

# Contents

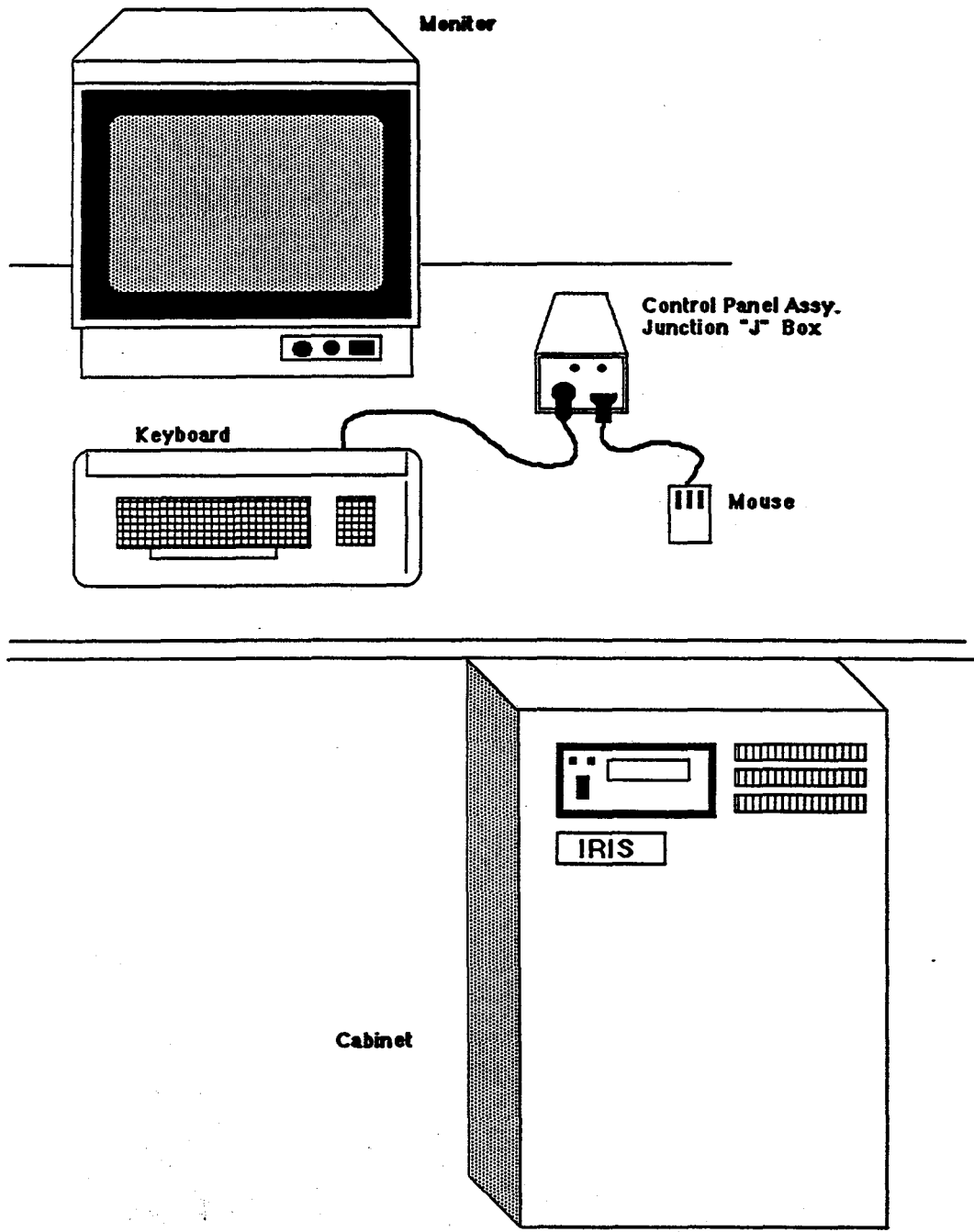
<b>LESSON 5: COMPONENT LOCATION</b> .....	1
MAJOR IRIS COMPONENTS .....	2
2400 TURBO CABINET BACK PANEL .....	3
MONITOR BACK PANEL .....	4
I/O COMPARTMENT COMPONENT LOCATION .....	5
POWER COMPARTMENT COMPONENT LOCATION .....	6
BASIC 68020 PCBA ASSIGNMENTS (4 meg, no fpa) .....	7
PCBA SLOT CONFIGURATION GUIDE .....	8
PCBA SWITCH SETTINGS - page 1 .....	9
PCBA SWITCH SETTINGS - page 2 .....	10
PCBA SWITCH SETTINGS - page 3 .....	11
PCBA SWITCH SETTINGS - page 4 .....	12
68020 COMPONENTS MATRIX - page 1 .....	13
68020 COMPONENTS MATRIX - page 2 .....	14
2400T SPARE PARTS STOCKING GUIDELINES - page 1 .....	15
2400T SPARE PARTS STOCKING GUIDELINES - page 2 .....	16

