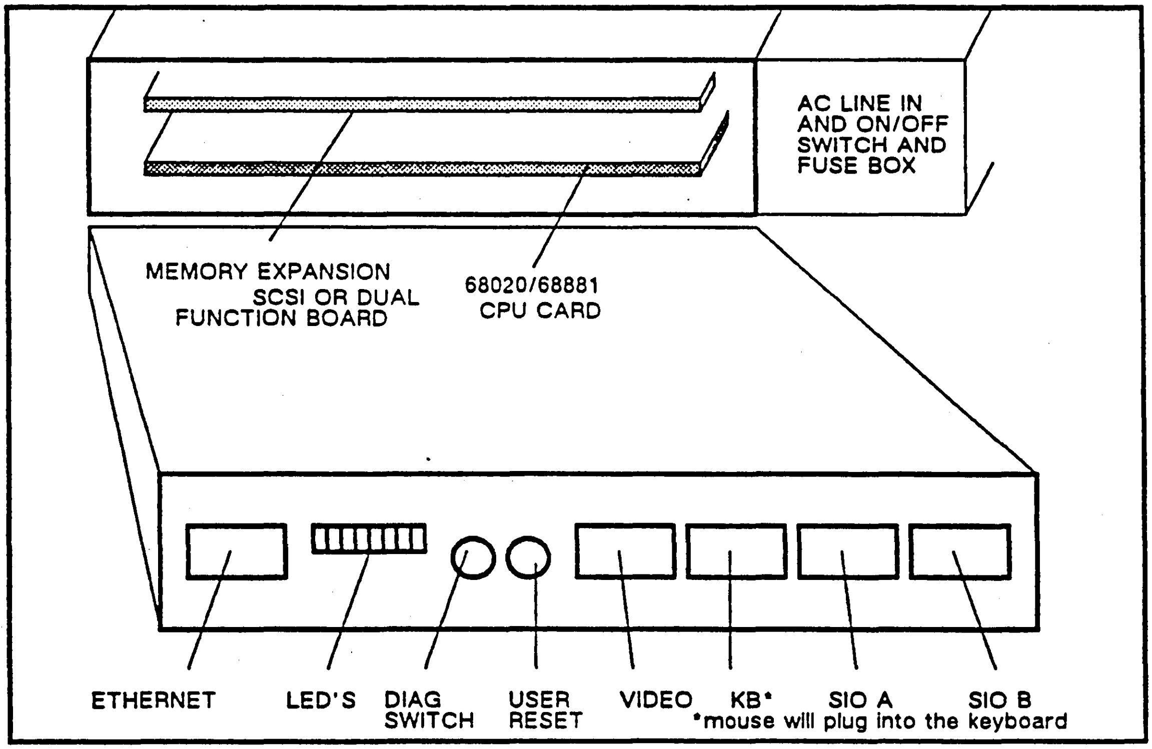
To run the system diagnostics, the user must have the appropriate console display connected to SIO port A. The console can be an ASCII terminal, and should be set for XON/XOFF, 8-data bit, 1-stop bit, no parity, full duplex and at 9600 baud. Since the SIO ports are set up as modem ports, a NULL MODEM cable will be required and set to DTE (Data Terminal Equipment) levels. FIGURE 3-3 illustrates the DTE to DTE null modem connection.

The second switch is the system RESET switch. This switch allows the user to reset the condition of the system by depressing the switch once. The reset will be the equivalent of a WATCHDOG RESET, as discussed in Chapter TWO. By utilizing the user switch, the system operator is able to reset the system without power cycling.

Note that the RESET switch can get you in to trouble if it is not used properly. The switch should only be depressed when the user wants the CPU to execute a WATCHDOG RESET.

If the user wishes to abort the current cycle, or reset the entire system, it is also possible to halt the system first by using the appropriate system software controlled disconnect command under the /etc directory. These executable disconnects include halt, sync, or reboot. The user must be the SU in order to halt the system. However, if you are not SU, and must halt the system, login as sync twice (to sync the discs), and execute an L1-A. The user can initiate a L1-A by depressing the appropriate keys, and thus, will return to the MONITOR LEVEL.




AC LINE IN AND ON/OFF SWITCH AND FUSE BOX
MEMORY EXPANSION SCSI OR DUAL FUNCTION BOARD
68020/68881 CPU CARD
ETHERNET
LED'S
DIAG SWITCH
USER RESET
VIDEO
KB*
SIO A
SIO B
*mouse will plug into the keyboard

FIGURES 3-1,3-2:SUN 3/75M BASE AND PANEL

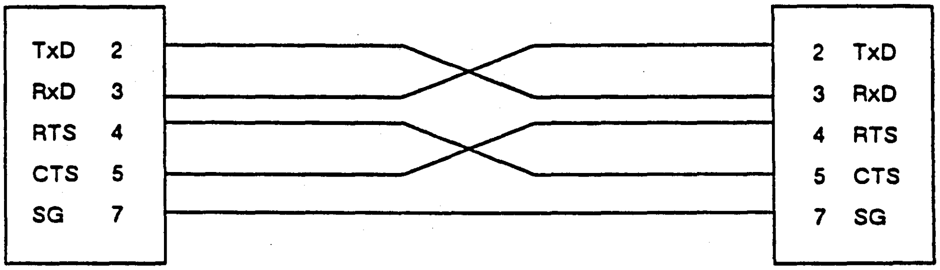


FIGURE 3-3: NULL MODEM CABLE FOR ASCII TERMINAL

3.3 SUN 3/160M and SUN 3/160C SYSTEM LAYOUT

The pedestal layout for the Sun monochrome and color workstations is the same in regard to pedestal frame and sub-system assembly. Both pedestals include a 12-slot card cage, and must be configured in a similar manner. The pedestal itself is in three sections including the main card cage section, the upper tray section and the removable door and power supply section. Figure 3-4 illustrates the pedestal layout. Full assembly drawings of the pedestal and the attachment modules are included at the end of this chapter.

Card cage voltage adjustments can be taken at the indicated test points along the side of the backplane. +5 VDC should be measured at the bus bars. The following table illustrates the internal wiring color code.

VOLTAGE	COLOR CODE
GROUND	BLACK
+5 VDC	RED
+12 VDC	BLUE
-12 VDC	BROWN
-5.2 VDC	YELLOW (WHITE)
+24 VCD	ORANGE (Expansion Pedestal)

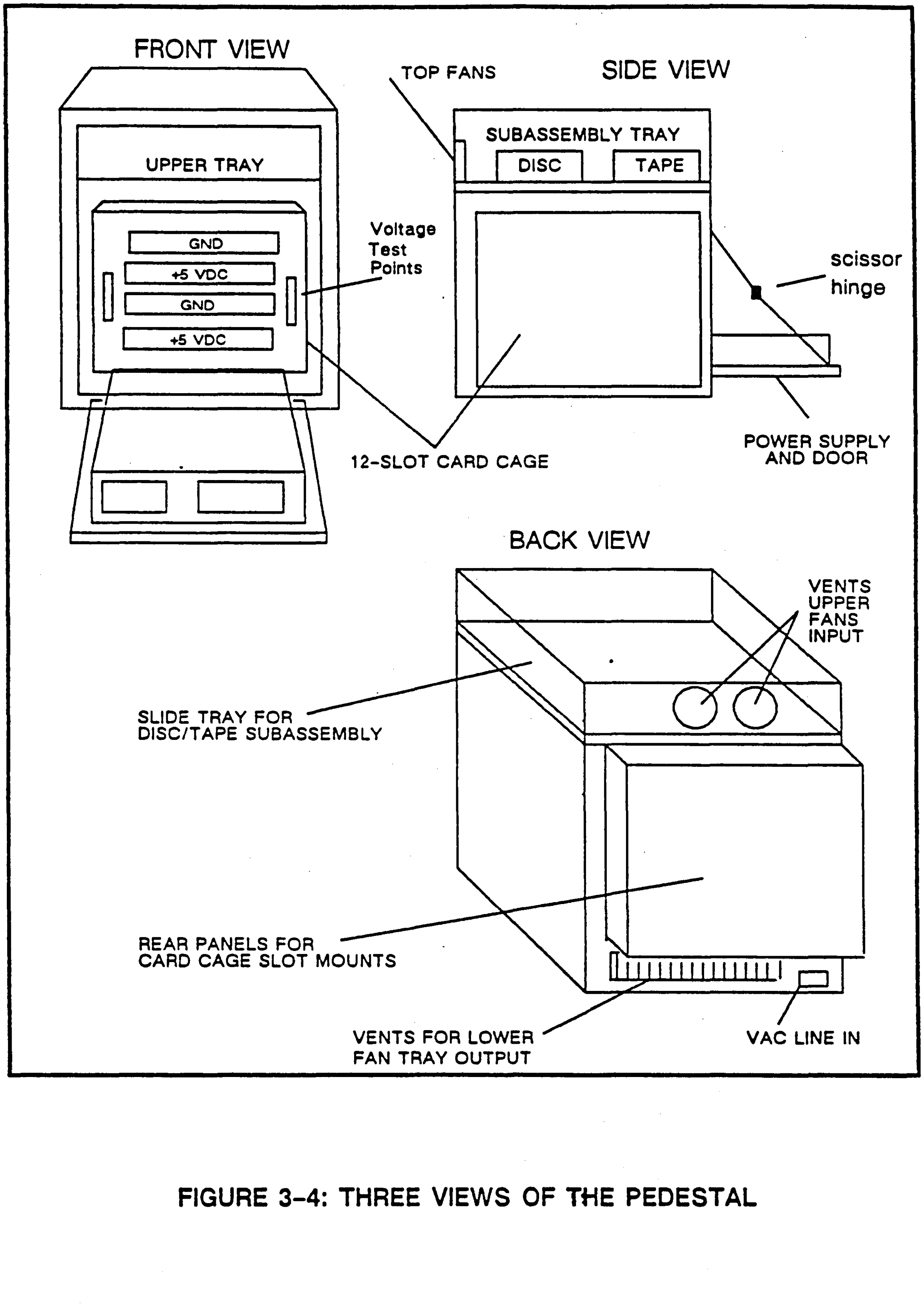
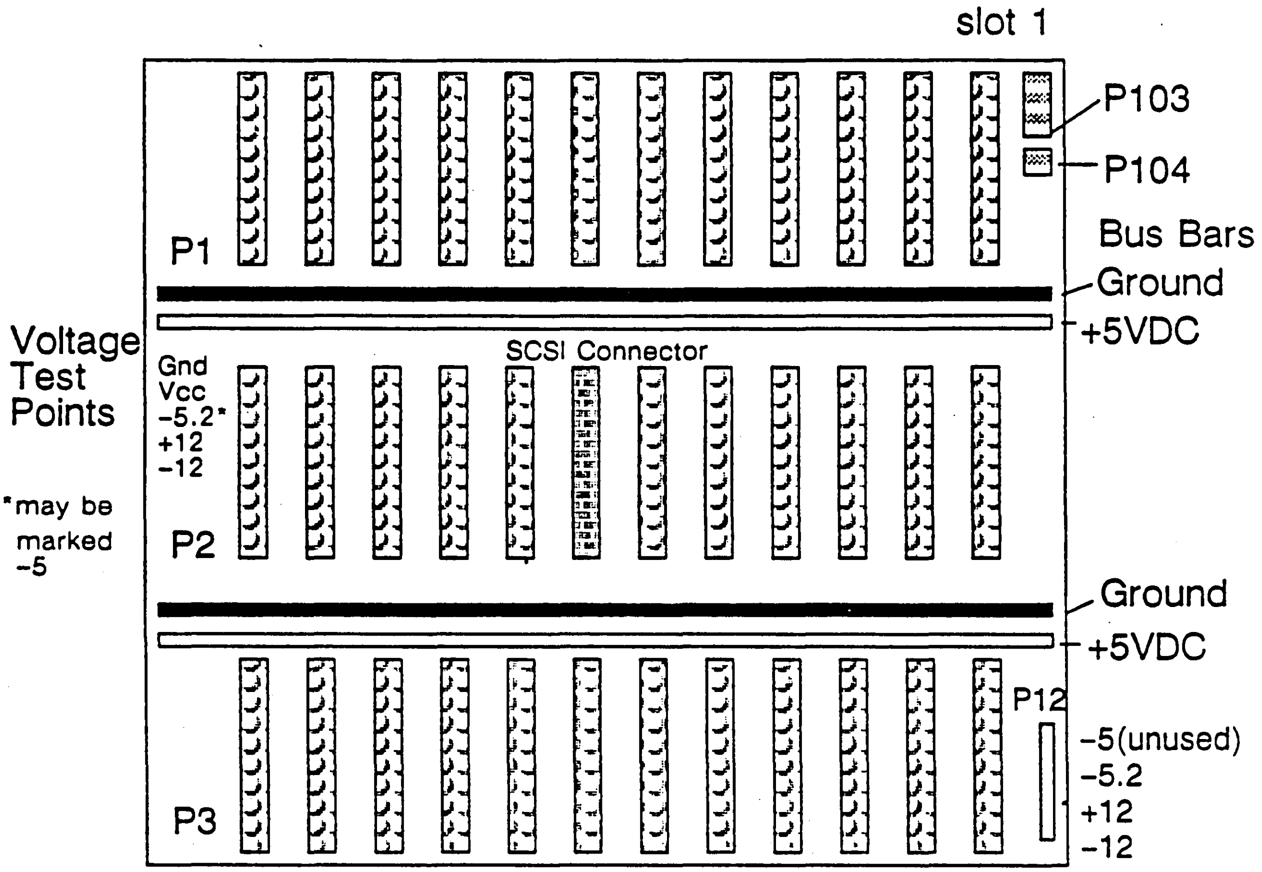


FIGURE 3-4: THREE VIEWS OF THE PEDESTAL

Configuring the card cage involves a little planning since there are major slotting combinations in a 12-slot card cage. To begin with, the VME backplane is divided up into three relative levels, referred to as P1, P2 and P3. These levels are designed to comply with the VME specification (see Chapter TWO), and are designed to slot the triple-height VME boards from Sun.

Each level of each triple-height board includes a 96-pin connector, which contains three rows of pins labeled A, B, and C. These connectors and pins will comply with the layout of the VME backplane found in Sun 3 pedestals. Figures 3-5 and 3-5.1 illustrate the connectors to the backplane, their rows, and their function.



CONNECTOR	FUNCTION	ROWS USED	SLOTS
P1	VME bus VME bus VME bus	A B C	1–12 1–12 1–12
P2	Private Memory VME bus Private Memory	A B C	1–6,10–12 1–12 1–6,10–12
P3	VCC,VEE,GND,+12 Private Memory VCC,VEE,GND,+12	A B C	1–12 1–6,10–12 1–12

FIGURE 3-5: VME CONNECTOR TABLE

3.4 PEDESTAL CARD CAGE LAYOUT

To configure your boards onto the 12-slot card cage, you must work from the rear of the pedestal. Facing the rear, the slots will be numbered from left to right, as slots 1 through 12. When facing the front of the pedestal (the power supply door), the slots will be read from right to left, as slots 1 through 12.

The user must take special care when configuring the card cage. It is important to note the relative positioning of the first 6 slots (1-6), so that boards installed in these slots will not interfere with private memory traffic. A second discrete private memory bus exists between slots 10-12. This facilitates communication between the Graphics Processor and the Graphics Buffer. To aid in configuring the pedestal, figures 3-6 and 3-7 will illustrate suggested card slotting. Note that the 3/150 is a six slot card cage. Slots 1-4 and 5-6 contain the discrete private memory buses. The 3/140 has a private memory bus over all three slots.

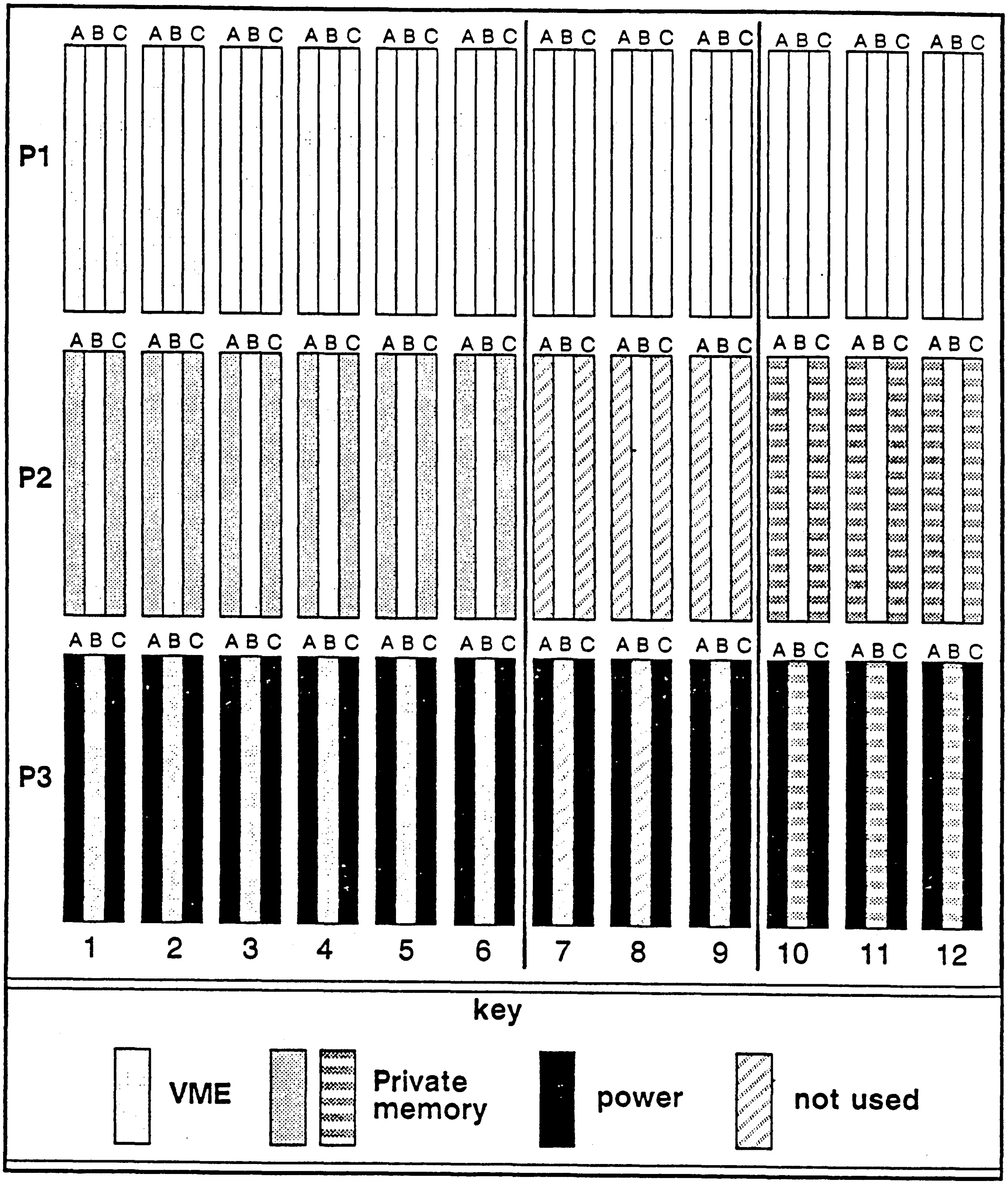
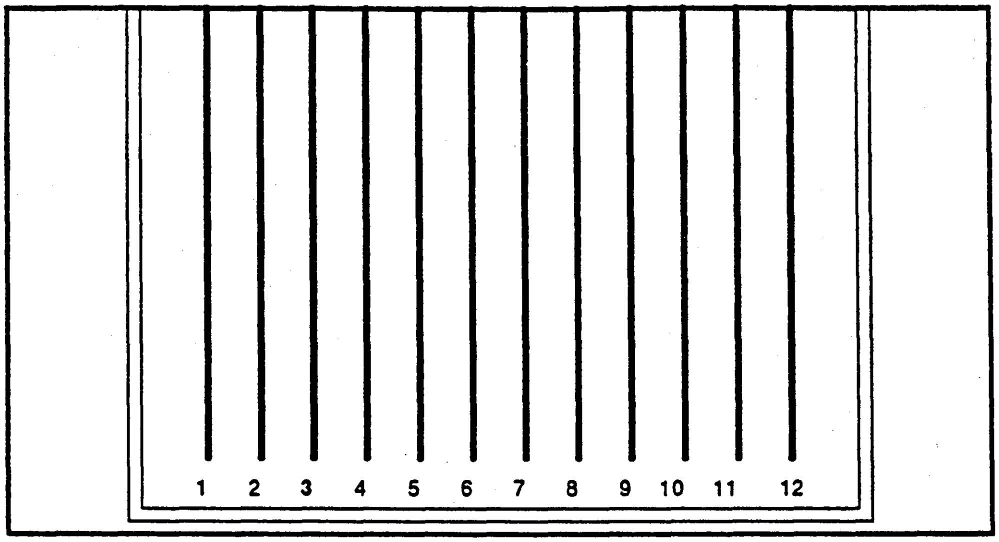


FIGURE 3-5.1 12 Slot Backplane VME Connector Diagram



REAR VIEW OF THE PEDESTAL(CARD SECTION SLOTS)

FIGURE 3-6: CARD SLOTS FROM THE REAR OF THE PEDESTAL

CARD TYPE	RECOMMENDED SLOT POSITION											
	1	2	3	4	5	6	7	8	9	10	11	12
Sun 3 CPU	A											
Memory Exp #1		A										
Memory Exp #2			A									
Memory Exp #3				A								
Sun VME FPA					A							
VME SCSI							A					
Sun GP/GP+/GP2										A		
Sun GB											A	
Sun Eth 2nd ctlr							A	B				
1/2 inch tape							A	B	C			
SMD 1st							A	B	C	D		
SMD 2nd								A	B	C	D	
Sun Color/CG3/CG5		K	A	B	C	D	E	F	G	H	I	J
ALM												A
Sunlink							A	B	C			
TAAC-1										A	*	*
Sunlink Channel Adpt.		A	A B	C B	C D	E D	E F	G F	G H	I H	I J	J

FIGURE 3-7: RECOMMENDED CONFIGURATION GUIDE

Pin	Signal	Pin	Signal
1	RXD0 (keyboard)	8	GND
2	GND	9	GND
3	TXD0 (keyboard)	10	VCC
4	GND	11	VCC
5	RXD1 (keyboard)	12	VCC
6	GND	13	VCC
7	TXD1 (mouse)	14	VCC
		15	VCC

Keyboard/Mouse DB15 Connector

Serial Ports A and B			
Pin	Signal	Pin	Signal
2	TXD	8	DCD
3	RXD	15	DB
4	RTS	17	DD
5	CTS	20	DTR
6	DSR	24	DA
7	GND	25	-5V

Pinout of Serial Ports A and B

Pin	Signal	Pin	Signal
1	VIDEO+	6	VIDEO-
2	GND	7	GND
3	HSYNC	8	GND
4	VSYNC	9	GND
5	No connection		

Pinout of Monochrome Video Connector

(Video+ and Video- are at ECL voltage levels; HSYNC and VSYNC are at TTL voltage levels.)

Ethernet			
Pin	Signal	Pin	Signal
1	chassis ground		
2	E.COL+	9	E.COL-
3	E.TXD+	10	E.TXD-
4	chassis ground		
5	E.RXD+	12	E.RXD-
6	GND	13	+12V
7	Unused		

Pinout of Ethernet Connector

CPU Board Connector Pinout and Serial Port Signals

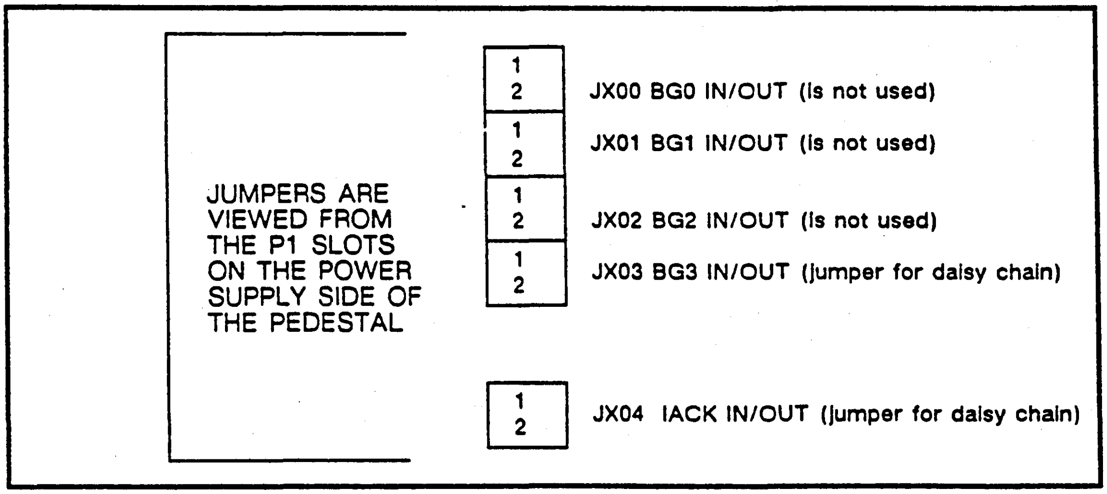
3.5 JUMPERING THE BACKPLANE

As mentioned earlier in chapters ONE and TWO, the backplane of the Sun 3/160M and Sun 3/160C and Sun 3/180FS must be jumpered appropriately to ensure that the Bus Grant3 In-to-Bus Grant3 Out (BG IN/OUT), and the Interrupt Acknowledge In-to-Interrupt Acknowledge Out (IACK IN/OUT) daisy chain is carried to the requesting slots.