## **LESSON 5: COMPONENT LOCATION**

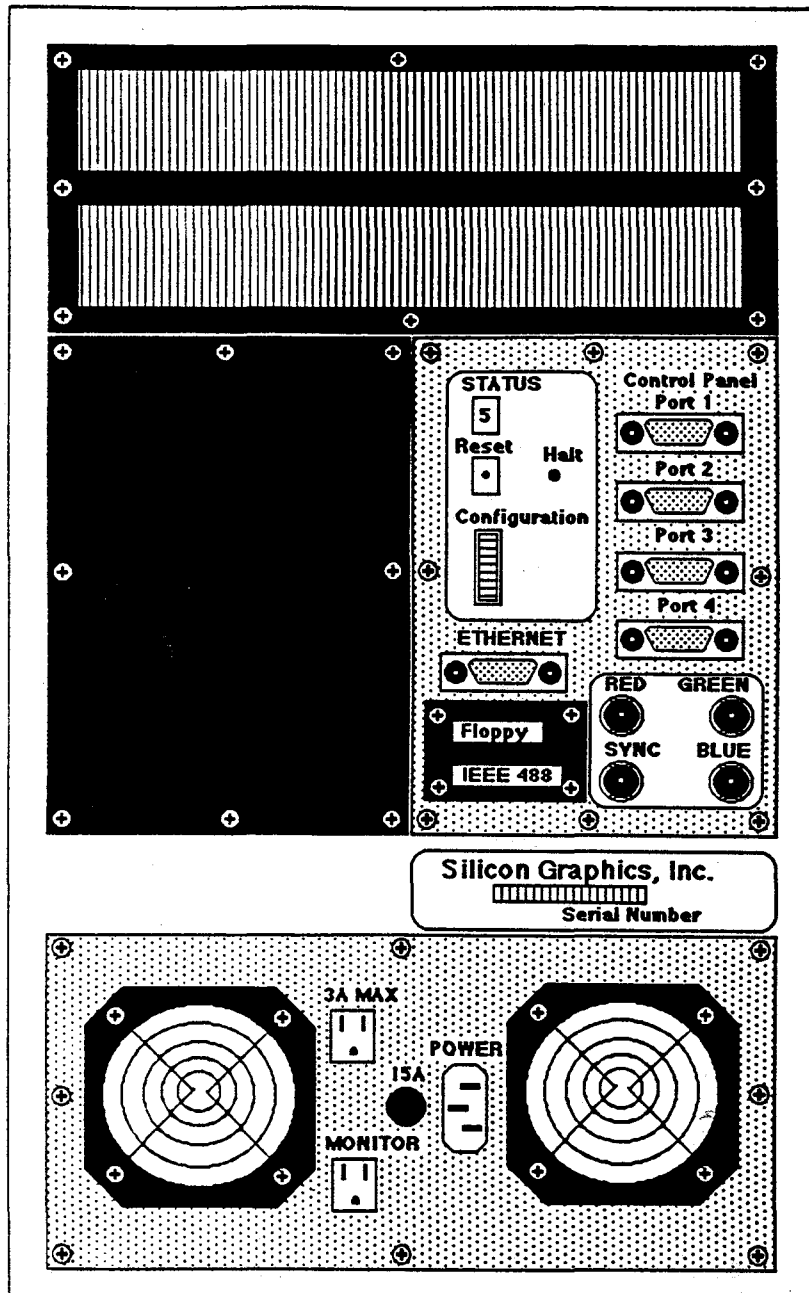
Upon completion of this lesson, the student, using available documentation will be able to:

- **Locate, remove, and re-install the following FRU's: Power supply, Winchester disk, Quarter inch tape drive, CPU board, Tape/Disk controller, Frame buffer controller, Update controller, Display controller, Bit plane boards, and memory boards.**
- **Locate specific hardware configuration switches and demonstrate that the switches can be set to select or enable a stated function.**

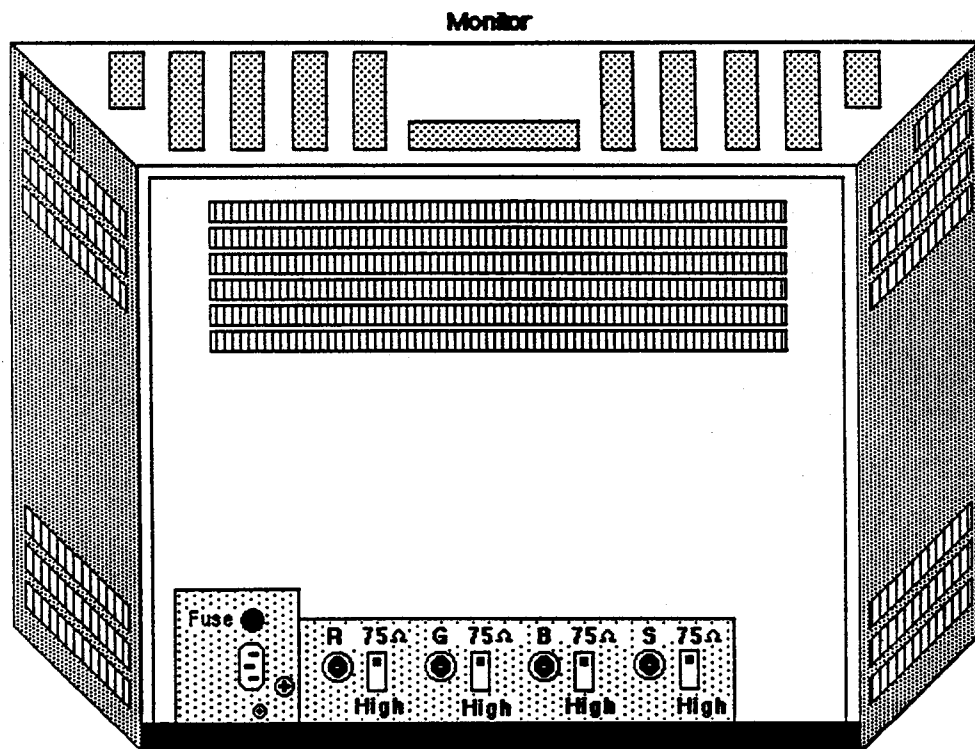
**MAJOR IRIS COMPONENTS**



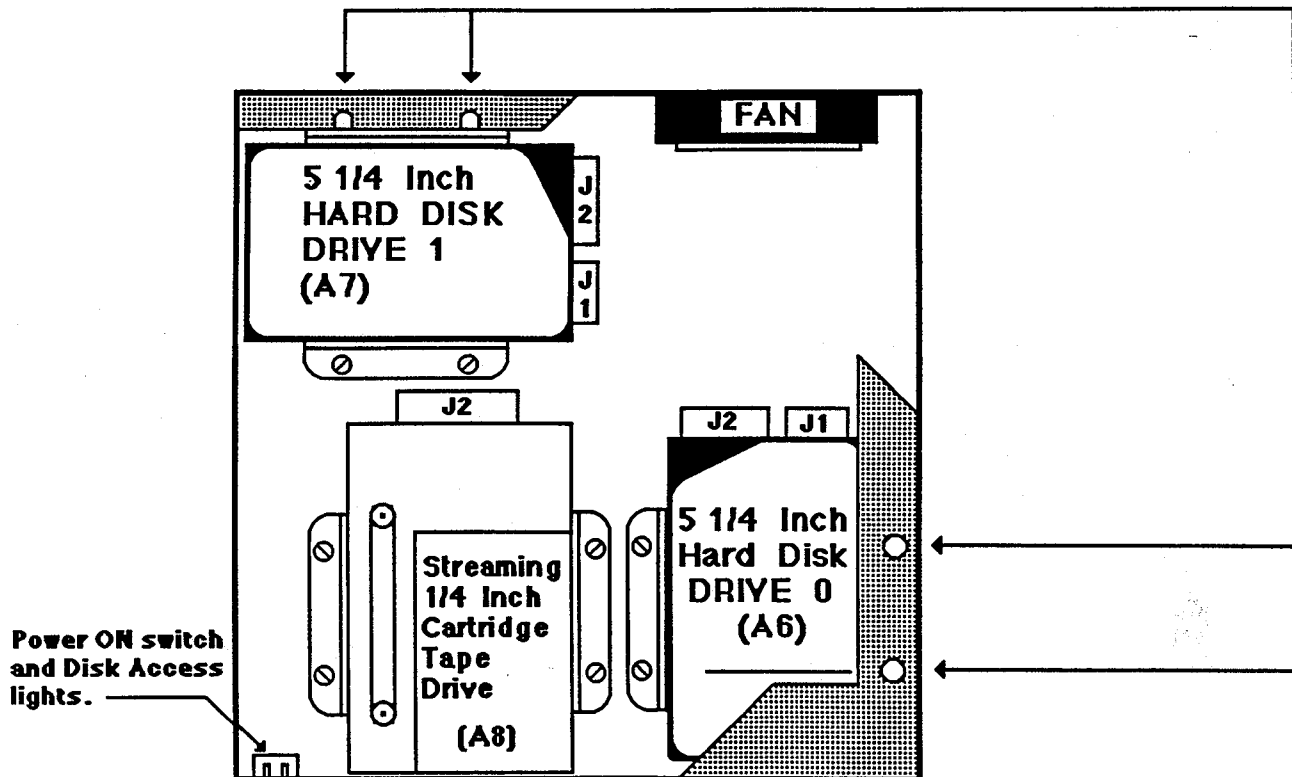