As a rule of thumb, any board that wishes to do a transaction with the CPU board, must be granted the bus, and if it has sent the CPU an interrupt, the requesting board must also receive an interrupt acknowledge. To do this, if there are no empty slots between the requesting board and the CPU, the daisy chain will be passed from one board to the next. All Sun circuit boards pass IACK and BG3 with the exception of the memory expansion board and the FPA.

Also, if there is an empty slot between the requesting board and the CPU, the backplane must be jumpered to continue the daisy chain. Figure 3-8 illustrates the positioning of the jumpers.

Note the reference to jumpers as "X03" and "X04". The "X" refers to the slot number and will be different for each jumper installed, while the last digit defines the function of the jumper. Only BG3 (X03) and IACK (X04) are installed. BG 0-2 (X00 - X02) are not installed. It should be noted that the BG 0-2 jumpers come installed on the backplane from the factory. The signals, however, are not used. Looking fron the rear of the backplane, the jumper block to the **right** of the slot in question contain the jumper posts for that slot.



JUMPERS ARE VIEWED FROM THE P1 SLOTS ON THE POWER SUPPLY SIDE OF THE PEDESTAL
1
2
JX00 BG0 IN/OUT (Is not used)
1
2
JX01 BG1 IN/OUT (Is not used)
1
2
JX02 BG2 IN/OUT (Is not used)
1
2
JX03 BG3 IN/OUT (jumper for daisy chain)
1
2
JX04 IACK IN/OUT (jumper for daisy chain)

FIGURE 3-8: JUMPERING ON THE BACKPLANE

3.6 THE SCSI SUBSYSTEM

The SCSI (Small Computer System Interface) subsystem can be configured to three different levels, including a 1/4-inch tape drive alone, the 1/4-inch tape drive with a single disk (up to 327 MB formatted), or the 1/4-inch tape drive configured with a dual-disk (2 x 327 MB total formatted) mass-storage subsystem.

The SCSI subsystem consists of several major assemblies. They are: the SCSI Host Adapter (either a VME card or an On Board OBIO device), one or more disk controllers and a tape controller. The disk controllers may have up to two drives attached to each controller while the tape controller is restricted to one tape drive.

In a pedestal, we are restricted to one disk controller (with up to two drives) and one tape controller and it's drive.

In a "Shoebox" Mass Storage Option we will have one disk controller with one or two drives and a tape controller with it's drive. Note that if the twin disk "Shoebox" option is chosen, no tape drive will be present. This is due to space and power limitations.

SCSI is accessed on many different levels and programs. The way we access a drive under the **diag** utility is different from the way we access it under SunOs. All access to SCSI devices are by way of the SCSI Host Adapter. Under SunOS and in the Boot PROM, the individual controllers are not referenced. Only in the **diag** utility are the controllers accessed by number.

The SCSI Host Adapter is the VME card that responds to address 0x200000 or 0x140000 on 3/50 and 3/60. It is **not** an actual controller. The actual *controllers* are attached to this Host Adapter via the SCSI bus.

controller sc0 at vme24d16 ? csr 0x200000 priority 2 vector scintr 0x40

This concept allows us to communicate with many different SCSI *controllers* and their attached devices by referencing only **one** address; that of the SCSI Host Adapter.

The SCSI controllers are located in the SCSI tray (for pedestals) or in the "Shoebox" option. We are limited to two (2) controllers, each responding to a different address. The first *controller* will be address zero (0) and the second *controller* will be address one (1). These addresses are set on the *controller* by a jumper block or address dips depending on the controller manufacturer. If there are two controllers on the SCSI bus that respond to the same address, the bus will "lock up" and no data will be passed.

The SCSI standard provides for up to **eight** drives per controller. The disk controllers we use (Adaptec and Emulex) will access up to two drives each. This is why, if you look at a GENERIC kernel configuration file (Appendix A), you will see the SCSI drives numbered **0** and **1** then **8** and **9**.

disk	**sd0 at sc0 drive 0 flags 0**	**Controller 0**
disk	**sd1 at sc0 drive 1 flags 0**	**Controller 0**
disk	**sd2 at sc0 drive 8 flags 0**	**Controller 1**

SunOs can, therefore, access up to four SCSI drives using a standard set of device drivers. Since we are limited to two drives per shoebox, we designate the drives in the first shoebox to be drives **0** and **1**. Numbers **2** and **3** are not used. The second shoebox then contains drives **4** and **5** with nubers **6** and **7** not used. We have these "unused" numbers to preserve the continuity with the SCSI standard that supports eight devices on a "controller". Remember, the SCSI Host Adapter is **not** a controller, but the system, under SunOS, thinks it is.

SunOs uses device drivers located in the */dev* directory to access the disks. It refers to each partition on the disk by a unique number. For example the / or root of the SunOs file system is on drive **0** partition **a**. It is referred to as **sd0a** by SunOs. On a single drive standalone system, the */usr* partition is referenced by the driver **sd0g** which is the seventh partition on the drive.

For a more complete breakdown of disk partitioning please refer to the Installation & Networking Student Guide, or the System Administrators section of the SunOs document box.

The Sun 3 Boot Prom accesses these same four drives in a different manner. It refers to the drives attached to controller **0** as drives **0** and **1**. And the drives atached to controller **1** as **4** and **5**.

When we access the SCSI disks from the monitor, we usually do so during the boot process. If we were to attempt to boot from the *first drive* on the *first controller*, we would do so with the following command line: *>b sd (0,0,0)*

The first digit refers to the controller which in the case of SCSI is **always** the Host Adapter and will **always** be **0**. The second digit refers to the *drive* attached to the controller. The third digit refers to the partition on the disk to boot from. In the case of a tape drive it refers to the file number. We typically use partition **0** on drive **0** to boot from.

For example, this is how we would boot from the *root* partition of the *second* drive on the *second* controller. We would type in **b sd (0,5,0)** *program_name.*

To boot from the */usr* partition on the *second* drive of the *first* controller we would type **b sd (0,1,6)** *program_name.*

The DIAG program refers to controllers as **targets** and drives as **units**. The possible values for target are **0** and **1**. And the possible values for unit are also **0** and **1**. Again, this is the only time we will refer to the individual controllers by *number.*

To help sort out the correct way to access the drives please refer to the following chart and SCSI overview drawing.

	Controller 0		Controller 1	
Utility	Drive 0	Drive 1	Drive 0	Drive 1
DIAG	0	1	0	1
BOOT	0	1	4	5
SunOs	sd0	sd1	sd2	sd3

FIGURE 3-10 SCSI DEVICES and ACCESS NUMBERS

The SCSI overview chart on the following page, can be used as a reference guide for most questions. However, should you need detailed information as to the workings of the SCSI Host Adapter, please refer to Chapter 7-A of this manual.

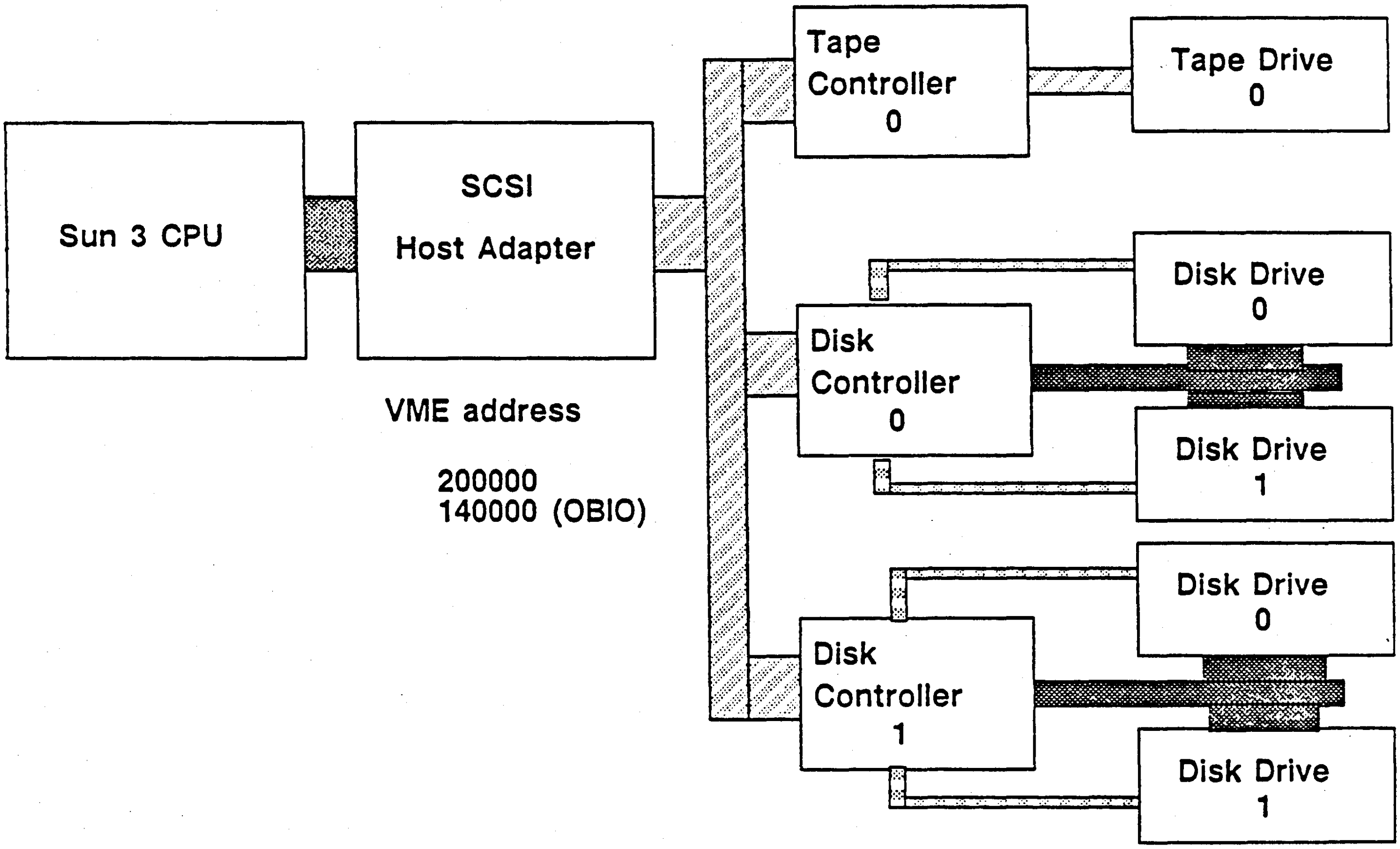


FIGURE 3-11 SCSI Technical Overview

As pointed out in the card configuration section, the SCSI interface will be configured in slot 7, due to the fact that the SCSI ribbon connects from this backplane position, to the subsystem located in the upper tray of the pedestal.

The following block diagrams illustrate the cabling layout for the SCSI subsystem.

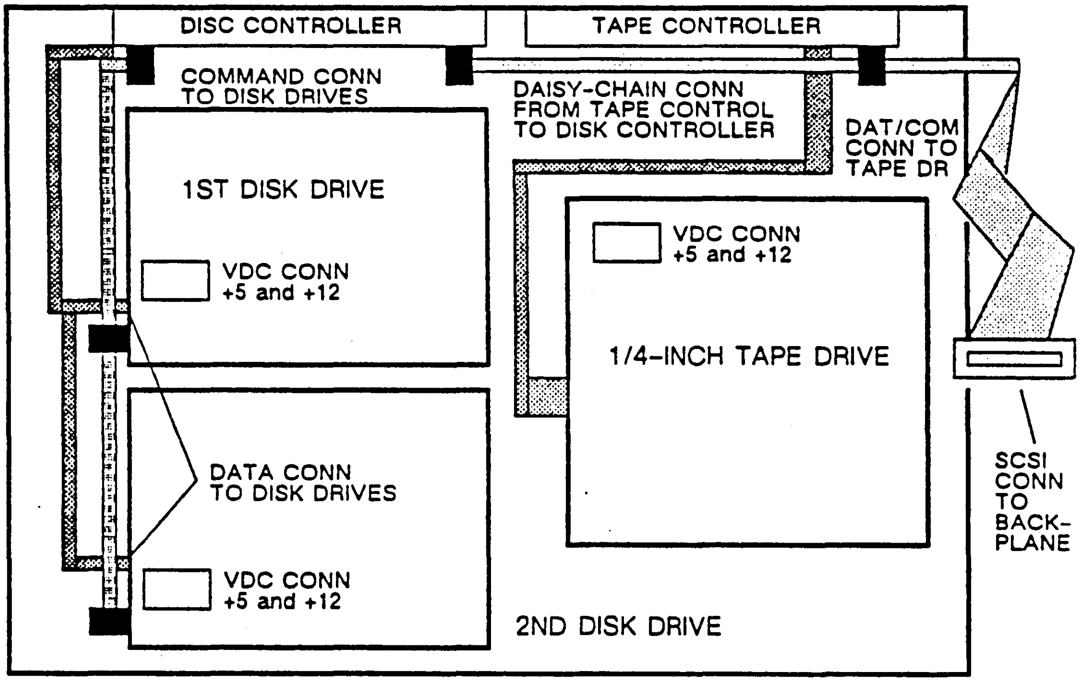


FIGURE 3-9: TOP VIEW OF SCSI SUBSYSTEM (old style)

The Tape controller has address and type switches. Emulex controllers have switches 3 and 5 on (Archive Tape Drive) or 3, 5 and 6 on (Wangtech Tape Drive). Sysgen controllers have switch 3 on ONLY.

Disk controllers also have address switches and jumnpers. Adaptec controllers have a jumper from "R" to "U". Emulex controllers have an address DIP. The switches should reflect the address of the controller. The first controller should have all switches off and the second controller should have switch #1 on.

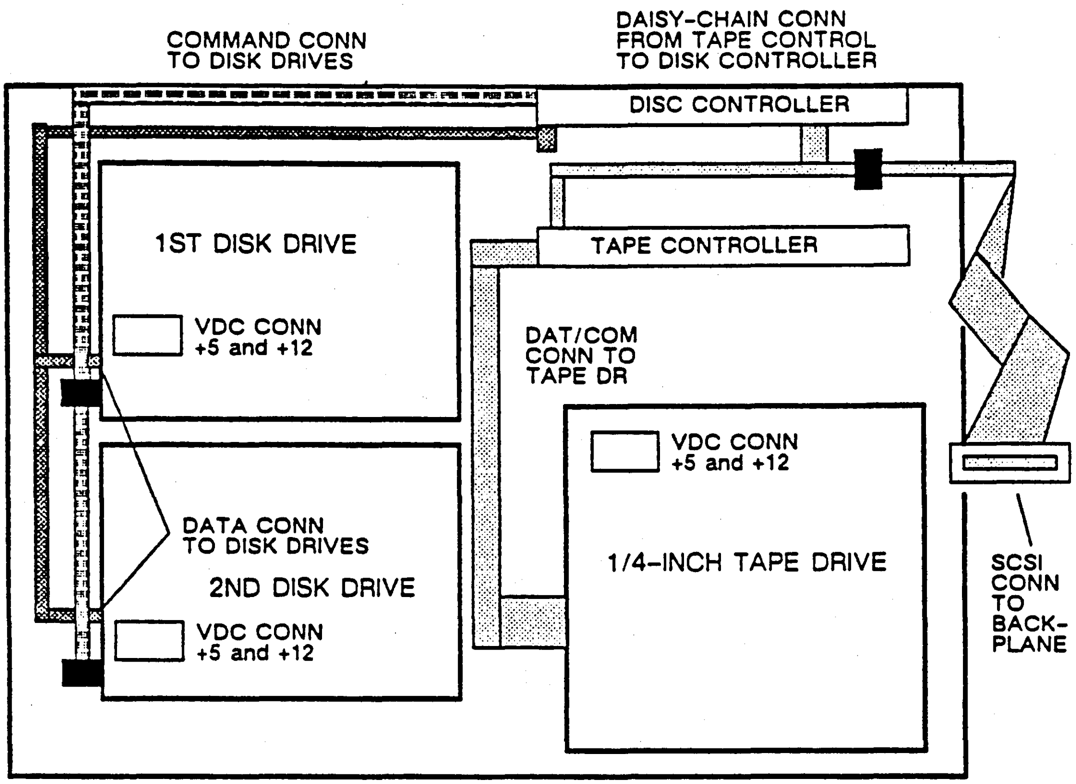


FIGURE 3-10: TOP VIEW OF SCSI SUBSYSTEM (new style)

Note how the positions of the tape and disk controllers is different in this version. To replace the disk controller, you will have to remove the tape controller and drive first, as access to the disk controller is blocked by the tape controller and the tape drive.

3.8 POWER SUPPLY OPTIONS ON THE WORKSTATION

There are three power supplies available for the Sun 3 workstation including the 150 Watt power supply for the Sun 3/75M, and two options of 850 Watt power supply for the Sun 3/160M and Sun 3/160C pedestals. The following blocks list the power supply input and output specifications for the Sun workstations.

GENERAL DESCRIPTION- the Sun 3/75M power supply requires 150 Watts of continuous single phase input, and a switching mechanism for 3 level outputs. This supply will serve the +5 VDC main output, a -12 VDC output and a +12 VDC output.

AC INPUT- 115 to 220 VAC(switch selectable) at a frequency range of 47 to 70 Hz

DC OUTPUTS-

+5 VDC at 25.0 amps +/-5%
+12 VDC at 1.5 amps +/-5%
-5.2 VDC at 1.5 amps +/-5%

FIGURE 3-11: SUN 3/75M POWER SUPPLY RATINGS

P/N	COMPONENT	AMPS +5 V	AMPS -5 V	AMPS +12 V	AMPS -12 V	TOTAL WATTS
501-1094	Sun 3 CPU	14.0	0.8	0.4	---	79.5
501-1096	Memory Exp 2MB	2.7	---	---	---	10.0
501-1097	Memory Exp 4MB	3.5	---	---	---	13.0
501-1059	VME 3/2 Adapter	1.6	---	---	---	8.0
501-1042	Backplane	0.5	---	---	---	2.5
501-1045	SCSI	3.2	---	---	---	16.0

FIGURE 3-12: SUN 3/75M BOARD POWER CONSUMPTION

GENERAL DESCRIPTION- the Sun 3/160M and the Sun 3/160C workstation pedestals will require 850 Watts of continuous power, including a single phase input, and switching mechanism for 3 levels of output servin a +5 VDC, -12 VDC and a +12 VDC output.

AC INPUT- nominal input 115 or 220 (switch selectable) at a frequency range of 47 to 70 Hz

DC OUTPUTS-

+5 VDC at 120 amps +/-5%
+12 VDC at 15 amps +/-5%
-12 VDC at 5 amps +/-5%
-5.2 VDC at 10 amps +/-5%

FIGURE 3-13: SUN 3/160M OR 3/160C POWER SUPPLY RATINGS

P/N	COMPONENT	Amps +5 V	Amps -5 V	Amps +12 V	Amps -12 V	Total Watts
501 -1094	Sun 3 CPU with 4 MB	13.5	0.8	0.4	---	77.0
501 -1097	Memory Exp 4 MB	2.5	---	---	---	12.5
501 -1055	Graphics Processor	20.0	---	---	---	80.5
501 -1020	Graphics Buffer	3.5	---	---	---	10.0
501 -1045	SCSI	3.0	---	---	---	15.0
501 -1014	Color board	15.3	5.3	---	0.2	105.4
501 -1053	Backplane	0.5	---	---	---	2.5
501 -1066	SMD with adapter	8.2	0.6	---	---	44.0
501 -1059	VME 3/2 adapter	1.6	---	---	---	8.0
501 -1054	VME/MB adapter	2.0	---	---	---	10.0
501 -1004	Sun Ethernet board	5.4	---	---	---	27.0
370 -0502	1/2-inch Tape controller	4.6	---	---	---	23.0
370 -1010	SCSI disk controller	1.8	---	---	---	9.0
370 -1034	85 MB disk drive	2.2	---	3.4	---	51.8
370 -1011	1/4-inch tape controller	2.6	---	---	---	13.0
370 -1037	1/4-inch tape drive	3.3	---	2.7	---	48.9
501 -1105	FPA	13.0	---	0.3	---	68.5

FIGURE 3-14: SUN 3/160 MODULE POWER CONSUMPTION

SYSTEM	AC FUSE RATING
PEDESTAL	15 amp slow-blow
BASE	3 amp slow-blow
COLOR MONITOR	5 amp slow-blow
GREY SCALE MONITOR	5 amp slow-blow
MONOCHROME MONITOR	1.6 amp slow-blow
ETHERNET	2 amp view
KEYBOARD	2 amp view

NOTE: For 220 Volt Systems cut Fuse value in half

FIGURE 3-15: SYSTEM FUSE RATINGS

3.9 ENVIRONMENTAL SPECIFICATIONS

The following blocks list the Environmental Specifications for the Sun 3 workstations. The user should pay close attention to the type and condition of the environment the Sun 3 workstation will be placed in.

Take special care to avoid placing any computer equipment in a high static environment. Electro Static Discharge (ESD) can be potentially hazardous to the Sun 3 workstation. Proper monitoring of the environment will keep the workstation intact.

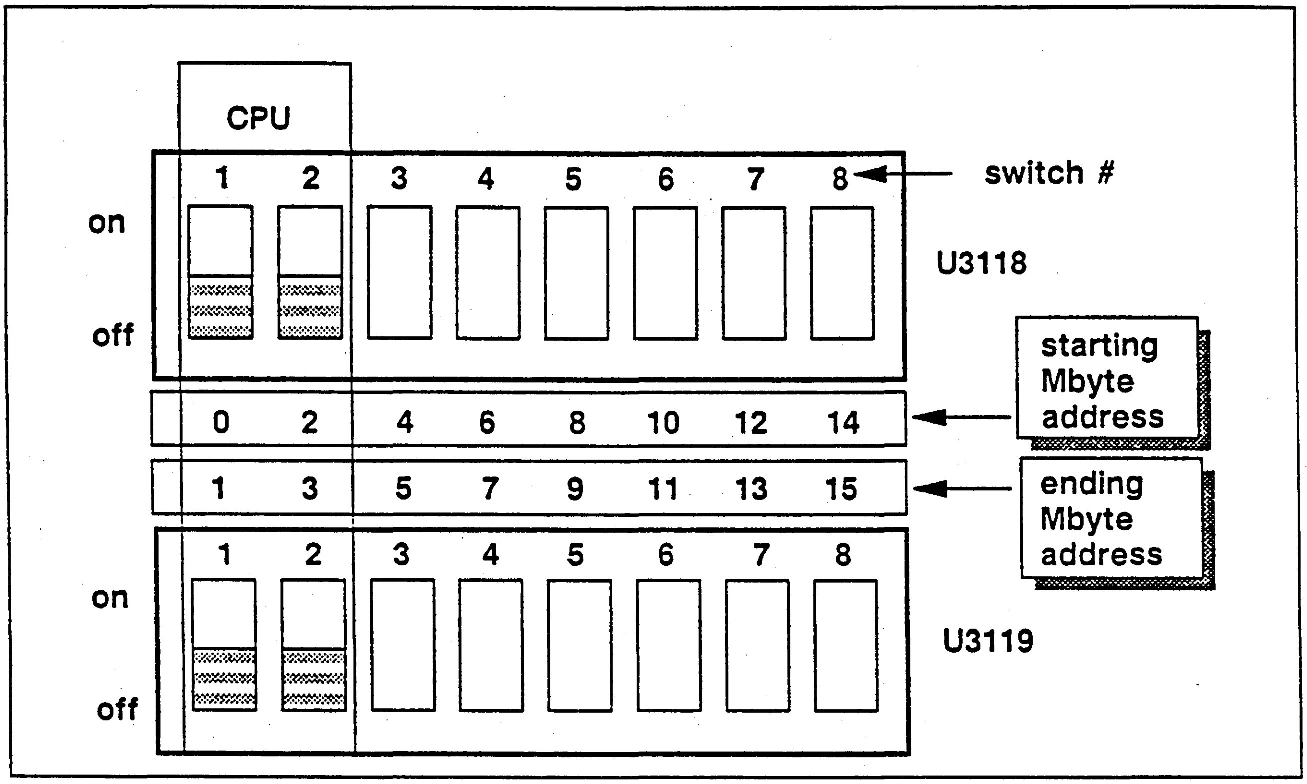
The user should also avoid placing the Sun workstation in a higher than normal RF environment. Especially if communication equipment is attached or configured with the workstation.

Refer to appropriate Industrial Standards specification manual for the placement of computer equipment.

3.10 SUN 3 MEMORY BOARD CONFIGURATION

The Sun 3 memory expansion board jumper configurations, set to factory default are listed in the following chart.

Memory Configuration Switches



The 1st 4 MB expansion:	U3118 sw# 3 on only
	U3119 sw# 4 on only
The 2nd 4 MB expansion:	U3118 sw# 5 on only
	U3119 sw# 6 on only

Two MB Examples

The 1st 2 MB expansion:	U3118 sw# 3 on only
	U3119 sw# 3 on only
2nd 2 MB expansion:	U3118 sw#4 on only
	U3119 sw#4 on only

Switches 1 and 2 (address 0-3 Mbyte) are always OFF because the first four Mbytes are on the CPU board.

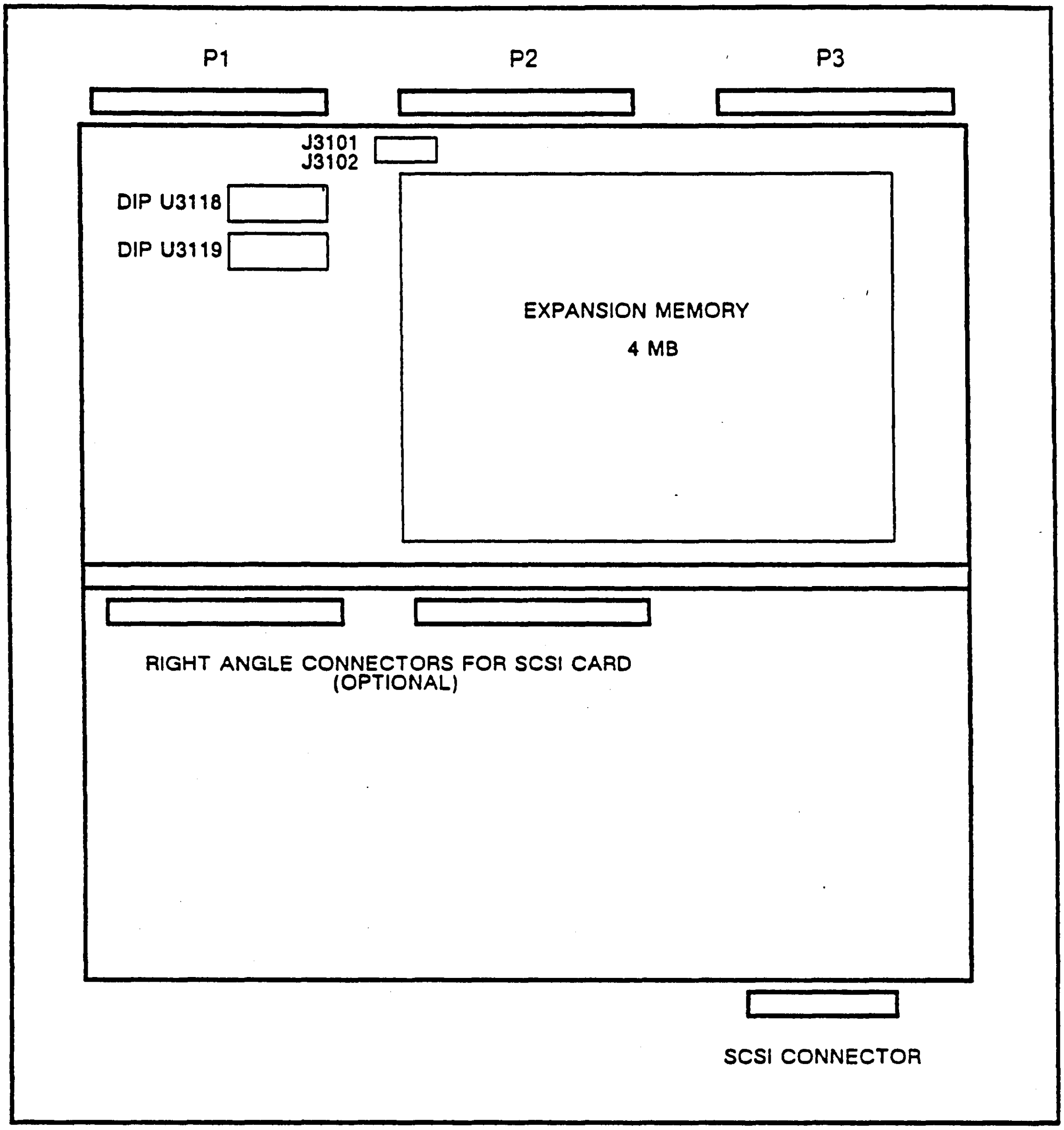


FIGURE 3-16: MEMORY EXPANSION CARD LAYOUT

Jumper J3101 in only if 2 MB on board (No longer in production)
Jumper J3102 in only if 4 MB on board

Dips U3118 and 3119 designate 1st and last MB on this board.

3.11 SUN 3/180FS RACK-MOUNTABLE FILE SERVER

The following section will describe the electrical, environmental and mechanical specifications for the Sun 3/180FS rack-mountable file server. The Sun 3/180FS, like all systems under the Sun 3 family of workstation, is a 68020 microprocessor- based, VME bus implemented system.

The system uses the same 1000 watt power supply found on Sun 3 pedestals, and conforms to the modular design of Sun workstations. This modular design includes a power supply tray, the main chassis and the card cage. The assembly and sub assemblies of these modules can be found in the assembly section following this chapter.

The Sun 3/180FS is designed to be built into a standard IEEE retma rail, 19-inch rack mountable system. The physical dimensions are as follows:

HEIGHT	WIDTH	LENGTH
19.133"	19.0"	20.0"

FIGURE 3-17: SUN 3/180FS RACK DIMENSIONS

3.12 PHYSICAL DESCRIPTION:

3.13 THE CHASSIS MODULE

The Sun 3/180FS Contains the 12 slot card cage and VME backplane that is used on the Sun 3 pedestal products. The design supports the 32-bit 68020 based architecture implemented with the triple height or double height VME cards (the double height uses a VME 3/2 adapter).

Since the backplane on the Sun 3/180FS conforms to the layout specified on the Sun 3 pedestal products, the same jumpering criteria for BG In/Out and IACK In/Out must be met for transactions on the VME bus. The jumpering specifications are found earlier in this chapter. The backplane and card current ratings are listed in the power consumption tables in this chapter.

3.14 THE FAN TRAY MODULE

The fan tray module provides the cooling for the cards configured in the chassis module. The fan tray is implemented with four mounted DC fans, which intake from the top of the module, and exhaust at the bottom of the chassis module. The design of the chassis module includes a perforated top and bottom, to allow for maximum airflow over the surfaces of the cards configured in the card cage, and is enhanced with convection cooling. The current ratings for the fans is contained in the power consumption charts in this chapter. Note that by using this method we avoid using air filters which can become clogged and inhibit air flow.

3.15 THE POWER TRAY MODULE

The power tray module is implemented with an 850 W power supply (Pioneer or ETA, see assemblies), the VAC receptacle, in-line fuse and line filter, and the power ON/OFF switch (labeled 0/1). Figure 3-19 lists the ratings for the power tray module.

VAC AMPS CIRCUIT BREAKER	The VAC recepticle is rated at 20 amps, as well as the line filter. The AC outlet must be twist locked at 30 A.
IN-LINE FUSE	The in-line fuse is rated at 10 amps for 110 VAC, and 5 amps for 220 VAC.
ON/OFF SWITCH	Will use a 12 V bulb and will be rated at 10-15 amps with a 90 to 270 VAC range.
SWITCHED RECEPTICLES	Will have a minimum of 8 switched 15 amp line recepticles
VAC STRAP RATINGS	98 VAC to 130 VAC RMS, at 47 to 63 Hz. Nominal input is 120 VAC RMS at 60 Hz.
	185 VAC to 265 VAC RMS, at 47 to 63 Hz. Nominal input is 240 VAC RMS at 50 Hz.

FIGURE 3-18: POWER REQUIREMENTS FOR VAC

SUN 3 CPU BOARD JUMPER FACTORY SETTINGS

JUMPER BLOCK	PINS	GRID LOCAL	INSTALLED	SIGNAL
J2502	1-2	A-3	YES	Enable VME CLK
J1200	3-4	A-3	NO	If 27256 BOOT PROM
J1201	5-6	A-3	YES	If 27512 BOOT PROM
J2501	1-2	B-6	YES	Enable Ethernet CLK
J2503	1-2	B-6	NO	+5 VDC to Ethernet
J2301	1-2	B-21	YES	Enable Video CLK
J1001	1-2	E-32	YES	Enable SCC CLK
J3101	1-2	K-11	NO	2 MB on CPU
J3102	3-4	K-11	YES	4 MB on CPU
J100	5-6	K-11	NO	Cache Disable
J2505	7-8	K-11	NO	Level 2 Ethernet
J400	1-2	N-11	YES	Enable 16.67 MHz CPU CLK
J400	3-4	N-11	NO	Enable 12.5 MHz CPU CLK
J400	5-6	N-11	NO	Enable 12.5 MHz FFP CLK
J400	7-8	N-11	YES	Enable 16.67 MHz FFP CLK
J300	1-2	S-5	YES	VME Interrupt level 0
J300	3-4	S-5	YES	VME Interrupt level 1
J300	5-6	S-5	YES	VME Interrupt level 2
J300	7-8	S-5	YES	VME Interrupt level 3
J300	9-10	S-5	YES	VME Interrupt level 4
J300	11-12	S-5	YES	VME Interrupt level 5
J300	13-14	S-5	YES	VME Interrupt level 6
J300	15-16	S-5	YES	VME Interrupt level 7
J2703	1-2	R-12	YES	Enable VME reset master
J2702	3-4	R-12	NO	Enable VME reset slave
J2701	5-6	R-12	YES	Enable VME request/arbiter
J2700	7-8	R-12	NO	Enable VME request only

NOTE: Pin "1" of jumper blocks can be located by looking on the silkscreen side of the board and looking for the 'V' block. Pins 1 and 2 will be at this end of the block.

FIGURE 3-20: SUN 3 CPU CARD JUMPERS

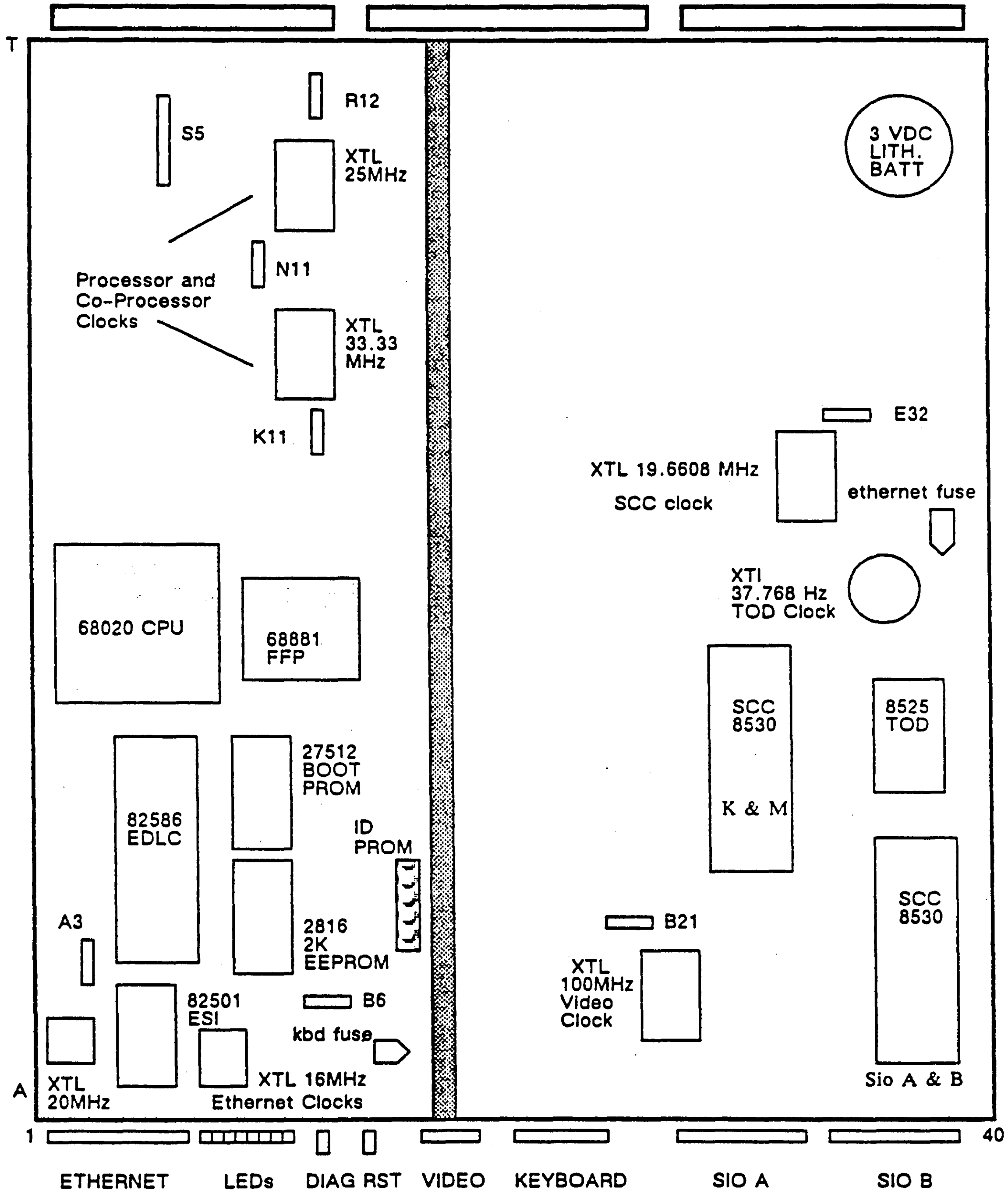


FIGURE 3-21: CPU CARD IC LAYOUT AND JUMPER LOCATIONS

3.16 SYSTEM OPTIONS FOR THE SUN 3/180FS

The Sun 3/180FS allows for extended configurations of both system cards and system peripherals. Since the 3/180FS is designed for file server applications, the system is typically configured with large capacity disc drives, high performance tape drives, and high capacity communication ports. The following table lists some of the numerous configurations available on the Sun 3/180 FS.

SUN 3/180FS CONFIGURATION OPTIONS
* 575 MB Hard Disk
* 1/2-inch 1600/6250 bpi tape drive
* Graphis Processor and Buffer
* Floating Point Accelerator
* 16 port ALM(Asynchronous Line Multiplexor
* Second Sun ethernet card for gateway
* Up to 16 M Bytes of memory
* Monochrome monitor
* Color monitor

FIGURE 3-19: SYSTEM OPTIONS

CHAPTER FOUR

SYSTEM DIAGNOSIS AND TROUBLESHOOTING

OVERVIEW

Sun Diagnostics are in four categories:
Prom based self tests
Prom based Extended tests
Tape based tests
Unix based tests

This chapter covers all Sun3 products using the 3/160 as an example. Minor differences in 3/50, 3/60, 3/110 and 3/200 are explained in their respective chapters.

4.1 PROM BASED SELF TEST

The Sun 3 architecture implements a number of devices which are utilized to provide support for diagnostic report. One such device includes the BOOT PROM, which contains resident diagnostic checkout routines, implemented with an external diagnostic switch on the CPU card.

Another device featured on the CPU card is the diagnostic display. This display includes an 8-bit diagnostic register and an eight LED display. The LED section of this chapter will illustrate how errors are reported to this device, and how the user can interpret their displayed patterns.

A user operable reset switch has also been included on the CPU card. This diagnostic switch, allows the user to set the system to a diagnostic state, thereby bypassing the normal forced BOOT FETCH sequence. The reset switch puts the system into a watchdog reset state if the CPU halts, which provides an automatic restart to initialize the processor during an system fault.

To implement these three diagnostic features, Sun 3 architecture provides for a readback of the state of the devices, which can be interpreted by the system via the firmware. The accompanying sections will define diagnostic states, error report, and will conclude with a practical troubleshooting flowchart.

4.2 RESIDENT BOOT PROM DIAGNOSTIC AND USER SWITCH

Sun 3 architecture implements a diagnostic side which is resident on the systems BOOT PROM. The Sun 3 BOOT PROM is designed to execute a power-up sequence of tests, as well as its normal BOOT sequence. In addition to these specifications, the BOOT PROM also provides the user a boot MONITOR which contains system debugger functions.

The code defined by the BOOT PROM contains the self test, the boot sequence, and the MONITOR functions. A small kernel is also included with a loader sequence program to support entry point code to find the appropriate boot device. The boot devices (in order of default priority) include the xy (for SMD) controller, the sd (for SCSI) and the ie (for booting over the net). Beyond these boot devices, the entry point will also allow for loading over from the mt or xt (1/2-inch) tape controller, and the st (SCSI 1/4-inch) tape controller.

There are two boot paths provided for by the BOOT PROM. One is the system software boot path, and the other is the diagnostic boot path. The two paths, then, will provide for the following functions:

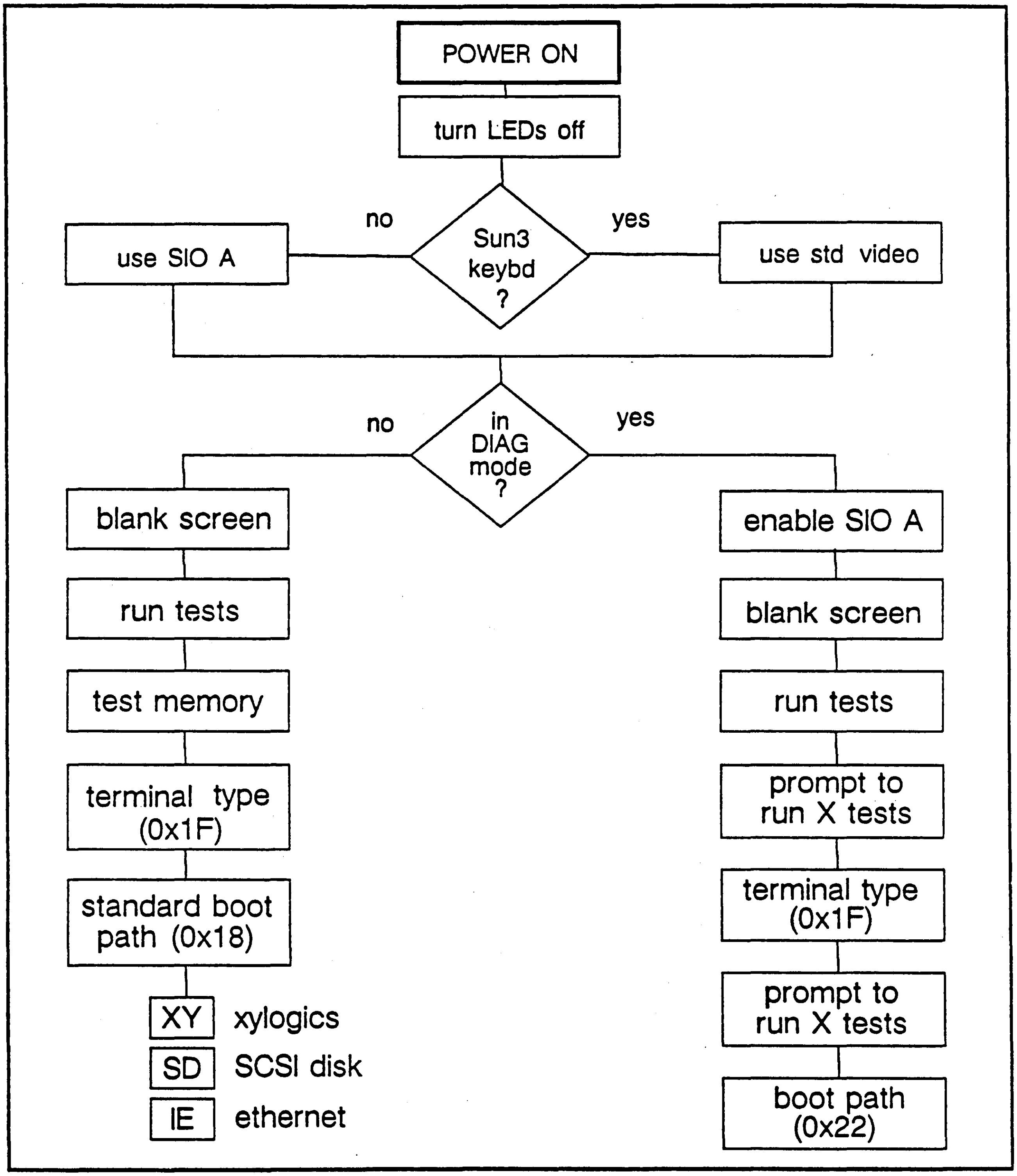
1. a system software boot path for normal default, which includes a quick self test followed by the booting of the UNIX operating system. The boot device will be specified by the user.
2. a diagnostic boot path which at the conclusion of executing the self test, will allow the user to specify additional tests for other hardware functions. At this time the user is prompted to input the desired action, or a default boot of an additional diagnostic is initiated.

FIGURE 4-1: TWO BOOT PATHS

4.1 THE SELF TEST PORTION OF THE BOOT PROM

The self test portion of the BOOT PROM will test CPU logic functions to the degree that a successful booting of the system software (operating system) will be executable. If the self test finds an error, the failing test number will be displayed by the corresponding LED's. It should be noted that if the user desires to redirect the displayed messages of the self test, the output is also piped to the serial ports. Therefore, the user can hook up an ASCII-type terminal to the SIO A port at the rear of the CPU card. If the system passes the self test, a system boot will execute if the external switch is set to NORMAL, indicating a normal boot path.

Boot Path Selection



4.1-1 Boot Path Selection Detail Flow Chart

4.4 DESCRIPTION OF THE SELF TEST DIAGNOSTICS

The self test diagnostics are a sequence of hardware 'crawl out' tests which include a selectable test menu and are executable under the MONITOR mode.

If the state of the CPU card is RESET, the processor is forced to fetch instructions from the BOOT PROM. At this time, the system enable register is reset, which will indicate the start of a minimal confidence sequence of the crawl-out tests. The crawl-out tests assume that the processor is working functionally and can therefore successfully fetch instructions from the BOOT PROM. During this portion of the test, the processor will not use any portion of memory (no stack) until it has successfully tested memory. The following indicates the state of the test:

1. If the diagnostic switch is OFF, the state is 0 which will select the UNIX boot path. Then,

   a. The self test will be executed,
   b. The loader will attempt to load UNIX.

2. If the diagnostic switch is ON, the state is 1 which will select the diagnostic path. Then,

   a. Self test will be executed,
   b. All memory found will be tested.
   c. Self test status information will be directed to SIO port A via the MMU bypass, until
   d. The devices needed for the video copy and video monitor has been successfully tested.

3. Any hardware failures found during the self test will invoke a scope loop.

FIGURE 4-2: BOOT PATH INSTRUCTIONS

To display the test on the ASCII-type device, the user should set the following characteristics:

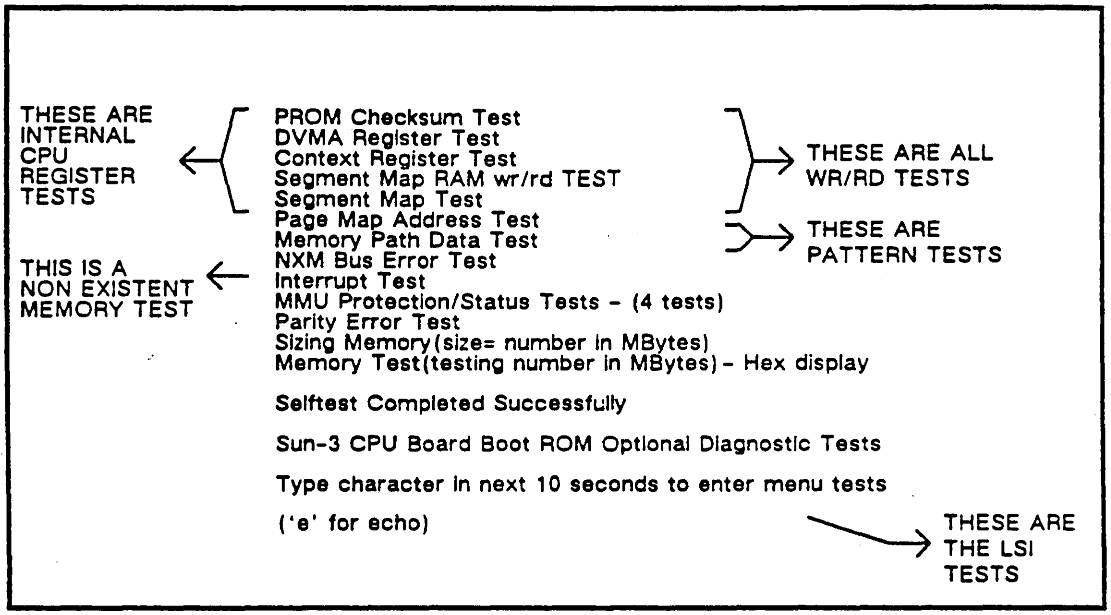
The terminal is set to 8-data bits, 1 stop bit, no parity, connected to SIO port A at 9600 Baud, one stop bit, no parity, full duplex using the null modem cable.

FROM PIN	TO PIN
2	3(cross)
4	5(cross)
7	7(straight through)

FIGURE 4-3: TERMINAL DISPLAY SETUP FOR SELF TEST

To execute the tests while in the diagnostic mode (indicated by the switch position) the system can be either power-cycled, or the user can input a K2 from the monitor mcde. A complete listing of the monitor mode inputs, as well as their use will follow in the next few pages.

After initiating the power-cycle or K2 reset, the following messages will be displayed on the terminals output screen:



```
PROM Checksum Test
DVMA Register Test
Context Register Test
Segment Map RAM wr/rd TEST
Segment Map Test
Page Map Address Test
Memory Path Data Test
NXM Bus Error Test
Interrupt Test
MMU Protection/Status Tests - (4 tests)
Parity Error Test
Sizing Memory(size= number in MBytes)
Memory Test(testing number in MBytes) - Hex display

Selftest Completed Successfully

Sun-3 CPU Board Boot ROM Optional Diagnostic Tests

Type character in next 10 seconds to enter menu tests

('e' for echo)
```

THESE ARE INTERNAL CPU REGISTER TESTS
THIS IS A NON EXISTENT MEMORY TEST
THESE ARE ALL WR/RD TESTS
THESE ARE PATTERN TESTS
THESE ARE THE LSI TESTS

FIGURE 4-4: SELF TEST MODE DISPLAY

As mentioned earlier, any test executing under the self test mode will loop on error. There is limited interaction which can be invoked if the test loops, and the user wished to continue. However, if a test is looped on error, as indicated by the LED pattern being displayed (to be discussed shortly), the user can alter the test by initiating the following inputs (remember that the following action will be invoked from the terminal):

Action taken during self test is entered as:

INPUT	RESULT
s	restarts the self test program
b	executes a burn-in test by repeating the test
<sp>	proceeds to the next test and is entered after the error looped test only

FIGURE 4-5: SELF TEST INPUT CHARACTERS

```
Self Test completed successfully.

     Sun Workstation, Model Sun-3/110LC or Sun-3/130LC, Sun-3 keyboard
     ROM Rev 1.8, 4MB memory installed, Serial #XX
     Ethernet address 8:0:20:1:XX:XX

Testing 4 Megabytes of Memory ...▨ Completed
```

The banner messages above are displayed on the monitor upon successful completion of the power-on self tests.