2400 TURBO CABINET BACK PANEL



**MONITOR BACK PANEL**

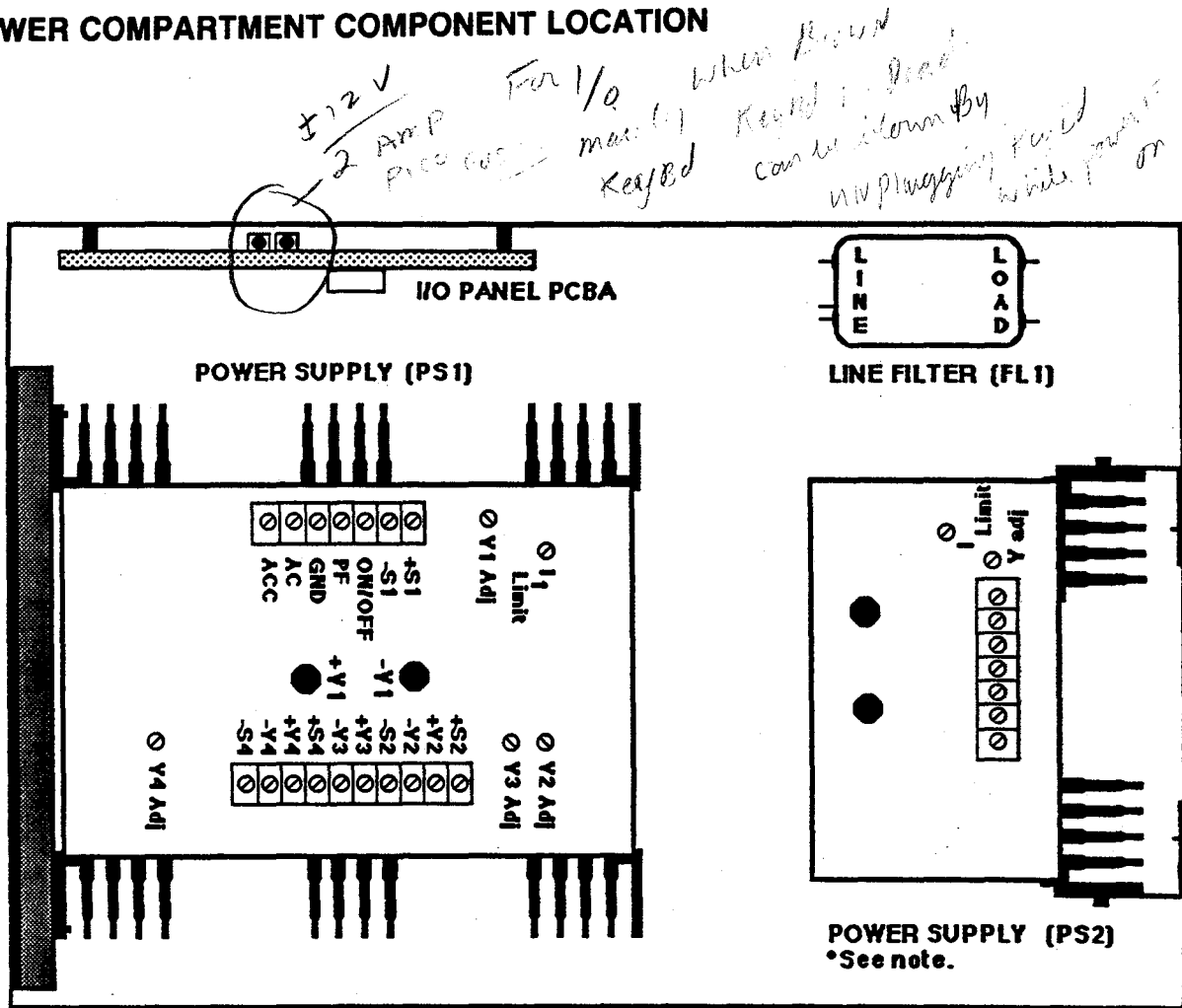


## I/O COMPARTMENT COMPONENT LOCATION

**NOTE:**

When removing the Hard Disk Drives, you will need a screw driver with a long neck that will allow you access to the two screws that are obstructed by the flange that is part of the sheet metal that comprises the workstation housing. Part of the flange showing access holes for the disk fastening screws is shown and highlighted by the drawing.