4.5 SELF TEST FAILURES, DISPLAY AND MEANING

During the execution of self test, the LED diagnostic display will play an important role in indicating the state of the system to the user. The 8 bit LED register (Sun 3 implementation) provided CPU status information by the patterns indicated on the LED display, and also include summary information about the individual tests themselves as indicated below:

LED bit 7- which indicates that the self tests have found an error.

LED bit 6- indicates an exception class error such as an illegal instruction, bus error, address error or unexpected interupt

LED bit 5- is refered to as the HEARTBEAT, and indicates that self test has completed successfully and that the processor is executing instruction(code) and is not hung.

LED 4 to 0- display the number of the test that is running, looping or stopped.

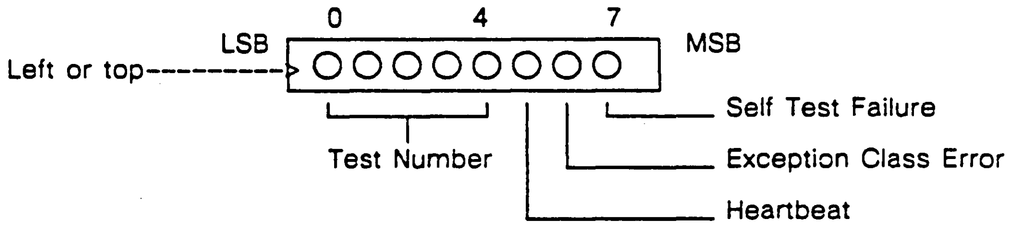


FIGURE 4-6: DIAGNOSTIC LED DISPLAY

(Note: 3/60 LED's indicators are reversed.)

The self tests execute in sequence and can be monitored by following the diagnostic display patterns on the LED display and comparing them to figure 4-8.

The final two tests executed under the self test mode are for memory. The memory checks consist of a memory sizing test, and then a memory test. The parameters for these two test can be specified (set) by the EEPROM. In NORM the amount of memory tested can be specified in location 0x15. However in DIAG mode, all memory found will be tested regardless of the value in 0x15.

The following chart illustrates the sequence of events that take place during the memory phase of the self test.

NOTE THAT BOTH MEMORY TESTS WILL TAKE LONGER THAN THE OTHER SELF TESTS AND WILL DELAY THE DISPLAY TO THE TERMINAL OUTPUT SCREEN.

1. SIZING MEMORY- will initialize memory with all F's for each long word above the trap and vector space and:

   a. size memory by detecting the timeout bus error from the first memory space and,
   b. display the memory size as Sizing memory (size = 000000n MBytes)
   c. tests all memory it finds(if diagnostic switch is set)
   d. if the switch is in the normal boot state, it will test only the amount of memory specified in the EEPROM configuration space.
   e. display the message as Memory Test (testing = 000000n MBytes)
   f. begins to test memory.

2. MEMORY TEST- begins by:

   a. initializes the context register, the segment maps, and the page maps with virtual mapped directly to physical addresses then,
   b. performs consecutive write then read then compare passes to the memory under test and,
   c. services any parity error interrupts.
   d. displays any memory errors found as

```
Error: addr 000000n, exp 000000n, obs 000000n
Memory parity err, addr 000000n, exp 000000n
```

FIGURE 4-7: SELF TEST DIAGNOSTIC FOR MEMORY

The following table lists the sequence of tests accomplished by the self tests, their display patterns, and the logic most likely causing the fault. Failure will be indicated with the test number latched up and the most significant bit latched on. Occasionally a test will indicate a failure by going into an infinite loop and not latching up MSB. This is indicated by a constantly repeating pattern in the LED display and (if enabled) a repeating pattern of tests on the ASCII terminal.

LSB LEDS MSB * = ON, 0 = OFF	WHAT THE SYSTEM IS DOING (LEDS CYCLING)	WHAT PART OF THE SYSTEM FAILED (LEDS FROZEN)
**** ****	A reset sets LEDs to this state	CPU, BOOT PROM,+5VDC
*000 0000	Test 1 checking the boot PROM	BOOT PROM
0*00 0000	Test 2 checking the DVMA register	CPU card
**00 0000	Test 3 checking the Context register	MMU section on CPU card
00*0 0000	Test 4 Segment map RAM w/r test	MMU section on CPU card
*0*0 0000	Test 5 checking Segment map RAM	MMU section on CPU card
0**0 0000	Test 6 checking Page map RAM	MMU section on CPU card
***0 0000	Test 7 checking memory data path	Resident or expansion memory
000* 0000	Test 8 checking bus error detection	CPU bus error register
00 0000	Test 9 checking interrupt capabilities	CPU card
0*0* 0000	Test 10 checking MMU read access	CPU card
**0* 0000	Test 11 checking MMU write access	CPU card
00** 0000	Test 12 writing to an invalid page	MMU section of CPU card
*0** 0000	Test 13 writing to a protected page	MMU section of CPU card
0*** 0000	Test 14 check does not make parity	Parity latch on CPU card
**** 0000	Test 15 check does make parity	Parity latch on CPU card
0000 *000	Test 16 initiates memory tests	Resident or expansion memory
0000 000*	Self test error report	CPU or expansion memory card
0000 00*0	Exception class error report	CPU , BOOT PROMS
0000 0*00	Self test done, Heartbeat	Waiting for boot
---*-**	Walking 1's pattern	UNIX executing successfully

**This chart does not include 3/200 reports. Refer to Table 10-15.

FIGURE 4-8: DIAGNOSTIC LED DISPLAY

If you are running to a standard bit map display at the successful conclusion of these self tests you will see the "heartbeat" LED flashing. If SunOS successfully boots you will see a "walking ones" pattern. If you do not have a display on the monitor you either have a bad monitor/cable or an incorrect EEPROM setting locations 0x16 or 0x1f (see Chapter 5).

NOTE THAT LEDS 4 THROUGH 0 WILL INDICATE WHICH TEST IS UNDER EXECUTION:

1. DIAGNOSTIC REGISTER TEST-this initial test loops through the LEDS and determines if the processor can fetch instructions from the boot PROM.

2. PROM CHECKSUM TEST- executes a checksum of the contents of the boot PROM and compares it to a stored value. If the test fails, the contents of the boot PROM cannot be assumed to be correct.

3. DVMA REGISTER TEST- writes then reads data from an address to/from the user DVMA register. This test checks the register for an expected value.

4. CONTEXT REGISTER TEST- writes then reads to and from the context register, and checks all data read for an expected value.

5. SEGMENT MAP RAM WR/RD TEST- performs write then read then compare tests to the segment map RAMS using consecutive values.

6. SEGMENT MAP ADDRESS TEST- writes then reads a shifting 1's pattern to every segment RAM address.

7. PAGE MAP TEST- sets the context to 0 and the segment maps to unity. (address 0 contains 0, address 1 contains 1, and on...), then performs consecutive write then read then compare passes of every page map RAM.

8. MEMORY PATH DATA TEST- tests the path between the CPU and DRAMS by writing then reading the lowest portion of memory with shifting 1's patterns(this is the first CPU to memory path test).

9. NXM(non existing memory) BUS ERROR TEST- tries to force a bus error by writing to non-existent memory, then checks each operation for the appropriate bus error message.

10. INTERRUPT TEST- sets the processor to enable all interrupts, then causes a level 1 interrupt and verifies that it occured.

11. MMU PROTECTION/STATUS TEST- a series of tests which checks the MMU protection and status functions as follows:

    a. read accesses a page and verifies that the access status bit is set for that page.
    b. write accesses a page and verifies that the access and modify bits are set for that page.
    c. read accesses an invalid page and verifies that a bus error occurs and the invalid access status bit in the bus error register is set.
    d. attempts to write to a write protected page, then verifies that a bus error occures and the protect error bit is set in the bus error register.

12. PARITY ERROR TEST- writes then reads to a specified address to verify that writing and reading does not cause a parity error. A second part of this test sets the parity test bit in the parity register so that it generates incorrect parity. A write then read will be executed in sequence to the specified address and will expect to generate a parity error. This is the last test reported to the LEDS.

FIGURE 4-9: SELF TEST DIAGNOSTIC MEANING

Sequence and number vary by model

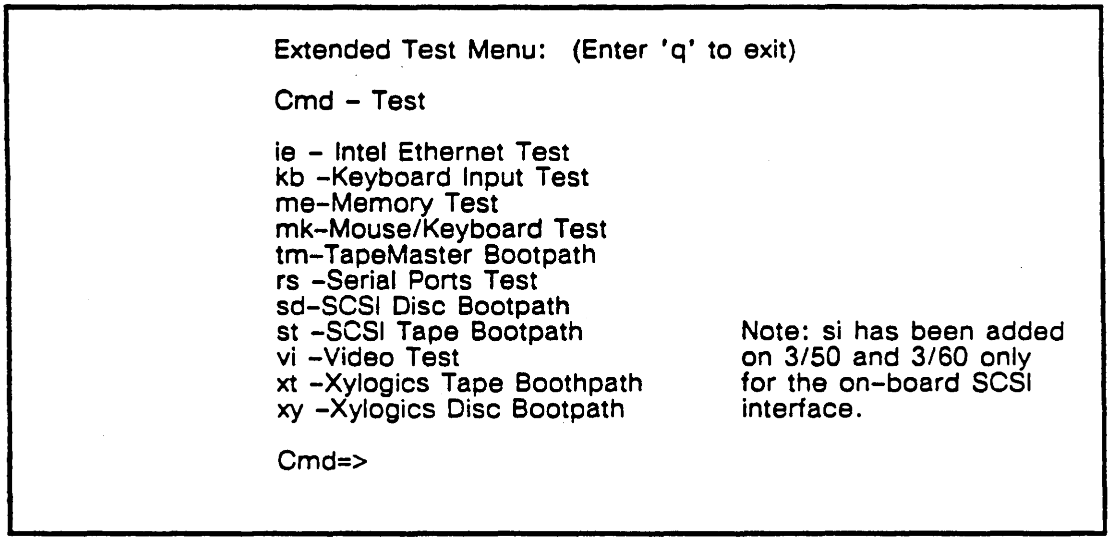
If you are connected to an ascii terminal in the DIAG mode, at the successful conclusion of the self test you should expect the following display:

```
Selftest Completed Successfully

Sun-3  Optional Menu Tests

Type a character within 10 seconds to enter Menu Test...(e) for echo...
```

The user can type in any character and depress the carriage return(<cr>). The boot PROM program will then display the following menu. Remember, we are dealing with the 3/75 CPU family here, if you are on a different machine such as a 3/50, turn to the proper chapter to be sure you can interpret the tests accurately.



```
Extended Test Menu:  (Enter 'q' to exit)

Cmd - Test

ie - Intel Ethernet Test
kb -Keyboard Input Test
me-Memory Test
mk-Mouse/Keyboard Test
tm-TapeMaster Bootpath
rs -Serial Ports Test
sd-SCSI Disc Bootpath
st -SCSI Tape Bootpath                 Note: si has been added
vi -Video Test                         on 3/50 and 3/60 only
xt -Xylogics Tape Boothpath            for the on-board SCSI
xy -Xylogics Disc Bootpath             interface.

Cmd=>
```

FIGURE 4-10: OPTIONAL MENU TESTS DISPLAY
Tests vary with model

4.6 USING THE OPTIONAL TEST MENU

The optional test menu allows the user to individually test the MOS and LSI device logic. These menu tests will only appear on the present level of Sun 3 products, and are subject to change for later releases. To use the menu, enter the desired command. The individual circuit tests are numbered and listed on the following pages:

1. THE INTEL ETHERNET TEST- This test is provided for the Intel 82586 Ethernet Data Link Controller(EDLC), and for the Intel 82501 Ethernet Serial Interface(ESI). This test checks the pathway between the back end and the front end, respectively. The following test modes are made available(displayed) by entering the ie command after the Cmd=>

```
          l -Local loopback

          e -Encoder loopback

          x -External loopback

     Cmd=>
```

The user enters the command ot the test noting the following:

a. in the local loopback mode the 82586 disconnects itself from the 82501 and performs an internal loopback within the EDLC.

b. in the Encoder loopback mode the 82501 loops back transmit to receive data internally. This tests both devices.

c. in the External loopback mode, an external terminator should be attached to the external Ethernet connector. Both devices and the connector will be tested.

External test will pass if machine is connected to a working, terminated net. You can make a wrap connector by connecting pins 3-5 and 10-12 at the Ethernet connector on CPU board or at the end of the cable to the transceiver.

(NOTE: see connector specifications on following page.)

When the operator enters the test command and enters a carriage return additional test menues are displayed as follows:

```
Test Options: (Enter 'q' to exit)

Cmd - Option

f -Loop forever
h -Loop forever with Halt on error
l  -Loop once with loop on error
n -Loop forever with error message inhibited
<cr> -Loop once
```

If the user enters the test, the following messages are likely to appear with the test and at the conclusion of the test:

```
(selected) test failed
(selected) test passed
```

If the test continues to fail after repeated trys, the CPU card is in fault and the user should exchange the CPU card after double-checking parameters.

FIGURE 4-11: THE INTEL ETHERNET TEST

THE MOUSE/KEYBOARD TEST can be entered at the prompt by invoking the mk command at the CMD=>

The following display will result:

```
Mouse/Keyboard Port Tests:
Enter port cmd: Cmd{port(M or K)}  {Baud rate decimal} {Hex byte pattern}

                      Cmd - Test

                      w -Write/Read SCC Reg 12
                      x -Xmit char
                      i -Internal loopback
                      e-External loopback

                      Cmd=> (enter selection here)
```

This selection will prompt the following display:

```
                    Test Options: (Enter 'q' to exit)
                      f -Loop forever
                      h -Loop forever with Halt on error
                      l -Loop once with loop on error
                      n-Loop forever with error messages inhibited
                      <cr> -Loop once

                      Cmd=> (enter selection here)
```

Examples for the Mouse/Keyboard test can be write then read tests of the SCC(8530) internal register 12 of channel k, with a data byte of 41(hex) for addressing and data to/fm the 8530 as :

```
Cmd=> w k 41 <cr>
```

An example of transmitting an ASCII character from the mouse port at 9600 baud would be entered as:

```
Cmd=> x m 9600 (ASCII value) <cr>
```

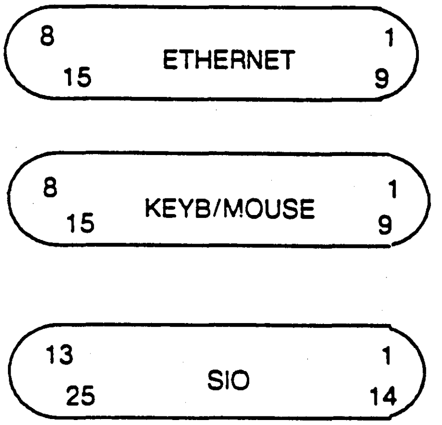
Results of these tests would be seen as:

```
(selected) test failed
(selected) test passed
```

At the the time of failure, the user will want to exchange the CPU card if the keyboard an mouse ports were not functional.

External loopback connections are: Pin 1 to 3 and 5 to 7.
Remember to check the fuse or look for the illuminated LED on the bottom of the mouse. If the LED is not on, the fuse may be bad.

Figure 4-11.0: THE KEYBOARD/MOUSE TEST



Jumper pins 3 to 5 and 10 to 12. (does not check +12VDC fuse)

Jumper pins 1 to 3 and 5 to 7. Check fuse.

Jumper pins 2 to 3, 4 to 5, and 6 to 20.

FIGURE 4-11.1: LOOPBACK CONNECTOR SPECIFICATIONS, VIEWED FROM MACHINE END

2. KEYBOARD INPUT TEST -this test can be initiated by entering the kb command after the Cmd=>

After enteing the kb command and entering a carraige return the following display will result:

```
Type keyboard keys.  To exit enter 'ESC' or ;
```

The user can now depress all the keys on the workstations keyboard. The video display or serial port output will display the ASCII code of the depressed character, followed by the character itself. As an example, if the user depresses the 'a', 'b', 'c' the display would show:

```
61 a
62 b
63 c
```

If there is an error, the workstation keyboard will not work. At this point the user will want to first double check the kb connection(check the cable), and then check the connections. If the keyboard does not pass this test the user should proceed to exchange the keyboard,cable or the fuse on the CPU board.

Do not plug Keyboard in with power on you will blow the fuse or keyboard

FIGURE 4-12: THE KEYBOARD INPUT TEST

3. THE MEMORY TEST - the memory test can be initiated by entering the me command after the Cmd=>

```
me <cr>
```

There are two types of memory tests:

a. a write/read data test where the user specifies a longword data pattern throughout the low and high address limits.
b. an address test in which the long word address is written as data at the long word address for each address within the low to high address limits.

Next a submenu is displayed as:

```
Enter Cmd {low address >=0} {high address <=0xNNNNNN} {pattern}
                     Cmd - Test
                     a - Address Test
                     c - compare
                     s - scan
                     w - Write/Read data pattern
                     Cmd=>(enter selection here)
```

This will prompt the display of another submenu as follows:

```
                Test Options: (Enter 'q' to exit)
                     Cmd - Option

                     f -Loop forever
                     h -Loop forever with Halt on error
                     l -Loop once with Loop on error
                     n -Loop forever with error messages inhibited
                     <cr> -Loop once

                     Cmd=> (enter selection here)
```

An example of the test would be to write/read memory from 0 to hex 1000 with a data pattern of 0xFFFFFF. This would be entered as:

```
Cmd=> w 0 1000 FFFFFF <cr>
```

Another example would be for the user to enter for a write/read address pattern at the same address. Here you would enter the following:

```
Cmd=> a 0 1000
```

Any resulting errors would prompt a display as follows:

```
(selected) test failed
(selected) test passed
```

The user would then exchange the CPU card(resident memory) or the memory expansion card depending on the reported range of error under test.

FIGURE 4-13: THE MEMORY TEST

4. THE SERIAL PORT TESTS -The serial port test can be entered at the prompt by invoking the rs command after the Cmd=>

The following display will result:

```
Serial Port Tests
Enter port cmd: Cmd{port(A or B)} {Baudrate(decimal #)} {hex byte pattern}

                       Cmd - Test

                       w -Write/Read SCC Reg 12
                       x -Xmit char
                       i -Internal loopback
                       e-External loopback

                       Cmd=> (enter selection here)
```

This selection will prompt the following display:

```
                    Test Options: (Enter 'q' to exit)
                       f -Loop forever
                       h -Loop forever with Halt on error
                       l -Loop once with loop on error
                       n-Loop forever with error messages inhibited
                       <cr> -Loop once

                       Cmd=> (enter selection here)
```

Examples for the serial port test can include a write then read test of the SCC(8530) internal register 12 of channel A, with a data byte of 41(hex) for addressing and data to/fm the 8530 as :

```
Cmd=> w a 41 <cr>
```

An example of transmitting an ASCII character from SIO B at 9600 baud could be entered as:

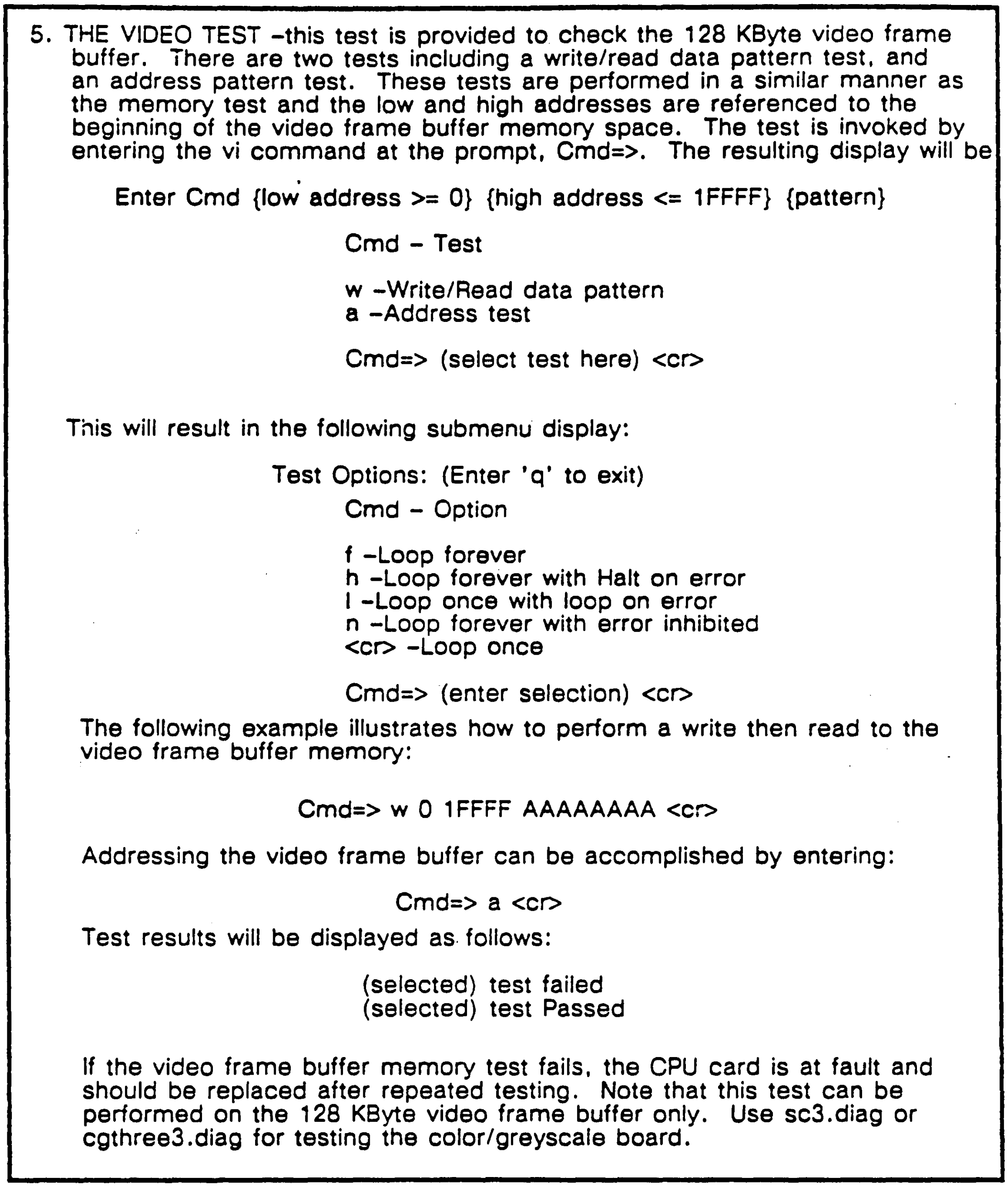
```
Cmd=> x b 9600 (ASCII value) <cr>
```

Results of these tests would be seen as:

```
(selected) test failed
(selected) test passed
```

At the the time of failure, the user will want to exchange the CPU card if the SIO ports and SCC controllers were not functional. Note that the proper operation of SIO port A is indicated by the users ability to use an ASCII terminal at the time of the test. Always double check the terminals set-up parameters as well.

FIGURE 4-14: THE SERIAL PORT TESTS



5. THE VIDEO TEST -this test is provided to check the 128 KByte video frame buffer. There are two tests including a write/read data pattern test, and an address pattern test. These tests are performed in a similar manner as the memory test and the low and high addresses are referenced to the beginning of the video frame buffer memory space. The test is invoked by entering the vi command at the prompt, Cmd=>. The resulting display will be

```
Enter Cmd {low address >= 0} {high address <= 1FFFF} {pattern}

                    Cmd - Test

                    w -Write/Read data pattern
                    a -Address test

                    Cmd=> (select test here) <cr>
```

This will result in the following submenu display:

```
          Test Options: (Enter 'q' to exit)
                Cmd - Option

                f -Loop forever
                h -Loop forever with Halt on error
                l -Loop once with loop on error
                n -Loop forever with error inhibited
                <cr> -Loop once

                Cmd=> (enter selection) <cr>
```

The following example illustrates how to perform a write then read to the video frame buffer memory:

```
Cmd=> w 0 1FFFF AAAAAAAA <cr>
```

Addressing the video frame buffer can be accomplished by entering:

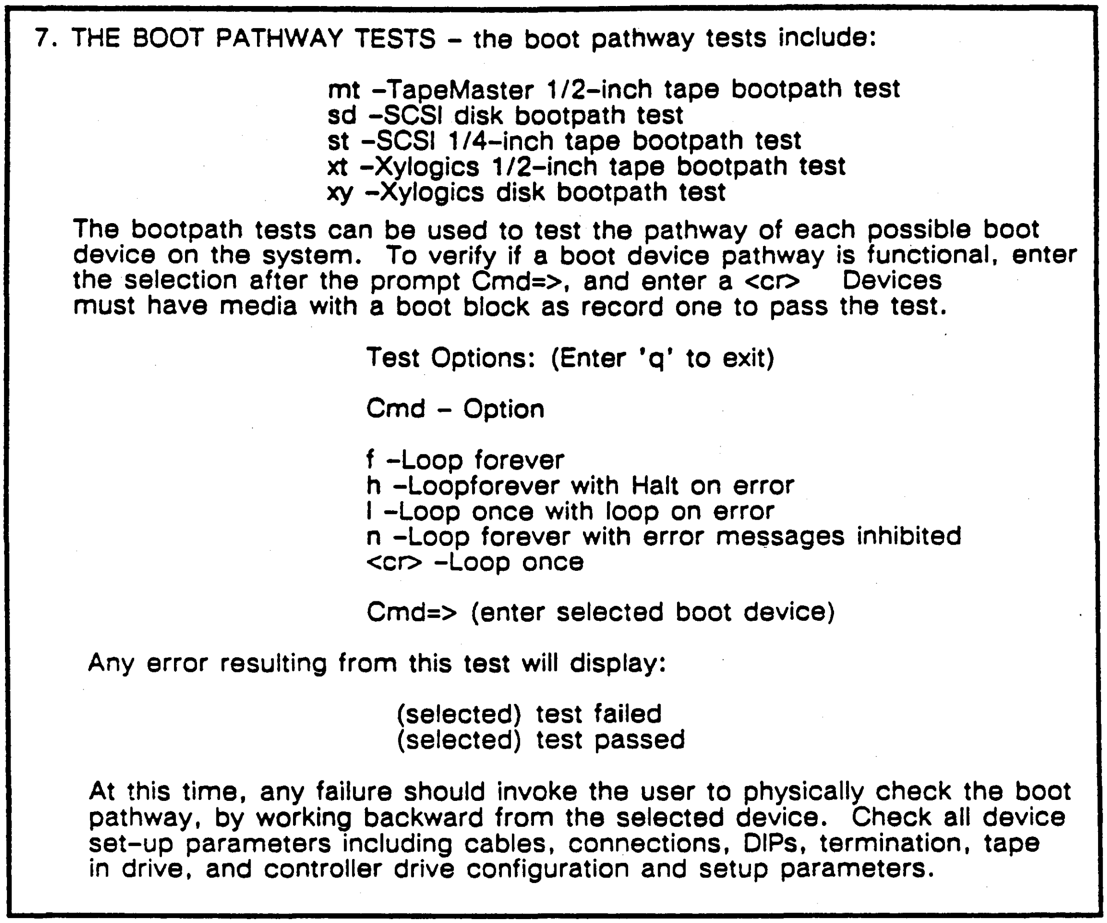
```
Cmd=> a <cr>
```

Test results will be displayed as follows:

```
(selected) test failed
(selected) test Passed
```

If the video frame buffer memory test fails, the CPU card is at fault and should be replaced after repeated testing. Note that this test can be performed on the 128 KByte video frame buffer only. Use sc3.diag or cgthree3.diag for testing the color/greyscale board.

FIGURE 4-15: THE VIDEO FRAME BUFFER MEMORY TEST



7. THE BOOT PATHWAY TESTS - the boot pathway tests include:

mt -TapeMaster 1/2-inch tape bootpath test
sd -SCSI disk bootpath test
st -SCSI 1/4-inch tape bootpath test
xt -Xylogics 1/2-inch tape bootpath test
xy -Xylogics disk bootpath test

The bootpath tests can be used to test the pathway of each possible boot device on the system. To verify if a boot device pathway is functional, enter the selection after the prompt Cmd=>, and enter a <cr> Devices must have media with a boot block as record one to pass the test.

Test Options: (Enter 'q' to exit)

Cmd - Option

f -Loop forever
h -Loopforever with Halt on error
l -Loop once with loop on error
n -Loop forever with error messages inhibited
<cr> -Loop once

Cmd=> (enter selected boot device)

Any error resulting from this test will display:

(selected) test failed
(selected) test passed

At this time, any failure should invoke the user to physically check the boot pathway, by working backward from the selected device. Check all device set-up parameters including cables, connections, DIPs, termination, tape in drive, and controller drive configuration and setup parameters.

FIGURE 4-17: DEVICE BOOTPATH TESTS

8. THE SCSI INTERFACE TEST – the interface test includes:

```
Cmd=> – si

SI = SCSI INTERFACE TEST
```

The user will be prompted to select from the following menu:

```
b – SCSI byte ctr wr/rd test
c – SCSI csr register wr/rd test
f – 8X60 fifo/udc dma test
s – 8350 sbc chip wr/rd test
u – 9516 udc chip wr/rd test
x – SCSI bus external loopback test
```

This will prompt the user to the next level menu as follows:

```
Test Options: (Enter 'q' to exit)

f – Loop forever
h – Loop forever with halt on error
i  – Loop once with loop on error
n – Loop forever with error messages inhibited

Cmd=> (enter selection here)
```

An example for the SCSI byte crs wr/rd test would be:

```
Cmd=> b <cr>
```

And the results would indicate test failed, or test passed. Note that a test failure on on-board SCSI interface tests would require the user to exchange the CPU card. Also note, that an external SCSI loopback cable and boot proms REV 2.6 or higher are required for this test

In addition to the Rev 2.6 boot proms, a loopback connector must be constructed using the following connections: Pin 38 to 16, 36 to 14, 44 to 12, 32 to 10, 48 to 8, 42 to 6, 46 to 4, 50 to 2 and 40 to 18.

This loopback connector is attached to the SCSI bus connector at the handle edge of the CPU board.

FIGURE 4–18: 3/50 ON BOARD SCSI INTERFACE TEST

4.7 USING THE MONITOR COMMANDS

The Sun-3 boot PROM contains a group of coding that is referred to as the MONITOR. The MONITOR section of the boot PROM provides the user with a command interface to the CPU card. You must be in the MONITOR mode to execute MONITOR commands. They will not work from SunOs. To enter the MONITOR mode from SunOs, use the sequence sync, halt to come down gracefully from SunOS.

While at the monitor level, the user can perform tasks which include calling up various device boot paths, redirecting the system console to the serial ports, examining memory locations, mapping registers, modifying the contents of a processor register at a specified value, clearing the modified registers, and invoking the optional diagnostic menu.

Once the user is at the monitor level, a > (greater than) prompt will be displayed. To invoke a monitor command, a specific monitor syntax must be entered as follows:

```
THE MONITOR SYNTAX:

{verb} space {argument} <cr>

AS REPRESENTED AS:

verb= a singular alphabetic character(see following list)
space= there should be a space between the character and
        the argument
argument= normally a hex number(this is sometimes optional
            and the user should refer to the following usage
            table)
<cr> = indicates that the entry must be followed by a
        carriage return

Note that the backspace or the delete key can be depressed
to erase a character. The Control U (^U ) can be entered to
erase the whole line. The following example specifies an
argument to invoke all console control to be piped to the
SIO A port

                    > u a <cr>
```

FIGURE 4-19: USING THE PROM MONITOR

The current address or register value and location can be invoked by the query functions at the monitor level. To access the monitor function table, a help command has been added to allow user interaction. To call it up to the display, the user enters an 'h' after the > (greater than prompt) and finishes the entry with a carriage return (<cr>) as follows:

```
> h <cr>
```

FIGURE 4-20: CALLING UP THE MONITOR HELP MENU

Once this line has been entered, the following table will be displayed onto the video or terminal screen.

CMD	ARGUMENT PARAMETER	ARGUMENT DESCRIPTION
a	number	open CPU addr register 7-0
b	device(control,unit,file)	bootload a device
c	address(hex)	continue process at this address
d	number	open CPU data register 7-0
e	address(hex)	open 16-bit word at memory addr
f	start addr,end addr,patrn,size	fill memory with the following pattern
g	address(hex)	start process subroutine at this addr
h	character	display the monitor help menu
k	number	RESET: 0=CPU 1=Control Space 2=System b=display banner
l	address(hex)	open 32-bit long word at mem addr
m	address(hex)	open segment map entry
o	address(hex)	open 8-bit byte location specified
p	p(virtual address)	open page map entry
q	address(hex)	open the EEPROM addr in addr space
r	hex or ?	open CPU registers(see following table)
s	number	sets or queries function code space
t	character(y,n,c)	sets trace mode
u	character(a,b)	pipes console to SIO ports
v	start addr,end addr,size	displays contents of addr(b=byte,etc)
w	hexadecimal	sets vector value
x	character	calls up diagnostic options menu
z	hexadecimal	sets the breakpoint

FIGURE 4-21: MONITOR HELP MENU DISPLAY

The following section explains how to use the monitor commands, and how to clear the system to its default state. A register table has also been added for sequencing access to the registers. Note that the boot devices were listed in earlier sections.

Specific arguments and sequence of entry must be followed closely to ensure that the monitor level is utilized correctly. The following table lists the value and type of register associated with the CPU.

REGISTER	DESCRIPTION
D6-D0	The data registers
A6-A0	The address registers(A7=SS)>
IS	The Interrupt Stack Pointer
MS	The Raster Stack
US	The user stack pointer
SF	The source function code register
DF	The destination function code register
VB	The vector base register
CA	The Cache Address
CC	The Cache Control
CX	Context
SR	The status register
PC	The program counter

FIGURE 4-22: THE CPU ADDRESS AND REGISTER NAMES

4.8 CALLING UP MONITOR ARGUMENTS

This section will include a brief listing of MONITOR command descriptions. Note that the user should use all arguments carefully, and clear the system with a K1 or K2 reset to ensure that all entered monitor inputs are erased for normal booting. Further note that the entry of a question mark (?) specifies that this entry on the command line means READ-only.

The A command opens the CPU address registers 7 through 0. Note that A7 is the system stack pointer. Each command entered will include the A cmd, followed by the number specifying the register you wish to access. Remember to enter a carriage return (<cr>) at the end of each entry.

The B command sets the loader in the boot PROM to the desired device driver, where the device can be ie, sd, st, mt, xt or xy (see the boot device description in the boot PROM section). The argument for the B command can be c (controller number 0 or 1), u (unit number 0 or 1), and p (specifying the file number). The B command resets the appropriate sections of the system, and then initiates the bootstrap process. The arguments for the B command initiates the loader program to search for the device controller, the device unit, and the file specified for booting.

Note that entering a b? by itself causes a menu of possible devices to be displayed. Further note, that when the user tries to call up the SunOS operating system with just the b command, the boot file may not be found. System boots of the operating system can be called up as follows:

```
> b vmunix
```

This entry at the monitor mode will search for the vmunix (virtual memory unix) file and initiate the bootstrap of the operating system.

Another useful version of the *b* command is b*. When *b* is followed by an asterisk, the contents of the boot device's root directory will be displayed. This is useful when you suspect someone has renamed or removed vmunix and you wish to inspect the partition. Note that this list will continue to be printed and you must *L1-A* to stop it.

The *c* command executes a continue program sequence. The address, if given restores the program. If the address is not given, the system defaults to the current value of the program counter. If the user enters a L1-A, this will put the user at the MONITOR level. Now if the user enters the *c* command, the system will return to the last process executed before the *L1-A* entry (last process recognized by the program counter.

The *d* command opens the CPU data registers D7 through D0. This command is provided to examine and modify any or all of the 8 data registers within the 68020.

The *e* command opens a 16-bit word at the specified memory address (default is zero) in the address space specified by the *s* command.

The *f* command can be used to fill memory address space, from the lower address to the higher address, with a constant parameter specified by the pattern size. The pattern size can be b (byte), w (word) or l (long word). The default for this pattern is long word (if the size is not specified).

As an example, an address block from 0x1000 to 0x2000 with the word pattern of 0xABCD can be entered as:

```
> f 1000 2000 abcd w <cr>
```

FIGURE 4-23: USING THE F COMMAND

The *g* command starts a program by executing a subroutine call to access a specified address, or else the default value of the current program counter will be given. An example of this would be calling up a memory location of a process during the loading of the operating system, as defined below:

Here the user was loading from the st(SCSI tape) at location:

```
BOOT:st(0,0,1) <cr>
```

Instead of hitting a carraige return, however, the program is somehow aborted. Her the user can return to the process location by using the following entry:

```
> g 4000 <cr>
```

This will call up the current location from the program counter, and the user will be able to continue the loading procedure.

FIGURE 4-24: USING THE G COMMAND

The *h* command simply calls up the system to display the menu of MONITOR commands, their arguments, and their definitions.

The *k* command indicates a reset instruction. If a *k0* (or no argument) is given a reset instruction is executed. The *k1* command does a "medium reset" which reinitializes most of the system, without clearing memory. The *k2* command is a "hard reset", which reinitializes the system and clears memory. The *k2* reset is the equivalent to a POR (power-on-reset). The *k2* also is used in the diagnostic mode to call up the self test and the diagnostic options menu. A total system reset can take 5 to 180 seconds, depending on how much memory is tested. Finally, the *kb* command can be used to display the banner messages.

The *l* command opens a 32-bit long word at the specified memory address (default is zero) in the address spaced defined by the *s* command.

The *m* command opens a segment map entry, which maps a virtual address (default is zero) in the current context. The segment map address is the virtual address field from the address bits of the CPU as presented to the MMU. The current context is determined by the *s* command supervisor or user.

The *o* command opens an 8-bit byte location specified by the address space, which is defined by the s command. The default is zero.

The *q* command opens the EEPROM address within the EEPROM address space. This command is used to define the EEPROM parameters and is described in Chapter FIVE.

The *r* command is used to open the register of the CPU, as described in Figure 4-21. Alterations to these registers do not take effect (with the exception of SC and UC) until the next *c* command is executed.

The *s* command sets or queries the address space to be used by the memory access commands. The number of the argument is the Function Code level which range from 7 to 1. The values include 7 (CPU space), 6 (device space at the supervisor program level), 5 (device space at the supervisor data level), 4 (reserved), 3 (control space for the memory maps), 2 (device space at the user program level) and 1 (device space at the user data level). If no argument is given, the current setting is displayed.

The *t* command sets the trace program which allows the CPU to facilitate instruction-by-instruction tracing. This will allow a debugger program to monitor the execution of a program under test. The arguments include yes (y), no (n), and continuous (c) tracing.

The *u* command manipulates the piping of the display to the serial ports. This defines the current input (keyboard) and output (display screen) device. The following argument are legal:

```
> u   A or B        as input and output device
      A or B io     as input and output device
      A or B i      for input only
      A or B o      for output only
      k             select keyboard for input
      ki            select keyboard for input
      s             select screen for output
      so            select screen for output
      ks,sk         keyboard for input,screen for output
      A or B #      set speed of serial port(9600 baud,)
      e             echo input to output
      ne            do not echo input to output
      addr          set virtual serial port address

  EXAMPLE:       > u space A sets the SIO a port as the
                   console device.
```

FIGURE 4-25: USING THE U COMMAND

The *v* command displays the contents of addresses from the low address to the high address where the size is specified as *b* (byte), *w* (word), and *l* (long word) format. Entering a size character displays, and hitting the space bar terminates the display. An example would be to display the contents of virtual address space from the low address 0x1000 to the high address 0x2000, in word format as:

```
> v 1000 2000 w <cr>
```

FIGURE 4-26: USING THE *V* COMMAND

The *x* command will call up the extended diagnostic option menu to the display. This will allow the user to test other devices on the system.

The *z* command sets the breakpoint for the 68020. This command provides a means of inserting a breakpoint into a target code, which will give a clear announcement of when it reaches a breakpoint during the execution of an illegal instruction.

4.9 USING DISK DIAG AS A DIAGNOSTIC

1. Mount tape #1 of Standard Software Release
2. Obtain monitor prompt: >
3. Enter: btape (,,,) [Tape=st, xt ,mt] Example bst(,,,)
4. You should get BOOT:
5. Enter tape (,,3) after the word BOOT: Example BOOT: st(,,3)

6. Answer questions about your disk until you get diag>
7. Enter diag> h for help menu

You may do reads and seeks without destroying contents of disk but any format, or indiscriminate write operation will wipe out the operating system and data files forcing you to reinstall UNIX.

4.10 SYSDIAG

Sysdiag test is a high-level diagnostic designed to test Sun Hardware and runs under the UNIX operating system. It is installed in /usr partition during UNIX install as optional software "user level diagnostics"

-To run sysdiag

1. Logon as sysdiag
   (instructor will tell you password if necessary)
2. Select "Automatic" mode from Menu
   This will test all devices detected by the Kernel except tape
3. Errors detected by Sysdiag will be logged (we will see log when we exit)
4. The window on the right is called "devtop". If you are on a color machine and no color tests are being run in this window move mouse arrow to console window in center of screen and enter:
   #cd /dev
   #MAKEDEV cgtwo
   #MAKEDEV cgfour
   Move back to devtop , enter:
   Control C
   #devtop
   this should now run color tests.
5. To exit Sysdiag you must stop all windows then exit Suntools.
   -Enter Control C in each window
   -Hold right button down on mouse, select exit, release button.
6. You will now be automatically "MORE"ing the error log
7. Enter Control D to exit log.
8. On ASCII terminals type *endt* and follow the instructions.

Never exit Sysdiag by powering down or L1A. You may damage disk data.

This page left blank.

Diagnostic Executive

Objectives: Upon completion of this module, the student will be able to:

- Load, configure, run, and interpret the results of any Exec based diagnostic in 20 minutes or less. (Standalone, over Ethernet and over a Modem link.
- Store the Exec on disk from tape.
- Describe the Exec in terms of a functional block diagram and diagnostic philosophy.

Evaluation:

Completion of Lab Exercise X.

References:

- *PROM Users Guide* (Library Copy)
- *SunDiagnostics Executive Users Guide* (Library Copy)

The Sun Diagnostic Executive™

○ Provides its own Operating System

○ The Diagnostics run ‘under’ this OS

○ Shipped on tape media

○ Can be run standalone or remote

○ Multitasking

○ Can be used for verification, intermittent problems or peripherals

The Sun Diagnostic Executive™

The Sun Diagnostic Executive (Exec) is a UNIX based operating system designed to provide an enviroment to run system and FRU/component level diagnostics. The Exec replaces Standalone diagnostics.

This System is composed of two elements:

1. The Exec operating system.
2. The Diagnostics.

The Diagnostics run 'under' the Exec and are selected from menus.

The Exec (including diagnostics) is shipped on 1/4" and 1/2" tapes that can be booted standalone or loaded to the disk.

The Exec diagnostics can be run standalone, over Ethernet, or over standard telephone lines (multipath).

Several different diagnostics can be run simultaneously under the Exec (Multitasking) to load down and simulate failing conditions.

The Exec can be used for system verification, intermittent problems, peripheral verification and diagnosis.

Contents of Exec Tape*

NAME	FILE NUMBER	Description/Comments *Rev 1.1
Sun-2 Boot Block	0	Load before booting Exec on Sun-2
Sun-3 Boot Block	1	Load before booting Exec on Sun-3
Table of Contents	2	Contains list of contents on tape
Extract_Exec	3	Script to copy diagnostics to disk
Copyright	4	Textfile containing Copyright notice
exec	5	The Diagnostic Executive for Sun-2 and 3
diags	6	The Diagnostic Menu and File names
color2.exec	7	Sun-2 Color Diagnostic
color3.exec	8	Sun-3 Color Diagnostic
cpu.exec	9	Sun CPU diagnostic
eeptool.exec	A	EEPROM programming tool
ether.exec	B	Sun Ethernet Diagnostic
exectest.exec	C	Exec Verification Suite
fpa.exec	D	Sun-3 Floating Point Accelerator Tests
gp1.exec	E	Graphics Processor One Diagnostic
kb.exec	F	Sun Keyboard Diagnostic
mcp.exec	10	Sun ALM2/MCP Diagnostic
mem.exec	11	Sun Memory Diagnostic
mouse.exec	12	Sun Mouse Diagnostic
mti.exec	13	Sun MTI/ALM1 Diagnostic
scsisub.exec	14	Sun SCSI Diagnostic
sky2.exec	15	Sky FP Accelerator Diagnostic
smd.exec	16	Sun SMD(disk) Diagnostic
tape.exec	17	1/2 inch Tape Diagnostic
video.exec	18	Sun Video Diagnostic
vidmon.exec	19	Sun Video Monitor Diagnostic
vme3.exec	1A	Sun-3 VME Diagnostic
netcon	1B	Network Console Program
logfile	1C	Error log file
eccmem3.diag	1D	Standalone ECC memory Diagnostic
cache3.diag	1E	Standalone cache memory Diagnostic
tarfile	1F	Archive of tape and data for extract_diag
Copyright	20	Textfile containing Copyright notice

Contents of Exec Tape*

*Rev 1.1

Brief Description of each diagnostic

Booting from Tape Standalone

```
>b st( , , 1)
boot: st ( , , 5)
```

Booting from Tape
Standalone

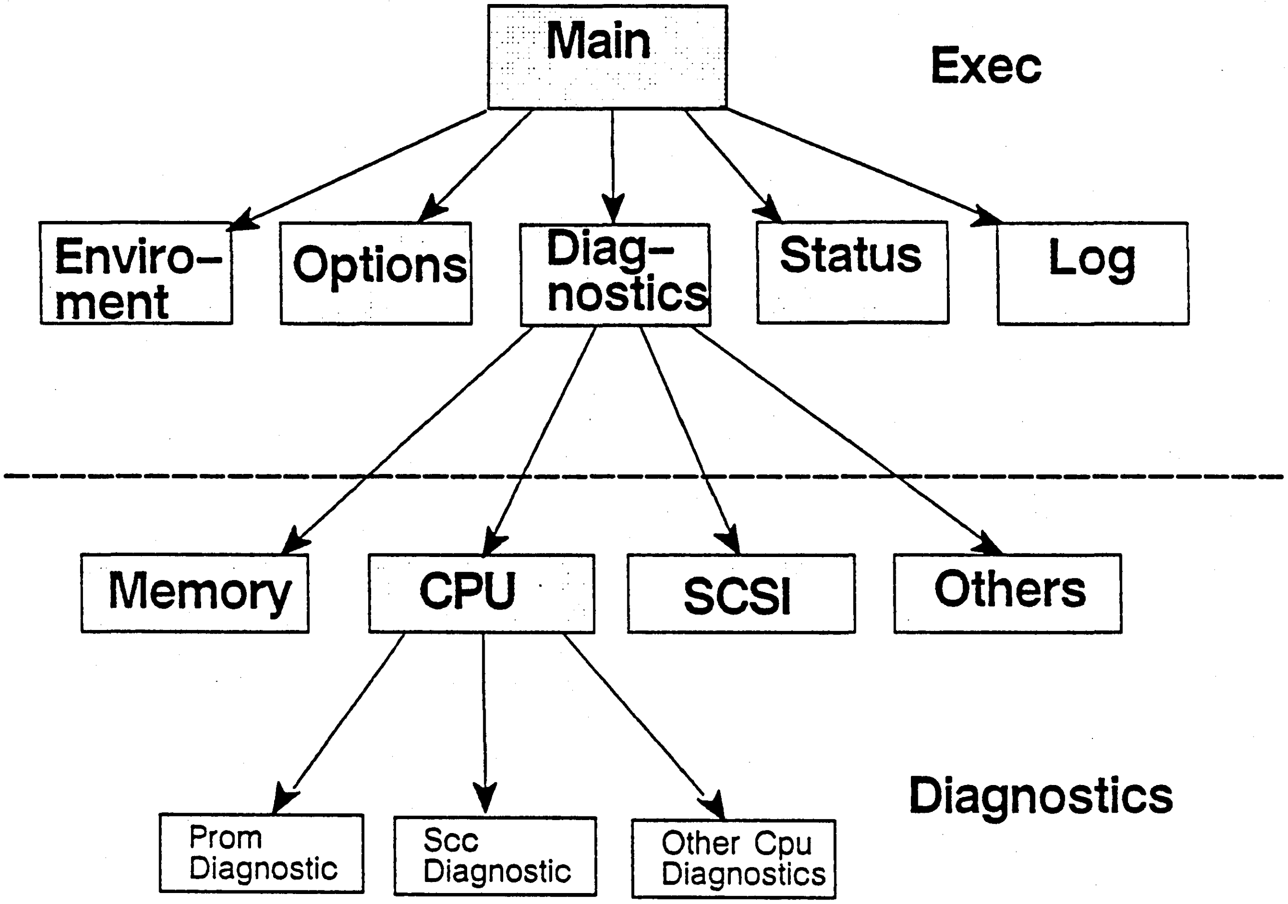
At the monitor prompt > enter: b *tape(, , N)*

tape=st, mt, or xt
N= 0 for Sun 2, 1 for Sun 3

When you get a prompt boot: enter *tape*(, , 5)

Notice from *contents of tape* page, that file 5 = The Diagnostic Executive

Exec Menu Hierarchy



Exec Menu Hierarchy

Modeled after the UNIX File system

To run any given diagnostic, make selections from menus and follow down the path.

Enter <*esc*> to move 'back' one menu.

Enter *Control C* to exit any diagnostic and return to main menu.

Enter *!* to see last five commands.

Enter @ to put test in the background.

Main Menu

```
Diagnostic Executive   Rev:1.x    25 Sept 1987   Main Menu


Enviroment             Set Executive Enviroment

Options                Set global diagnostic options

Diagnostics            Available Diagnostics

Status                 Display task status

Log                    Display Error log

Script=                Source a script file

Boot                   Exit and boot another program


Command==>
```

Main Menu

Select from menu by entering the selection on the command line.
(The abbreviation you can use for a selection is the Upper Case leters in the menu)

Example:

Select diagnostics menu by entering *d* on the command line.

Diagnostics Menu

```
Diagnostic Executive ..............................Diagnostics Menu

C2          Sun–2 Color Board Diagnostic
C3          Sun–3 Color Board Diagnostic
CPu         Sun–3 CPU Board Diagnostic
EEprom      EEPROM Editing Tool
Ether       Ethernet Diagnostic
Fpa         Sun–3 FPA Diagnostic
Gp          Graphics Processor and Buffer Diagnostics
Kb          Keyboard Diagnostic
MCp         MCP and ALM2 Diagnostics
Mem         Memory Diagnostic
MOuse       Mouse Diagnostic
MTi         MTI and ALM Diagnostic
SUBsystem   SCSI Subsystem Diagnostic
SMd         SMD Subsystem Diagnostic
Tape        1/2 " Tape Diagnostic
Video       Video Sun–3 Video Diagnostic
VIDMon      Video Monitor Diagnostic
VME         Sun–3 VME Interface Diagnostic

Command==>
```

Diagnostics Menu

Select from this menu on the command line.

Example:

Select SMD subsystem diagnostics by entering *sm* on the command line.