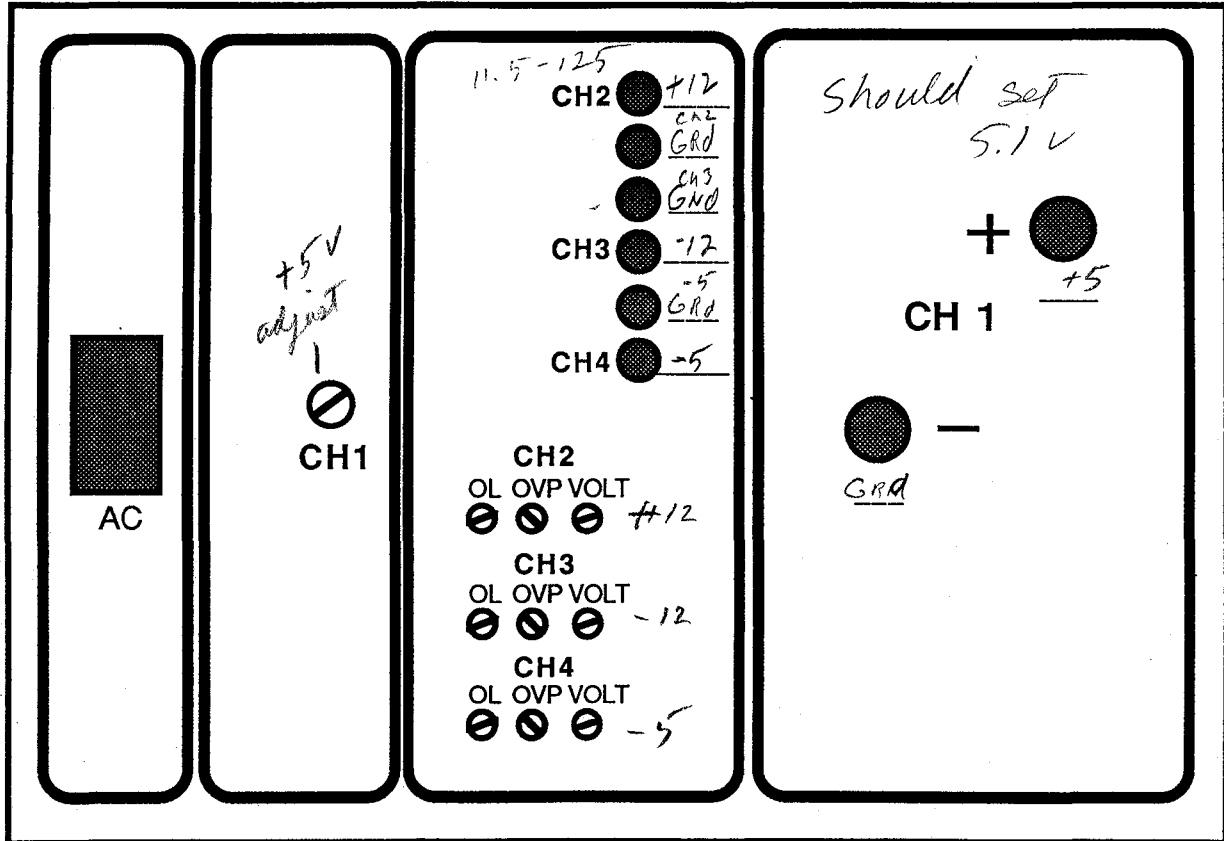
POWER COMPARTMENT COMPONENT LOCATION



**NOTE:**

This power supply is added to 2400 workstations when they are upgraded to a 2400 Turbo. The existing supply (an LM-54) does not have enough current drive to supply all the options. The additional supply (an LM-11) **CAN ONLY** be paralleled with an LM-54. For installation instructions, see document # 003-0002.

# 3000 POWER COMPARTMENT



notes:

1. OL (OverLimit protection) : output current of channel
2. OVP (Over Voltage Protection) : output voltage of channel (shut down voltage)





**PCBA SLOT CONFIGURATION GUIDE**

*2400/2500  
order of Bd made  
a difference  
3000 doesn't  
matter*

**Processor Side  
(without Floating Point Accelerator)**

SLOT	1	2	3	4	5	6	7	8	9
	IM1	IP2	ENET	DSD	(IEEE)				(CG1)
	(IM1)	IM1	IP2	ENET	DSD	(IEEE)			(CG1)
	(IM1)	(IM1)	IM1	IP2	ENET	DSD	(IEEE)		(CG1)
	(IM1)	(IM1)	(IM1)	IM1	IP2	ENET	DSD	(IEEE)	(CG1)

*GenLock Bd  
video*

( ) Indicates an Option

**Processor Side  
(with Floating Point Accelerator)**

SLOT	1	2	3	4	5	6	7	8	9
	IM1	(FP1)	IP2	ENET	DSD	(IEEE)			(CG1)
	(IM1)	IM1	(FP1)	IP2	ENET	DSD	(IEEE)		(CG1)
	(IM1)	(IM1)	IM1	(FP1)	IP2	ENET	DSD	(IEEE)	(CG1)