Sun SMD Diagnostic Main Menu

```
SMD Subsystem Exerciser    REV 1    Date    Main Menu


Controller            Controller tests Menu

Drive                 Drive test Menu

All                   All test sequence

Burnin                Burn-in Controller tests

Quick                 Quick test sequence

PARameters            Parameter Display

?               Display the command syntax of this menu


Command==>
```

Sun SMD Diagnostic Main Menu

You may execute diagnostics from this menu (*q*) or go down to subsequent menus (*d*).

To execute SMD diagnostics you must change the Controller and Drive parameters on any command line.

Once you have changed them, they remain set as long as you remain in the SMD diagnostics.

You can check the paramters by entering *par*.

Example:

To run a Quick sequence of tests to the first Fuji Eagle 2361 on the first 451 controller, enter the following on the command line:

```
Command==> q ct0=xy451 dt0=fuj2361
```

A complete list of diagnostic variables and parameters is located at the end of the SMD chapter.

Putting Bootable Exec on the Disk

- Install a bootblock in the /usr partition:

```
cd /usr/mdec
installboot bootsd /dev/rsd0g (standalone)
        or
installboot bootxy /dev/rxy0f   (server)
cp /boot /usr
```

- Change directories to where the diagnostics will be written

```
cd /usr/stand
```

- Mount tape in drive and position as follows:

```
mt –f /dev/nrst0 rew            (nrmt0 for 1/2")
mt –f /dev/nrst0 fsf 3
```

- Pull extract_exec program off the the tape

```
tar xvf /dev/nrst0
mt –f /dev/nrst0 rew
```

- Use extract_exec to install the diagnostics on disk:

```
extract_exec st0               (mt0 for 1/2")
```

- Executing Exec from Disk:

```
b sd(0,0,6)stand/exec          (0,0,7 for server)
```

Putting Bootable Exec on the Disk

This page left blank.

CHAPTER FIVE

USING THE SUN 3 SYSTEM EEPROM

OVERVIEW

Sun 3 architecture has included a user configurable EEPROM for setting system parameters. This section of the training manual will describe what the EEPROM is used for, and how the user can interact with system set-up parameters located on the device.

Some of the uses of the EEPROM can include booting from a predefined device, setting up a custom boot banner, setting up how much memory will be tested during self test, and defining which output device will be used for the display. The total layout of the EEPROM is 2 K Bytes.

5.1 PROGRAMMING THE EEPROM

To use the EEPROM (at its present state in Sun 3 products) the user must enter the MONITOR level by depressing the L1-A key, or by halting the system using the appropriate command (typically the halt or sync command is used).

While in the MONITOR level, the user should enter the Q command to invoke the address of the EEPROM address space as follows:

```
> q space argument <cr>
```

FIGURE 5-1: CALLING UP THE EEPROM

The EEPROM repeatedly uses the values 00 and 12 as flags. A 00 is taken as a signal to use default settings, where a 12 tells the system to look for more information.

WARNING: The EEPROM has a 10 msec minimum delay factor which must be met for back-to-back writes. The EEPROM also has a maximum of 10,000 writes to a single location.

The following address locations can be modified by the user:

LOCATION	AREA DEFINITION	PAGE
0x000014	MBytes of installed memory	5-3
0x000015	MBytes to selftest (diag switch=normal)	5-3
0x000016	display screen size	5-4
0x000017	set watchdog reset resolution	5-4
0x000018	setting default booting (diag switch=normal)	5-5
0x000019	Alternate boot path (diag switch=normal)	5-5
0x00001F	set the primary terminal	5-6
0x000020	set display logo(data start - 0x68)	5-7
0x000021	Keyboard click on/off	5-7
0x000022	Diag boot device (diag switch=diag)	5-7
0x000050	High resolution columns	5-9
0x000051	High resolution rows	5-9
0x000058	SIO A baud rate	5-9
0x000059	Alternate baud rate	5-10
0x00005B	SIO A DTR/RTS set/unset	5-10
0x000060	SIO B baud rate	5-11
0x000061	Alternate baud rate	5-11
0x000063	SIO B DTR/RTS set/unset	5-12
0x000068	Custom banner	5-12

FIGURE 5-2: EEPROM ADDRESS SPACE LOCATIONS

5.2 CHANGING THE EEPROM VALUES

To set the number of MBytes of installed memory on the system address location 0x14 is used. This byte will contain the total number (in hex) or MBytes of memory installed in the system. To call up the total number of installed memory in MBytes the user enters the Q command, and views the following display while at the Monitor level:

```
> q 14 <cr>
EEPROM 014:04
```

FIGURE 5-3: CHANGING THE MBYTES OF MEMORY FOUND

This notifies the user that 4 M Byte of memory has been found. To set up how much of the available the user wants to be checked by the system self test, the user enters the test address space of the EEPROM as follows:

```
> q 14 <cr>
EEPROM 014:04 <cr> or space <cr>
EEPROM 015:04(displays when <cr> is invoked)
```

FIGURE 5-4: CHANGING THE MBYTES OF MEMORY TO TEST

By entering a carriage return after the EEPROM 014:04, the value of the next location EEPROM 015:04 is called up to the display. This byte at location 0x15 will contain the total number (in hex) of M Bytes of memory that the firmware will test before attempting to boot the OS. Note that this number is read while the NORM/DIAG switch is in the NORM position, but, will be ignored while in the DIAG position. All memory will be tested while in DIAG mode, regardless of the EEPROM value at 0x15.

To change the value of the MBytes to be tested under self test, enter the new value in MBytes, and enter a space and return to the MONITOR level as illustrated below:

```
> q 15 <cr>
EEPROM 015:04(indicates 4 MB will be tested under NORM)
```

FIGURE 5-5: CHANGING THE MB OF MEMORY TO TEST ON BOOT

The next address location, 0x16, selects the appropriate video display screen size for the color or monochromatic monitor. The illustrations below indicate the screen size values:

```
> q 16 <cr>
EEPROM 016:000?12 space <cr>
```

FIGURE 5-6: CHANGING THE SCREEN SIZE

Here the value of 0x12 is entered specifying that 0x12 equaled the 1K screen at location 016, and the default value was 0x00 for the 1152 by 900 screen. Use the next Figure to input the appropriate screen size value:

SCREEN SIZE VALUE	VALUE DEFINITION	
0x00	1152 by 900 screen	STD Resolution
0x12	1024 by 1024 screen	OPTIONAL
0x13	1600 by 1280 screen	High Resolution
0x14	1440 by 1440 screen	OPTIONAL

FIGURE 5-7: SELECTION FOR THE MONITOR SCREEN SIZE

The next address location, 0x17, is a byte which selects the appropriate action for the firmware after a 'watchdog reset' (double bus fault) results. See Figure 5-8 below:

```
> q 17 <cr>
EEPROM 017:00?12 space <cr>
```

WHERE:	017:12 = the resultant action of the watchdog reset will be to perform a POR and boot UNIX
	017:00 = the watchdog reset will force the system into the monitor level

FIGURE 5-8: CHANGING THE WATCHDOG RESET ACTION

To define the boot path, location 0x18 will contain a byte value for loading the OS. If this value is 00, the bootstrap program polls the devices by priority. By default, the bootstrap will poll for the xy disk, then the sd disk, and then will default to booting over the Ethernet. The system will also accept boot polling from the tape drives including st (SCSI 1/4-inch tape), mt (TapeMaster 1/2-inch tape), and xt (Xylogics 1/2-inch tape). The user can invoke the 'KB' command at the Monitor level to see the default. To change the default boot path the user should enter and define the polling as follows:

```
> q 18 <cr>
EEPROM 018:00? <cr>
```

FIGURE 5-9: POLLING THE DEFAULT BOOT DEVICE

In the example above, the 00 forces the firmware to boot the OS over the default path. If a '12' option is used, the option specified by the EEPROM will be used starting at address location 0x19 (Figure 5-10 and 5-11):

OS PATH	PATH DEFINITION
0x00	Firmware polls default devices(xy, sd, ie)
0x12	Uses specified device path indicated by EEPROM

FIGURE 5-10: SELECTING THE OS BOOT PATH

The following five bytes allow the user to specify a boot path string instead of the default path:

EEPROM ADDRESS	PATH DEFINITION
0x19	Boot device, 1st character
0x1A	Boot device, 2nd character
0x1B	Controller number in hex
0x1C	Unit number in hex
0x1D	Partition number in hex

FIGURE 5-11: SELECTED OS BOOT PATH

The following figure converts the selected boot device value from ASCII to hex:

SELECTED BOOT DEVICE	ADDRESS 0x19	ADDRESS 0x1A
xy=Xylogics 450/451 disk	0x78	0x79
sd=SCSI disk	0x73	0x64
ie=Intel Ethernet	0x69	0x65
le=AMD Ethernet	0x6C	0x65
st=SCSI 1/4" tape	0x73	0x74
xt=Xylogics 1/2" tape	0x78	0x74
mt=Tapemaster 1/2" tape	0x6D	0x74

FIGURE 5-12: BOOT DEVICE SPECIFICATION VALUE

To provide a primary terminal display pathway when more that one display is available, byte location 0x1F of the EEPROM address space can be used to manipulate the output device as follows:

> q 1F <cr> EEPROM 01F:00? space <cr>	
WHERE:	10 = SIO A SERVERS
	11 = SIO B
	12 = color monitor
	00 = monochrome monitor

FIGURE 5-13: THE PRIMARY TERMINAL VALUE

The next byte on the EEPROM is 0x20, which is used for selecting the Sun logo (display) or displaying a custom banner when booting the system. By default the Sun logo will be displayed during POR. A custom banner can be defined and displayed in an 80-character buffer located on the EEPROM from 0x68 to 0xB8.

> q 20 <cr> EEPROM 020:00? space <cr>		
	DISPLAY LOGO	VALUE DEFINITION
WHERE:	0x00	Display the default Sun logo
	0x12	Display the custom banner at 0x68

FIGURE 5-14: LOGO DISPLAY SELECTION

The next byte defines if the Sun 3 keyboard will be initialized with its keyclick option ON or OFF, as follows:

> q 21 EEPROM 021:00? space <cr>		
	CLICK VALUE	VALUE DEFINITION
WHERE:	0x00	Turns the keyclick OFF by default
	0x12	Turns the keyclick ON

FIGURE 5-15: KEYCLICK SELECTION

During the diagnostic analysis of a system, that is the user switch is set to the DIAG position, a number of diagnostic boot paths can be defined by the EEPROM. These next five bytes of the EEPROM define the path as follows:

EEPROM ADDRESS	PATH DEFINITION
0x22	Boot device, 1st character
0x23	Boot device, 2nd character
0x24	Controller number in hex
0x25	Unit number in hex
0x26	Partition number in hex

FIGURE 5-16: DIAGNOSTIC PATH SELECTION

SELECTED BOOT DEVICE	ADDRESS 0x22	ADDRESS 0x23
xy=Xylogics 450/451 disk	0x78	0x79
sd=SCSI disk	0x73	0x64
ie=Intel Ethernet	0x69	0x65
le=AMD Ethernet	0x6C	0x65
st=SCSI 1/4" tape	0x73	0x74
xt=Xylogics 1/2" tape	0x78	0x74
mt=Tapemaster 1/2" tape	0x6D	0x74

FIGURE 5-17: SELECTED DEVICE VALUES

In addition to selecting the diagnostic boot device, the user can also set the diagnostic boot path starting at 0x28. The next 40 bytes of the EEPROM represent those 40 byte locations from 0x28 to 0x4F. For example if the diagnostic boot path was 'sd( ) stand/diag', the user could input that string starting at 0x28. Note that the 40th byte is set to 0x00 by default to terminate the string as follows:

> q 28 <cr> EEPROM 028:00? <cr>						
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	0x00	----	----

FIGURE 5-18: SELECTION OF DIAGNOSTIC BOOT PATH

With the introduction of the 3/200 series of Sun workstations, the user can now enjoy the selection of a high resolution display at 1600 by 1280. The high resolution screen size can be configured by the EEPROM. These next two bytes on the EEPROM allow the selection of the number of columns and number of rows for the high resolution map at address location 0x50 and 0x51, as follows:

```
> q 50 <cr>
EEPROM 050:50? space <cr>
EEPROM 051:30? <cr>
```

FIGURE 5-19: COLUMNS/ROWS FOR THE HIGH RESOLUTION DISPLAY

Note that by default, as indicated in Figures 5-19 and 5-20, the default is 80 columns by 32 rows. To get a full screen set 50=88 ,51=88.

The next byte on the EEPROM selects if the SIO A port will be at 9600 baud (by default), or at a rate specified by the user at address location 0x58. The baud rates for SIO A are defined at address location 0x59 to 0x5A, as follows in the next two Figures. Note that SIO A, by factory default, is set to 9600 baud.

```
> q  58 <cr>
EEPROM 058:00? space <cr>
```

WHERE:	RATE DEFAULT	SETTING DEFINITION
	0x00	Set to 9600 by default
	0x12	Set by the user starting at 0x59

FIGURE 5-20: BAUD RATE DEFAULT FOR SIO A

The next two bytes are used to define, or rather initialize, the SIO A port at the user specified rate when 0x58 has been set to 0x12. These two bytes will be the hex equivalent of the desired baud rate, and are indicated by the following Figure. Note that 0x59 contains the high byte, and 0x5A contains the low byte.

BAUD RATE	HEX EQUIVALENT	LOCATION 0x59	LOCATION 0x5A
300	0x012C	0x01	0x2C
600	0x0258	0x02	0x58
1200	0x04B0	0x04	0xB0
2400	0x0960	0x09	0x60
4800	0x12C0	0x12	0xC0
9600	0x2580	0x25	0x80
19200	0x4B00	0x4B	0x00
38400	0x9600	0x96	0x00

FIGURE 5-21: SIO A PORT BAUD RATE SELECTION

The next byte selects if the SIO A port will have the DTR and RTS signals asserted during initialization. This function can be written to the EEPROM at address location 0x5B, as follows:

```
> q 5B
EEPROM 05B:00? space <cr>
```

	VALUE	VALUE DEFINITION
WHERE:	0x00	Asserts both DTR and RTS
	0x12	Does not assert DTR and RTS

FIGURE 5-22: SIO A PORT DTR/RTS OPTION

The values listed above are also available for the SIO B port. The next four bytes represent the default and selected values which can be specified by EEPROM address locations 0x60, 0x61 to 0x62, and 0x63:

```
> q  60 <cr>
EEPROM 060:12? <cr>
```

	RATE DEFAULT	SETTING DEFINITION
WHERE:	0x00	Set to 9600 by default
	0x12	Set by the user starting at 0x61

FIGURE 5-23: BAUD RATE DEFAULT FOR SIO B

The next two bytes are used to define, or rather initialize, the SIO B port at the user specified rate when 0x60 has been set to 0x12. Note that at the time of the release of this manual the SIO B port is set to 1200 baud, by default.

These two bytes will be the hex equivalent of the desired baud rate, and are indicated by the following Figure. Note that 0x61 contains the high byte, and 0x62 contains the low byte.

BAUD RATE	HEX EQUIVALENT	LOCATION 0x61	LOCATION 0x62
300	0x012C	0x01	0x2C
600	0x0258	0x02	0x58
1200	0x04B0	0x04	0xB0
2400	0x0960	0x09	0x60
4800	0x12C0	0x12	0xC0
9600	0x2580	0x25	0x80
19200	0x4B00	0x4B	0x00
38400	0x9600	0x96	0x00

FIGURE 5-24: SIO B PORT BAUD RATE SELECTION

The next byte selects if the SIO B port will have the DTR and RTS signals asserted during initialization. This function can be written to the EEPROM at address location 0x63, as follows:

```
> q 63 <cr>
EEPROM 063:00? space <cr>
```

	VALUE	VALUE DEFINITION
WHERE:	0x00	Asserts both DTR and RTS
	0x12	Does not assert DTR and RTS

FIGURE 5-25: SIO B PORT DTR/RTS OPTION

The last section of the EEPROM which can be set by the user (at the time of release of this manual) is address locations 0x68 through 0xB7. These 80 bytes represent a character buffer which can be set by the user to display a custom banner instead of the Sun logo, when 0x20 is set to 0x12. Note that these locations are written to with the ASCII equivalent, and are terminated with 0x00 at location 0xB7, as illustrated below:

```
> q 68 <cr>
EEPROM 068:00? <cr>
```

ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
ASCII	ASCII	0x00				

FIGURE 5-26: SELECTING A CUSTOM BANNER

Once again, it is important to note that the EEPROM has a minimum 10 msec delay factor for back-to-back writes, and that there is a maximum of 10,000 writes per address location. The user should not attempt to write to an address location which has not been covered by this manual.

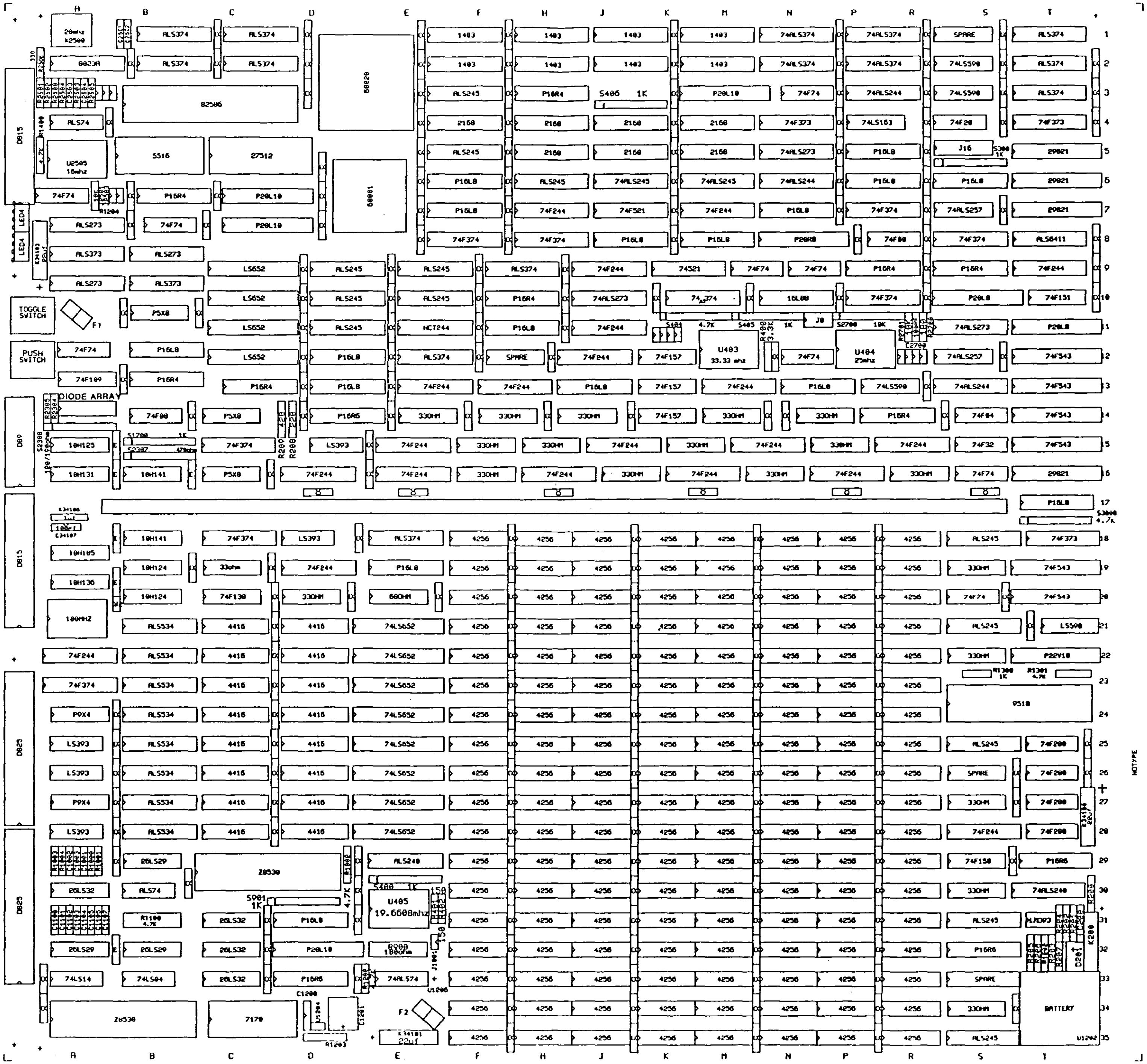
ASCII to HEX/HEX to ASCII CONVERSION CHART

ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX
nl (line feed)	0A	6	36	N	4E	f	66
cr (return)	0D	7	37	O	4F	g	67
sp (space)	20	8	38	P	50	h	68
!	21	9	39	Q	51	i	69
"	22	:	3A	R	52	j	6A
#	23	;	3B	S	53	k	6B
$	24	<	3C	T	54	l	6C
%	25	=	3D	U	55	m	6D
&	26	>	3E	V	56	n	6E
'	27	?	3F	W	57	o	6F
(	28	@	40	X	58	p	70
)	29	A	41	Y	59	q	71
*	2A	B	42	Z	5A	r	72
+	2B	C	43	[	5B	s	73
,	2C	D	44	\	5C	t	74
–	2D	E	45	]	5D	u	75
.	2E	F	46	^	5E	v	76
/	2F	G	47	_	5F	w	77
0	30	H	48	‘	60	x	78
1	31	I	49	a	61	y	79
2	32	J	4A	b	62	z	7A
3	33	K	4B	c	63	{	7B
4	34	L	4C	d	64	|	7C
5	35	M	4D	e	65	}	7D
						~	7E
						del	7F

This page left blank.

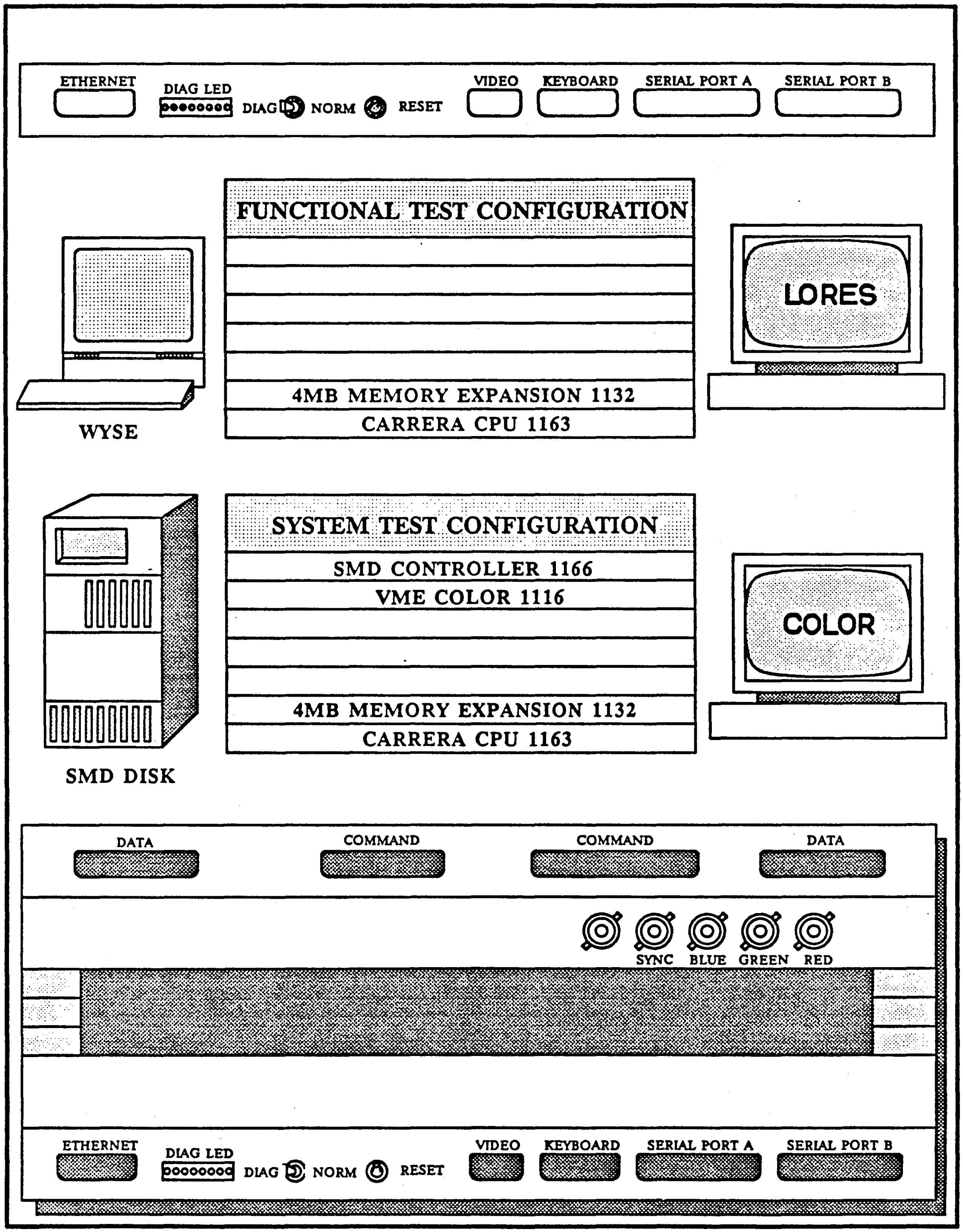
CARRERA CPU 270-1118-03 REV 50

COMPONENT SIDE LEGEND



A B C D E F H J K M N P R S T
20mhz x2500
ALS374 ALS374 1403 1403 1403 1403 74ALS374 74ALS374 SPARE ALS374 1
8023A ALS374 ALS374 1403 1403 1403 1403 74ALS374 74ALS374 74LS590 ALS374 2
68020
82586 ALS245 P16R4 S406 1K P20L10 74F74 74ALS244 74LS590 ALS374 3
ALS74 2168 2168 2168 2168 74F373 74LS163 74F20 74F373 4
DB15
5516 27512 ALS245 2168 2168 2168 74ALS273 P16L8 J16 S300 1K 29821 5
U2505 16mhz
P16L8 ALS245 74ALS245 74ALS245 74ALS244 P16L8 P16L8 29821 6
74F74 P16R4 P20L10 68881
R1204
P16L8 74F244 74F521 74F244 P16L8 74F374 74ALS257 29821 7
LED4
ALS273 74F74 P20L10 74F374 74F374 P16L8 P16L8 P20R8 74F00 74F374 ALS6411 8
LED4
ALS373 ALS273
ALS273 ALS373 LS652 ALS245 ALS245 ALS374 74F244 74S21 74F74 74F74 P16R4 P16R4 74F244 9
TOGGLE SWITCH
F1
P5X8 LS652 ALS245 ALS245 P16R4 74ALS273 74F374 16L88 74F374 P20L8 74F151 10
LS652 ALS245 HCT244 P16L8 74F244 S404 4.7K S405 R400 3.3K 1K J8 S2700 10K 74ALS273 P20L8 11
PUSH SWITCH
74F74 P16L8 LS652 P16L8 ALS374 SPARE 74F244 74F157 U403 33.33 mhz 74F74 U404 25mhz C2700 74ALS257 74F543 12
74F109 P16R4 P16R4 P16L8 74F244 74F244 P16L8 74F157 74F244 P16L8 74LS590 74ALS244 74F543 13
DIODE ARRAY
74F00 P5X8 R200 220 P16R6 330HM 330HM 330HM 74F157 330HM 330HM P16R4 74F04 74F543 14
DB9
S1700 1K S2307 47ohm
10H125 74F374 R209 R208 LS393 74F244 330HM 330HM 74F244 330HM 74F244 330HM 74F244 74F32 74F543 15
10H131 10H141 P5X8 74F244 74F244 330HM 74F244 330HM 74F244 330HM 74F244 330HM 74F74 29821 16
P16L8 17
S3000 4.7K
10H141 74F374 LS393 ALS374 4256 4256 4256 4256 4256 4256 4256 4256 ALS245 74F373 18
10H105
DB15
10H124 33ohm 74F244 P16L8 4256 4256 4256 4256 4256 4256 4256 4256 330HM 74F543 19
10H136 10H124 74F138 330HM 680HM 4256 4256 4256 4256 4256 4256 4256 4256 74F74 74F543 20
100MHZ
ALS534 4416 4416 74LS652 4256 4256 4256 4256 4256 4256 4256 4256 ALS245 LS590 21
74F244 ALS534 4416 4416 74LS652 4256 4256 4256 4256 4256 4256 4256 4256 330HM P22V10 22
R1300 1K R1301 4.7K
74F374 ALS534 4416 4416 74LS652 4256 4256 4256 4256 4256 4256 4256 4256 23
9518
P9X4 ALS534 4416 4416 74LS652 4256 4256 4256 4256 4256 4256 4256 4256 24
DB25
LS393 ALS534 4416 4416 74LS652 4256 4256 4256 4256 4256 4256 4256 4256 ALS245 74F280 25
LS393 ALS534 4416 4416 74LS652 4256 4256 4256 4256 4256 4256 4256 4256 SPARE 74F280 26
P9X4 ALS534 4416 4416 74LS652 4256 4256 4256 4256 4256 4256 4256 4256 330HM 74F280 27
LS393 ALS534 4416 4416 74LS652 4256 4256 4256 4256 4256 4256 4256 4256 74F244 74F280 28
26LS29 Z8530 ALS240 4256 4256 4256 4256 4256 4256 4256 4256 74F158 P16R6 29
26LS32 ALS74 S400 1K 4.7K 150 4256 4256 4256 4256 4256 4256 4256 4256 330HM 74ALS240 30
DB25
R1100 4.7K 26LS32 S901 1K P16L8 U405 19.6608mhz 4256 4256 4256 4256 4256 4256 4256 4256 ALS245 31
26LS29 26LS29 26LS32 P20L10 R900 180ohm 150 4256 4256 4256 4256 4256 4256 4256 4256 P16R6 D201 32
74LS14 74LS04 26LS32 P16R6 74ALS74 J1001 U1200 4256 4256 4256 4256 4256 4256 4256 4256 SPARE 33
C1200 C1201 F2 K200
Z8530 7170 4256 4256 4256 4256 4256 4256 4256 4256 330HM BATTERY 34
R1203 K34101 22uf 4256 4256 4256 4256 4256 4256 4256 4256 ALS245 U1202 35
A B C D E F H J K M N P R S T
NOTYPE

Depot Test Procedure



FUNCTIONAL TEST CONFIGURATION

4MB MEMORY EXPANSION 1132

CARRERA CPU 1163

SYSTEM TEST CONFIGURATION

SMD CONTROLLER 1166

VME COLOR 1116

4MB MEMORY EXPANSION 1132

CARRERA CPU 1163

Test Procedure 501-1163

Carrera CPU Board 2MB

REFERENCE DOCUMENTS	
SUN-3 CPU HARDWARE ENGINEERING MANUAL	800-1163
SUNDIAGNOSTIC EXECUTIVE USER'S GUIDE	800-2326
SYSTEM DIAGNOSTIC USER'S GUIDE	800-2517
TIME OF DAY CLOCK CALIBRATION PROCEDURE	PRC-0100

EQUIPMENT REQUIREMENTS	
7/12 SLOT TEST STATION WITH 850W POWER SUPPLY	
FILESERVER ETHERNET DROP	
WYSE TERMINAL WITH RS-232 CABLE	WYSE-30
SUN KEYBOARD AND MOUSE	370-1063 / 370-1058
LO-RES MONITOR AND CABLE	540-1162 / 530-1133
SERIAL PORT LOOPBACK CABLE RS-232 A<->B	530-1205
COLOR MONITOR AND CABLE	565-1000 / 530-1307
SMD CONTROLLER BOARD	501-1166
SMD DISK SUBSYSTEM	
VME COLOR BOARD	501-1116
4MB MEMORY EXPANSION	501-1132
CARRERA ID PROM	

SOFTWARE REQUIREMENTS
DIAGNOSTIC EXECUTIVE WITH CPU, MEM, ETHER, VIDEO.EXEC FILES
SYSTEM DIAGNOSTIC (SYSDIAG) VERSION 2.60 OR LATER
VMUNIX OPERATING SYSTEM RELEASE 3.2 OR LATER

JUMPER INFORMATION

LOC	LABEL	PINS	SHUNT
S5	J300	ALL	IN
N11	J400	1-2	IN
		3-4	OUT
		5-6	OUT
		7-8	IN
E32	J1001	1-2	IN
A3	J1201	1-2	IN
	J1200	1-2	OUT
	J2502	1-2	IN
B20	J2301	1-2	IN
A7	J2503	1-2	OUT
	J2501	1-2	IN
R12	J2700	1-2	OUT
	J2701	1-2	IN
	J2702	1-2	OUT
	J2703	1-2	IN
K12	J2505	1-2	OUT
	J100	1-2	OUT
	J3102	1-2	OUT
	J3101	1-2	IN

INSTALL THE CPU BOARD UNDER TEST IN THE TEST STATION AS INDICATED.

7/12 SLOT TEST STATION

SLOT	BG3	IACK	BOARD TYPE
12			
11			
10			
9			
8			
7	OUT	OUT	SMD CONTROLLER 501-1166
6	IN	OUT	VME COLOR 501-1116
5			
4			
3			
2	IN	IN	4MB MEM EXP 501-1132
1	OUT	OUT	CARRERA CPU 501-1163

Rev. 11/88

sun microsystems

ACCEPTANCE CRITERIA:

SUCCESSFULLY EXECUTE SELFTEST AND CLEAR THE EEPROM

SUCCESSFULLY EXECUTE THE DEFAULT TEST SUITE FOR ONE ERROR FREE PASS OF EACH OF CPU.EXEC, MEM.EXEC, ETHER.EXEC, AND VIDEO.EXEC

COMPLETE TIME OF DAY CLOCK CALIBRATION

TEST THE CARRERA CPU WITH A COLOR BOARD AND SMD DISK SUBSYSTEM FOR 1.0 HOUR WITHOUT ERROR RUNNING SYSDIAG IN AUTOMATIC MODE

TEST PROCEDURE

[1] PERFORM PHYSICAL INSPECTION

[2] INSTALL TEST ID PROM AT LOCATION B10

[3] CONFIGURE THE BOARD AND INSTALL IN THE TEST STATION

[4] CONNECT CABLES: WYSE TERMINAL TO SERIAL PORT A, ETHERNET, KEYBOARD, VIDEO TO LORES MONITOR

[5] SET DIAG/NORM TO DIAG AND EXECUTE SELFTEST

[6] AT THE MONITOR PROMPT > ENTER q* TO CLEAR EEPROM

[7] CONFIGURE THE EEPROM AS FOLLOWS;

LOC	VAL	LOC	VAL
14	10	B8	AA
15	10	B9	55
16	00	BC	01
		BD	04
1F	12	BE	01

[8] TYPE k2 TO CHECK SOFTWARE RESET

[9] WHEN SELFTEST STARTS, PRESS THE RESET SWITCH AND OBSERVE WATCHDOG RESET MESSAGE

Rev. 11/88

TEST PROCEDURE CONTINUED

[10] LOAD THE DIAGNOSTIC EXECUTIVE

[11] EXECUTE THE DEFAULT TEST SUITE FOR ONE ERROR FREE PASS FOR EACH OF THE FOLLOWING DIAGNOSTICS:

CPU DIAGNOSTIC
MEMORY DIAGNOSTIC
ETHERNET DIAGNOSTIC
VIDEO DIAGNOSTIC

[12] PERFORM TIME OF DAY CLOCK CALIBRATION

[13] POWER OFF AND INSTALL COLOR BOARD AND SMD CONTROLLER. ATTACH CABLES APPROPRIATELY.

[14] INSTALL SERIAL PORT LOOPBACK CABLE A<->B

[15] SET DIAG/NORM SWITCH TO NORM AND POWER ON

[16] BOOT UNIX FROM SMD DISK SUBSYSTEM

[17] CLEAR THE ERROR LOG

[18] START-UP SYSTEM DIAGNOSTIC

[19] RUN SYSDIAG FOR ONE HOUR - WITHOUT ERROR

[20] CHECK ERROR LOG FOR ABSENCE OF ERRORS

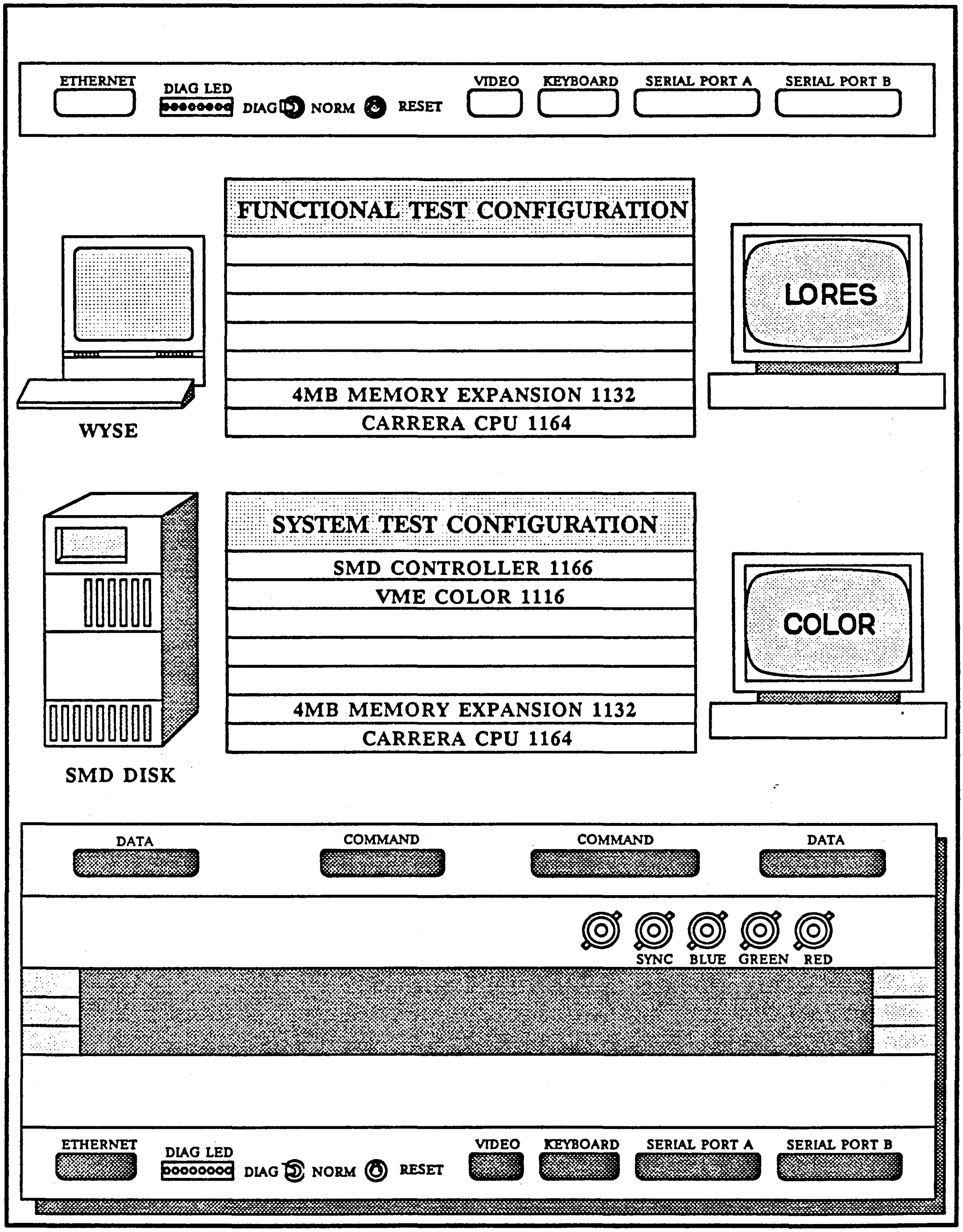
[21] HALT UNIX AND POWER OFF

TEST COMPLETE

Test Procedure

501–1164

Carrera CPU Board



FUNCTIONAL TEST CONFIGURATION

4MB MEMORY EXPANSION 1132

CARRERA CPU 1164

SYSTEM TEST CONFIGURATION

SMD CONTROLLER 1166

VME COLOR 1116

4MB MEMORY EXPANSION 1132

CARRERA CPU 1164

Rev. 11/88

Test Procedure 501-1164

Carrera CPU Board

REFERENCE DOCUMENTS	
SUN-3 CPU HARDWARE ENGINEERING MANUAL	800-1163
SUNDIAGNOSTIC EXECUTIVE USER'S GUIDE	800-2326
SYSTEM DIAGNOSTIC USER'S GUIDE	800-2517
TIME OF DAY CLOCK CALIBRATION PROCEDURE	PRC-0100

EQUIPMENT REQUIREMENTS	
7/12 SLOT TEST STATION WITH 850W POWER SUPPLY	
FILESERVER ETHERNET DROP	
WYSE TERMINAL WITH RS-232 CABLE	WYSE-30
SUN KEYBOARD AND MOUSE	370-1063 / 370-1058
LO-RES MONITOR AND CABLE	540-1162 / 530-1133
SERIAL PORT LOOPBACK CABLE RS-232 A<->B	530-1205
COLOR MONITOR AND CABLE	565-1000 / 530-1307
SMD CONTROLLER BOARD	501-1166
SMD DISK SUBSYSTEM	
VME COLOR BOARD	501-1116
4MB MEMORY EXPANSION	501-1132
CARRERA ID PROM	

SOFTWARE REQUIREMENTS
DIAGNOSTIC EXECUTIVE WITH CPU, MEM, ETHER, VIDEO.EXEC FILES
SYSTEM DIAGNOSTIC (SYSDIAG) VERSION 2.60 OR LATER
VMUNIX OPERATING SYSTEM RELEASE 3.2 OR LATER

JUMPER INFORMATION

LOC	LABEL	PINS	SHUNT
S5	J300	ALL	IN
N11	J400	1-2	IN
		3-4	OUT
		5-6	OUT
		7-8	IN
E32	J1001	1-2	IN
A3	J1201	1-2	IN
	J1200	1-2	OUT
	J2502	1-2	IN
B20	J2301	1-2	IN
A7	J2503	1-2	OUT
	J2501	1-2	IN
R12	J2700	1-2	OUT
	J2701	1-2	IN
	J2702	1-2	OUT
	J2703	1-2	IN
K12	J2505	1-2	OUT
	J100	1-2	OUT
	J3102	1-2	IN
	J3101	1-2	IN

INSTALL THE CPU BOARD UNDER TEST IN THE TEST STATION AS INDICATED.

7/12 SLOT TEST STATION

SLOT	BG3	IACK	BOARD TYPE
12			
11			
10			
9			
8			
7	OUT	OUT	SMD CONTROLLER 501-1166
6	IN	OUT	VME COLOR 501-1116
5			
4			
3			
2	IN	IN	4MB MEM EXP 501-1132
1	OUT	OUT	CARRERA CPU 501-1164

Rev. 11/88

ACCEPTANCE CRITERIA:

SUCCESSFULLY EXECUTE SELFTEST AND CLEAR THE EEPROM

SUCCESSFULLY EXECUTE THE DEFAULT TEST SUITE FOR ONE ERROR FREE PASS OF EACH OF CPU.EXEC, MEM.EXEC, ETHER.EXEC, AND VIDEO.EXEC

COMPLETE TIME OF DAY CLOCK CALIBRATION

TEST THE CARRERA CPU WITH A COLOR BOARD AND SMD DISK SUBSYSTEM FOR 1.0 HOUR WITHOUT ERROR RUNNING SYSDIAG IN AUTOMATIC MODE

TEST PROCEDURE

[1] PERFORM PHYSICAL INSPECTION

[2] INSTALL TEST ID PROM AT LOCATION B10

[3] CONFIGURE THE BOARD AND INSTALL IN THE TEST STATION

[4] CONNECT CABLES: WYSE TERMINAL TO SERIAL PORT A, ETHERNET, KEYBOARD, VIDEO TO LORES MONITOR

[5] SET DIAG/NORM TO DIAG AND EXECUTE SELFTEST

[6] AT THE MONITOR PROMPT > ENTER q* TO CLEAR EEPROM

[7] CONFIGURE THE EEPROM AS FOLLOWS;

LOC	VAL	LOC	VAL
14	10	B8	AA
15	10	B9	55
16	00	BC	01
		BD	04
1F	12	BE	01

[8] TYPE k2 TO CHECK SOFTWARE RESET

[9] WHEN SELFTEST STARTS, PRESS THE RESET SWITCH AND OBSERVE WATCHDOG RESET MESSAGE

Rev. 11/88

TEST PROCEDURE CONTINUED

[10] LOAD THE DIAGNOSTIC EXECUTIVE

[11] EXECUTE THE DEFAULT TEST SUITE FOR ONE ERROR FREE PASS FOR EACH OF THE FOLLOWING DIAGNOSTICS:

CPU DIAGNOSTIC
MEMORY DIAGNOSTIC
ETHERNET DIAGNOSTIC
VIDEO DIAGNOSTIC

[12] PERFORM TIME OF DAY CLOCK CALIBRATION

[13] POWER OFF AND INSTALL COLOR BOARD AND SMD CONTROLLER. ATTACH CABLES APPROPRIATELY.

[14] INSTALL SERIAL PORT LOOPBACK CABLE A<->B

[15] SET DIAG/NORM SWITCH TO NORM AND POWER ON

[16] BOOT UNIX FROM SMD DISK SUBSYSTEM

[17] CLEAR THE ERROR LOG

[18] START-UP SYSTEM DIAGNOSTIC

[19] RUN SYSDIAG FOR ONE HOUR - WITHOUT ERROR

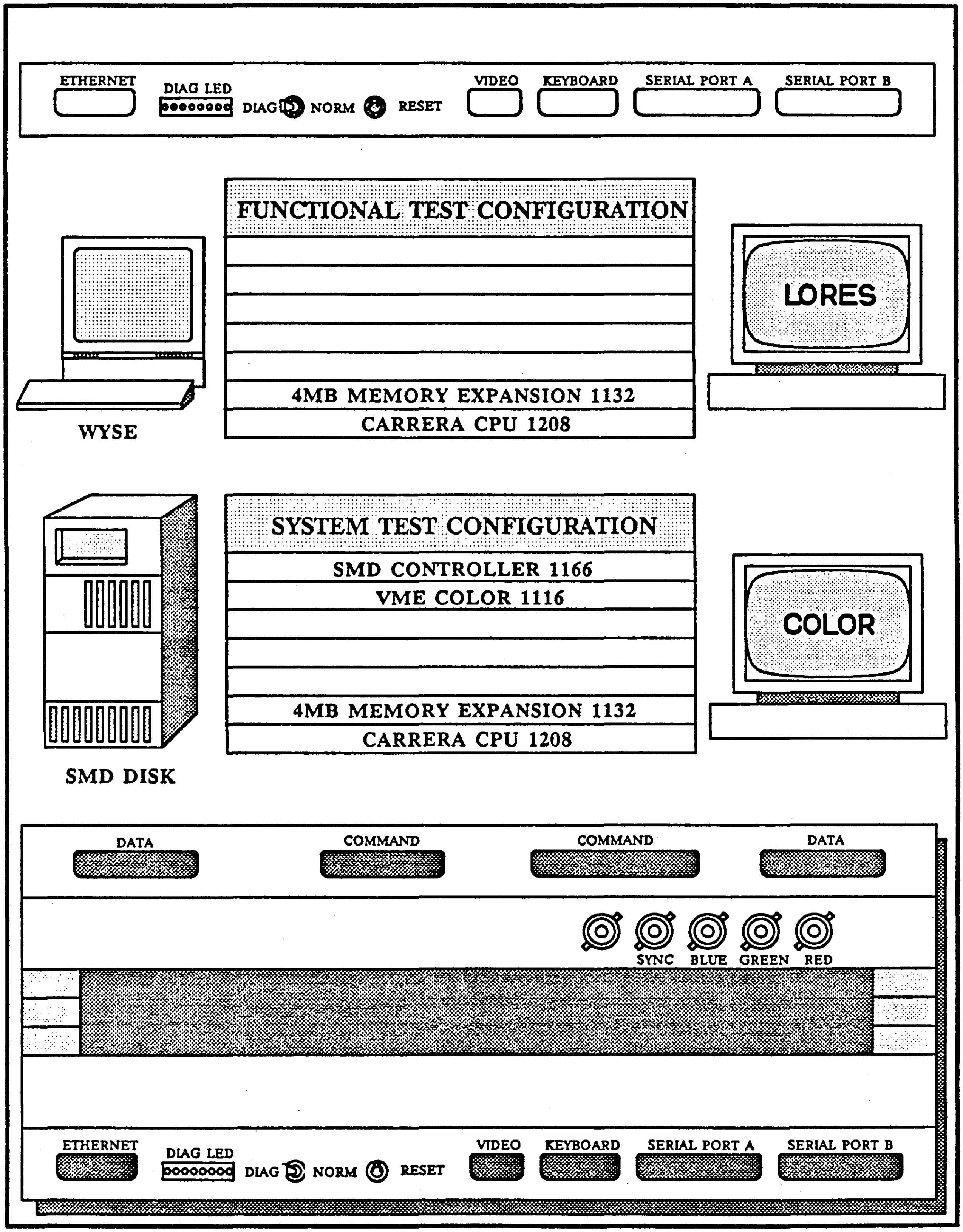
[20] CHECK ERROR LOG FOR ABSENCE OF ERRORS

[21] HALT UNIX AND POWER OFF

TEST COMPLETE

Test Procedure 501–1208

Carrera CPU Board



FUNCTIONAL TEST CONFIGURATION

- 4MB MEMORY EXPANSION 1132
- CARRERA CPU 1208

SYSTEM TEST CONFIGURATION

- SMD CONTROLLER 1166
- VME COLOR 1116
- 4MB MEMORY EXPANSION 1132
- CARRERA CPU 1208

Rev. 11/88

Test Procedure 501-1208

Carrera CPU Board

REFERENCE DOCUMENTS	
SUN-3 CPU HARDWARE ENGINEERING MANUAL	800-1163
SUNDIAGNOSTIC EXECUTIVE USER'S GUIDE	800-2326
SYSTEM DIAGNOSTIC USER'S GUIDE	800-2517
TIME OF DAY CLOCK CALIBRATION PROCEDURE	PRC-0100

EQUIPMENT REQUIREMENTS	
7/12 SLOT TEST STATION WITH 850W POWER SUPPLY	
FILESERVER ETHERNET DROP	
WYSE TERMINAL WITH RS-232 CABLE	WYSE-30
SUN KEYBOARD AND MOUSE	370-1063 / 370-1058
LO-RES MONITOR AND CABLE	540-1162 / 530-1133
SERIAL PORT LOOPBACK CABLE RS-232 A<->B	530-1205
COLOR MONITOR AND CABLE	565-1000 / 530-1307
SMD CONTROLLER BOARD	501-1166
SMD DISK SUBSYSTEM	
VME COLOR BOARD	501-1116
4MB MEMORY EXPANSION	501-1132
CARRERA ID PROM	

SOFTWARE REQUIREMENTS
DIAGNOSTIC EXECUTIVE WITH CPU, MEM, ETHER, VIDEO.EXEC FILES
SYSTEM DIAGNOSTIC (SYSDIAG) VERSION 2.60 OR LATER
VMUNIX OPERATING SYSTEM RELEASE 3.2 OR LATER

JUMPER INFORMATION

LOC	LABEL	PINS	SHUNT
S5	J300	ALL	IN
N11	J400	1-2	IN
		3-4	OUT
		5-6	OUT
		7-8	IN
E32	J1001	1-2	IN
A3	J1201	1-2	IN
	J1200	1-2	OUT
	J2502	1-2	IN
B20	J2301	1-2	IN
A7	J2503	1-2	OUT
	J2501	1-2	IN
R12	J2700	1-2	OUT
	J2701	1-2	IN
	J2702	1-2	OUT
	J2703	1-2	IN
K12	J2505	1-2	OUT
	J100	1-2	OUT
	J3102	1-2	IN
	J3101	1-2	IN

INSTALL THE CPU BOARD UNDER TEST IN THE TEST STATION AS INDICATED.

7/12 SLOT TEST STATION

SLOT	BG3	IACK	BOARD TYPE
12			
11			
10			
9			
8			
7	OUT	OUT	SMD CONTROLLER 501-1166
6	IN	OUT	VME COLOR 501-1116
5			
4			
3			
2	IN	IN	4MB MEM EXP 501-1132
1	OUT	OUT	CARRERA CPU 501-1208

Rev. 11/88

ACCEPTANCE CRITERIA:

SUCCESSFULLY EXECUTE SELFTEST AND CLEAR THE EEPROM

SUCCESSFULLY EXECUTE THE DEFAULT TEST SUITE FOR ONE ERROR FREE PASS OF EACH OF CPU.EXEC, MEM.EXEC, ETHER.EXEC, AND VIDEO.EXEC

COMPLETE TIME OF DAY CLOCK CALIBRATION

TEST THE CARRERA CPU WITH A COLOR BOARD AND SMD DISK SUBSYSTEM FOR 1.0 HOUR WITHOUT ERROR RUNNING SYSDIAG IN AUTOMATIC MODE

TEST PROCEDURE

[1] PERFORM PHYSICAL INSPECTION

[2] INSTALL TEST ID PROM AT LOCATION B10

[3] CONFIGURE THE BOARD AND INSTALL IN THE TEST STATION

[4] CONNECT CABLES: WYSE TERMINAL TO SERIAL PORT A, ETHERNET, KEYBOARD, VIDEO TO LORES MONITOR

[5] SET DIAG/NORM TO DIAG AND EXECUTE SELFTEST

[6] AT THE MONITOR PROMPT > ENTER q* TO CLEAR EEPROM

[7] CONFIGURE THE EEPROM AS FOLLOWS;

LOC	VAL	LOC	VAL
14	10	B8	AA
15	10	B9	55
16	00	BC	01
		BD	04
1F	12	BE	01

[8] TYPE k2 TO CHECK SOFTWARE RESET

[9] WHEN SELFTEST STARTS, PRESS THE RESET SWITCH AND OBSERVE WATCHDOG RESET MESSAGE

TEST PROCEDURE CONTINUED

[10] LOAD THE DIAGNOSTIC EXECUTIVE

[11] EXECUTE THE DEFAULT TEST SUITE FOR ONE ERROR FREE PASS FOR EACH OF THE FOLLOWING DIAGNOSTICS:

CPU DIAGNOSTIC
MEMORY DIAGNOSTIC
ETHERNET DIAGNOSTIC
VIDEO DIAGNOSTIC

[12] PERFORM TIME OF DAY CLOCK CALIBRATION

[13] POWER OFF AND INSTALL COLOR BOARD AND SMD CONTROLLER. ATTACH CABLES APPROPRIATELY.

[14] INSTALL SERIAL PORT LOOPBACK CABLE A<->B

[15] SET DIAG/NORM SWITCH TO NORM AND POWER ON

[16] BOOT UNIX FROM SMD DISK SUBSYSTEM

[17] CLEAR THE ERROR LOG

[18] START-UP SYSTEM DIAGNOSTIC

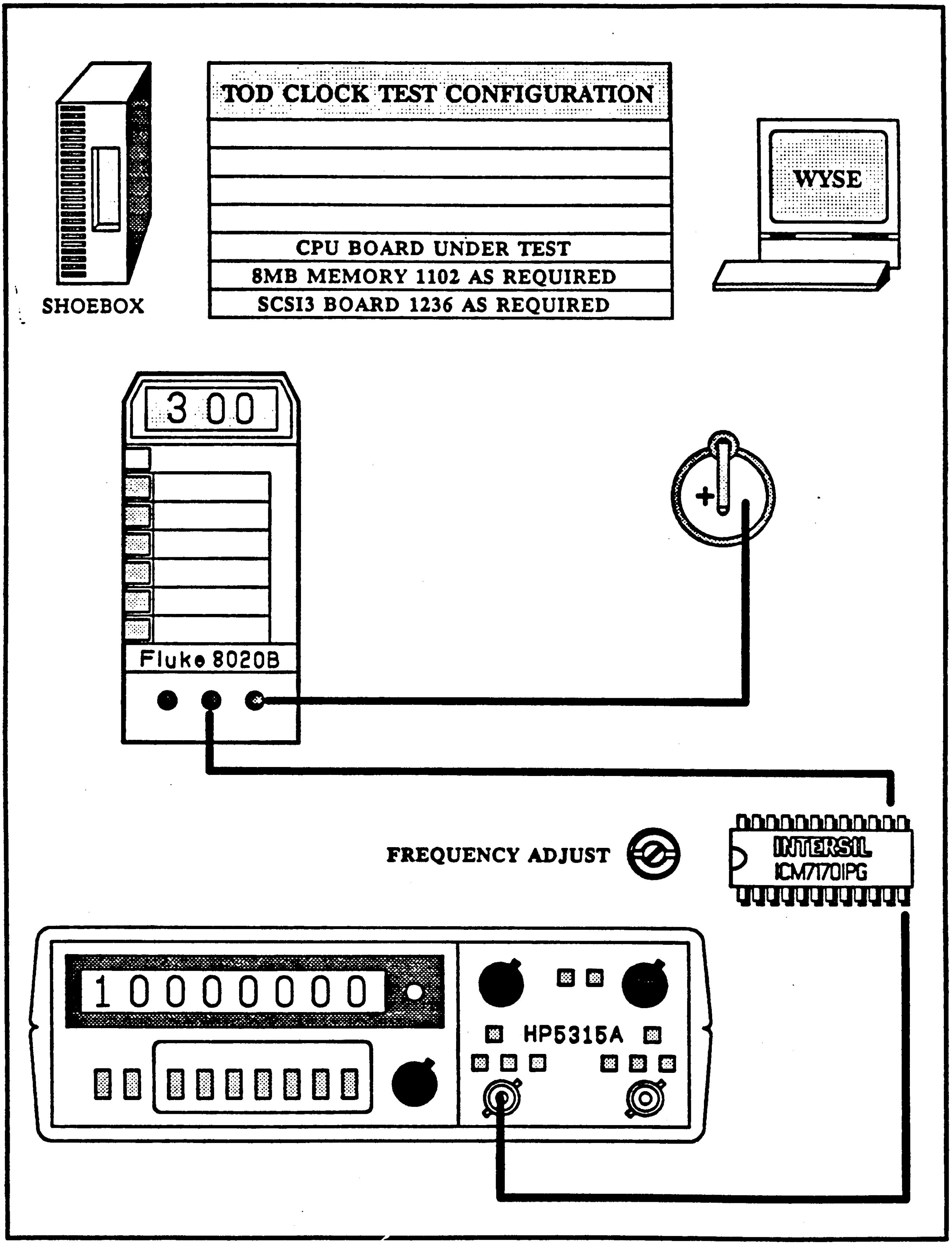
[19] RUN SYSDIAG FOR ONE HOUR - WITHOUT ERROR

[20] CHECK ERROR LOG FOR ABSENCE OF ERRORS

[21] HALT UNIX AND POWER OFF

TEST COMPLETE

Rev. 11/88


TOD CLOCK TEST CONFIGURATION
CPU BOARD UNDER TEST
8MB MEMORY 1102 AS REQUIRED
SCSI3 BOARD 1236 AS REQUIRED
SHOEBOX
WYSE
300
Fluke 8020B
+
FREQUENCY ADJUST
INTERSIL
ICM7170IPG
10000000
HP5315A

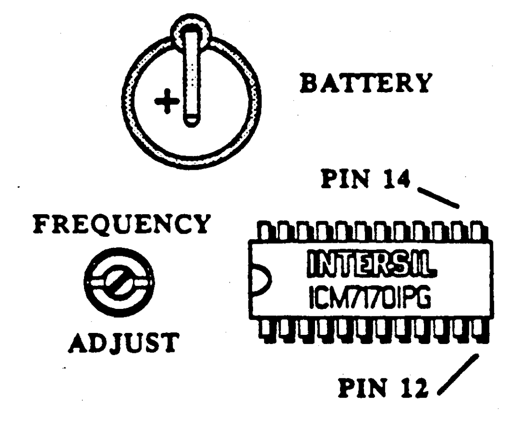
Test Procedure PRC–0100

Time of Day Clock Calibration

REFERENCE DOCUMENTS	
SUNDIAGNOSTIC EXECUTIVE USER'S GUIDE	800-2326

EQUIPMENT REQUIREMENTS	
7/12 SLOT TEST STATION WITH 850W POWER SUPPLY	
FILESERVER ETHERNET DROP	
WYSE TERMINAL WITH RS-232 CABLE	WYSE-30
SHOEBOX	595-1395
SUN CPU ID PROM	
UNIVERSAL COUNTER	HEWLET PACKARD 5315A
DIGITAL VOLT METER	FLUKE 8020B

SOFTWARE REQUIREMENTS
DIAGNOSTIC EXECUTIVE WITH CPU.EXEC FILE



INSTALL THE CPU BOARD UNDER TEST IN THE TEST STATION AS INDICATED.

7/12 SLOT TEST STATION

SLOT	BG3	IACK	BOARD TYPE
12			
11			
10			
9			
8			
7			
6			
5			
4			
3	OUT	OUT	CPU BOARD UNDER TEST
2	IN	IN	8MB MEMORY 501-1102
1	OUT	OUT	SCSI3 BOARD 501-1236

ACCEPTANCE CRITERIA:

BATTERY VOLTAGE IS GREATER THAN +3.00 VDC

CLOCK FREQUENCY IS STABLE AT 1.000000X Hz

TEST PROCEDURE

[1] INSURE TOD BATTERY IS CORRECTLY INSTALLED

[2] INSTALL THE CPU BOARD IN THE TEST STATION

[3] CONNECT CABLES: WYSE TERMINAL TO SERIAL PORT A, ETHERNET, SCSI TO SHOEBOX AS APPROPRIATE

[4] SET DIAG/NORM TO DIAG AND EXECUTE SELFTEST

[5] AT THE MONITOR PROMPT > ENTER U Aio TO SELECT THE WYSE TERMINAL FOR CONTROL/MESSAGES

[6] LOAD THE DIAGNOSTIC EXECUTIVE

[7] SELECT THE CPU DIAGNOSTIC

[8] FROM THE CLOCK TEST MENU, INITIATE THE TOD CALIBRATION TEST

[9] MEASURE THE BATTERY VOLTAGE BETWEEN THE TOP OF THE BATTERY (+) AND PIN 14 (VBAK) OF THE 7170 TOD CLOCK . THIS VOLTAGE MUST BE GREATER THAN +3.00 VDC

[11] MEASURE THE CLOCK FREQUENCY AT PIN 12 OF THE 7170 TOD CLOCK AND ADJUST THE VERICAP AS NECESSARY TO OBTAIN A STABLE READING OF 1.000000X Hz. THE SEVENTH DECIMAL POINT DIGIT (X) WILL VARY WITH EACH SAMPLE AND IS ACCEPTABLE

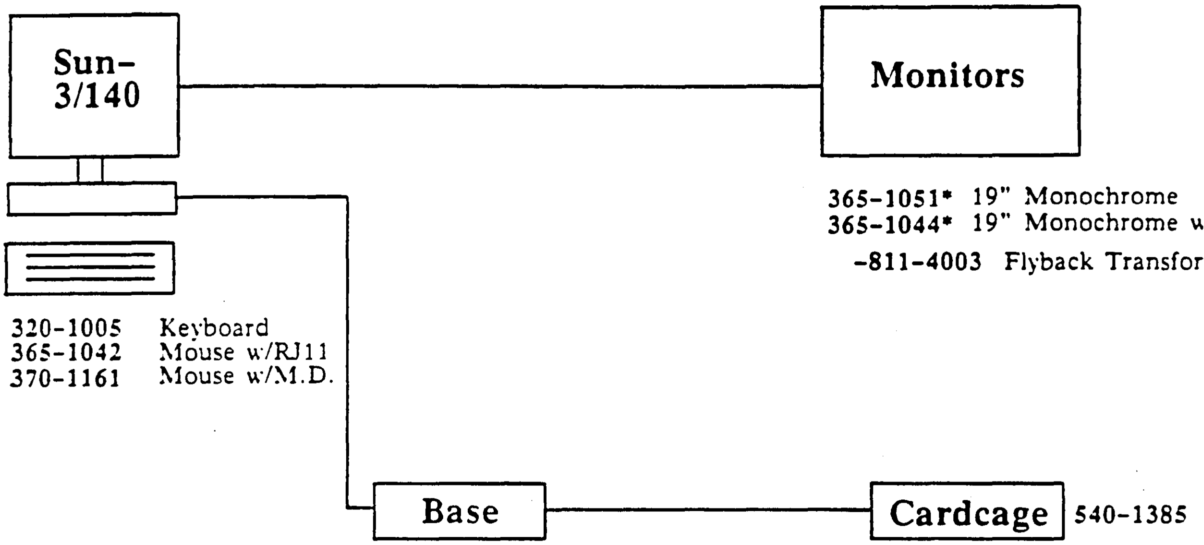
[12] IF ADJUSTMENT OF THE VARICAP WAS NECESSARY, APPLY COLORED VARNISH (TORQUE-SEAL) TO LOCK THE ADJUSTMENT

TEST COMPLETE

Spares Information

IV. Systems/Options Parts Breakdown

Sun-3/140 System Parts Breakdown



365-1051*	19" Monochrome
365-1044*	19" Monochrome w/OCLI
-811-4003	Flyback Transformer

320-1005	Keyboard
365-1042	Mouse w/RJ11
370-1161	Mouse w/M.D.

180-1097	AC Power Cord
140-1019	Fuse
501-1193	Fan PCB
540-1455	Fan Assy
300-1022*	Power Supply

501-1127	Backplane
501-1208*	CPU Bd, 4MB, w/FPP, 16MHZ
501-1132	4MB Memory Exp. Bd.
501-1158	VME/M-Bus Sunlink
501-1105	FPA
501-1153	2nd Ethernet Assy
501-1220	3x2 Adapter Bd.
501-1045	VME SCSI Bd.
501-1217	Sun-3 VME SCSI Assy

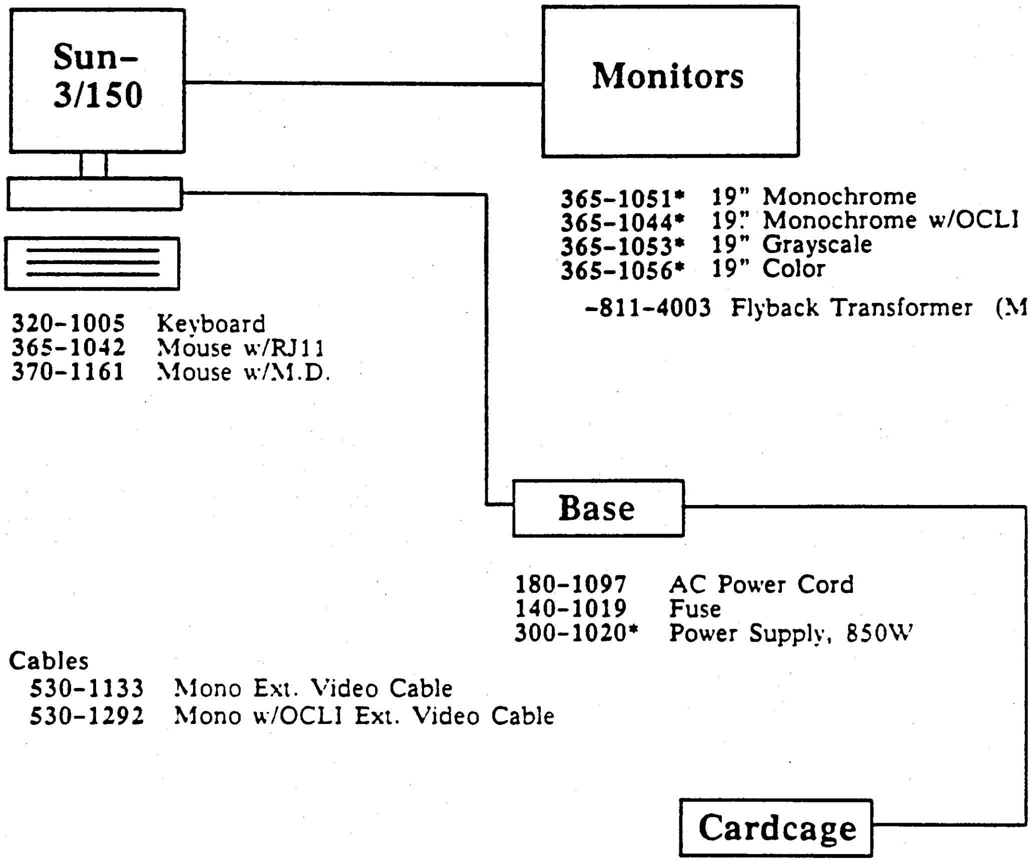
Cables

530-1133	Mono Ext. Video Cable
530-1292	Mono w/OCLI Ext. Video Cable

*Key FRU

IV. Systems/Options Parts Breakdown

Sun-3/150 System Parts Breakdown



320-1005	Keyboard
365-1042	Mouse w/RJ11
370-1161	Mouse w/M.D.

365-1051*	19" Monochrome
365-1044*	19" Monochrome w/OCLI
365-1053*	19" Grayscale
365-1056*	19" Color
-811-4003	Flyback Transformer (Mono only)

180-1097	AC Power Cord
140-1019	Fuse
300-1020*	Power Supply, 850W

Cables

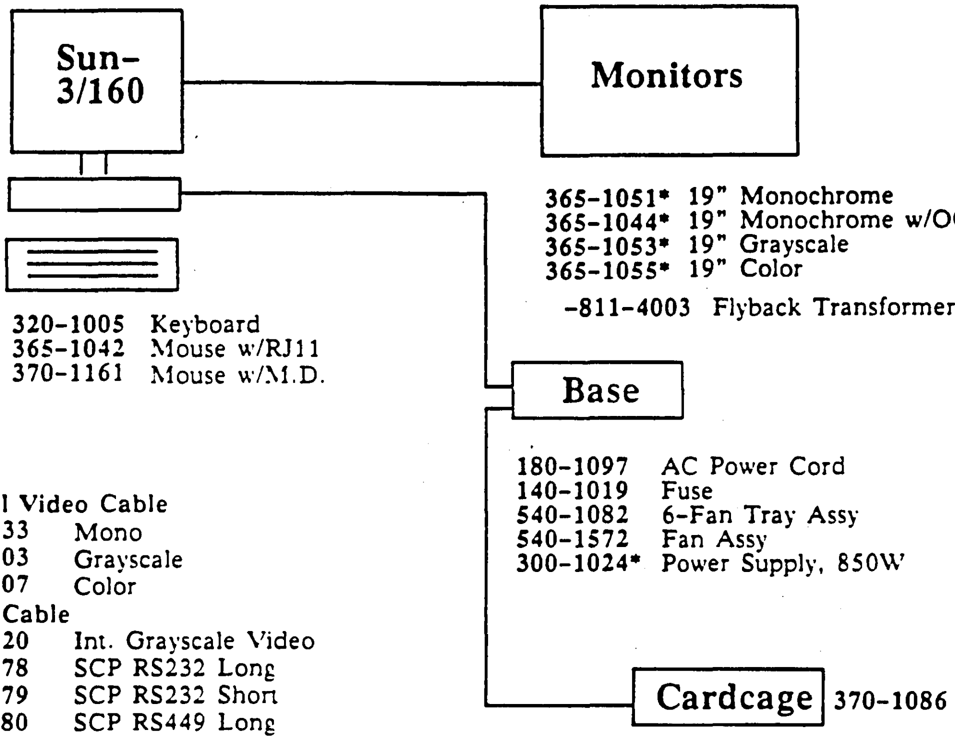
530-1133	Mono Ext. Video Cable
530-1292	Mono w/OCLI Ext. Video Cable

501-1233	6-Slot Fan Controller Board
501-1128	Backplane
501-1208*	CPU Board w/4MB
501-1116	Color Board
501-1132	4MB Memory Bd
501-1158	Sunlink Assy
501-1105	FPA
501-1153	2nd Ethernet
501-1217	Sun-3 VME SCSI Assy

*Key FRU

IV. Systems/Options Parts Breakdown

Sun-3/160 System Parts Breakdown



Sun-3/160

Monitors

365-1051*	19" Monochrome
365-1044*	19" Monochrome w/OCLI
365-1053*	19" Grayscale
365-1055*	19" Color
-811-4003	Flyback Transformer (Mono only)

320-1005	Keyboard
365-1042	Mouse w/RJ11
370-1161	Mouse w/M.D.

Base

180-1097	AC Power Cord
140-1019	Fuse
540-1082	6-Fan Tray Assy
540-1572	Fan Assy
300-1024*	Power Supply, 850W

Cables

External Video Cable	
530-1133	Mono
530-1303	Grayscale
530-1307	Color
Output Cable	
530-1220	Int. Grayscale Video
530-1178	SCP RS232 Long
530-1179	SCP RS232 Short
530-1180	SCP RS449 Long
530-1181	SCP RS449 Short
530-1251	ALM Controller
530-1252	ALM J2-J5
530-1255	ALM5-Conductor Power
530-1147	SMD Disk Data
530-1148	SMD Disk Command
530-1173	Int. Ethernet
530-1165	Int. 1/2" Tape 1600/6250 BPI
AC	
530-1207	Power Input Harness
530-1231	Internal Harness
DC	
530-1322	Top Fan Cable
530-1324	Power Harness
530-1206	PS to Backplane
Misc	
530-1323	Air Box Harness, 4-Fans

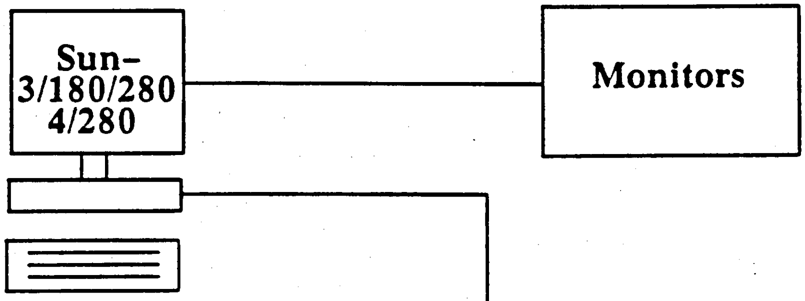
Cardcage 370-1086

501-1117	Backplane
501-1208*	CPU Bd, 4MB, w/FPP/16MHZ
501-1131	Memory Expansion Board, 2MB
501-1132	Memory Expansion Board, 4MB
501-1116	Color Board
501-1267	CG5
Options	
501-1153	2nd Ethernet Assy
501-1054	VME to Multibus Adapter Board
501-1269	VME to Eurocard Adapter Assy
501-1170	Sun2 3x2 Adapter w/SCSI Assy
501-1045	Sun2 SCSI Board
501-1158	SunLink Assy
501-1166	Xylogics 451 SMD Disk Option Assy
370-1082	Xylogics 472 Tape Controller Board
501-1155	Xylogics 472 1/2" Tape Option Assy
370-1067	Xylogics 472 Tape Controller Board
501-1157	16 Chan ALM Communications Assy
501-1268	Graphics Processor,2
501-1058	Graphics Buffer
501-1105	Floating Point Accelerator Board
501-1125	IPC Board
501-1383	TAAC-1™
501-1203	ALM-2

*Key FRU

IV. Systems/Options Parts Breakdown

Sun-3/180, 3/280 and 4/280 System Parts Breakdown



Cables

Output

530-1220	Int. Grayscale Video
530-1178	SCP RS232 Long
530-1179	SCP RS232 Short
530-1180	SCP RS449 Long
530-1181	SCP RS449 Short
530-1251	ALM Controller
530-1252	ALM J2-J5
530-1255	ALM5 Conductor Power
530-1147	SMD Disk Data
530-1148	SMD Disk Command
530-1173	Int. Ethernet (2nd PCB)
530-1165	Int. 1/2" Tape 1600/6250 BP

Base

180-1097	AC Power Cord
140-1019	Fuse
540-1366	4-Fan Tray Assy
540-1509	Fan Assy
300-1024*	Power Supply, 850W

Cardcage 370-1086

501-1208*	CPU Bd (3/180)
501-1206*	CPU Bd (3/280)
501-1522*	CPU Bd (4/280)
501-1131	Memory Expansion Board, 2MB
501-1132	Memory Expansion Board, 4MB
501-1116	Color Board
540-1443	VME Rear Filler Panel

Options

501-1153	2nd Ethernet Assy
501-1054	VME to Multibus Adapter Board
501-1059	VME to Eurocard Adapter Assy
501-1217	Sun2 3x2 Adapter w/SCSI Assy
501-1045	Sun2 SCSI Board
501-1158	SunLink Assy
501-1166	Xylogics 451 SMD Disk Option Assy
370-1082	Xylogics 472 Tape Controller Board
501-1155	Xylogics 472 1/2" Tape Option Assy
370-1067	Xylogics 472 Tape Controller Board
501-1157	16 Chan ALM Communications Assy
501-1268	Graphics Processor, 2
501-1058	Graphics Buffer
501-1105	Floating Point Accelerator Board
501-1125	IPC Board
501-1203	ALM II

*Key FRU