	(IM1)	(IM1)	(IM1)	IM1	(FP1)	IP2	ENET	DSD	(CG1)

( ) Indicates an Option

**Graphics Side**

Slot	10	11	12	13	14	15	16	17	18	19	20
							BP3	BP3	DC4	UC4	GF2
						(BP3)	BP3	BP3	DC4	UC4	GF2
				(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2
			(BP3)	(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2
		(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2
	(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2
	(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2

*min configure*

PCBA SWITCH SETTINGS - page 1

Memory and Bit Plane boards.

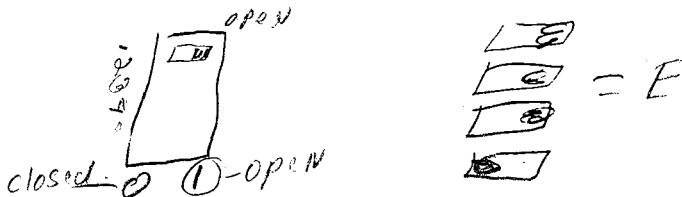
**Memory Switches:** The switch pack is located on the front edge of each board. Each pack contains four switches.

*MOST SIGNIF. BIT* →

		Switch				
		1	2	3	4	
Board	1	C	C	C	C	0
	2	C	C	C	O	1
	3	C	C	O	C	2
	4	C	C	O	O	3

LSB →

-C- = Closed 0  
-O- = Open 1



**Bit Plane Switches:** The switch pack is located on the front edge of each board. There are four switches per pack, and the setting produces a hex address starting at 8.

*1 0 0 0 = 8*

		Switch				
Addr	Slot	1	2	3	4	
8	17	O	C	C	C	-8
9	16	O	C	C	O	-9
A	15	O	C	O	C	10
B	14	O	C	O	O	11
C	13	O	O	C	C	12
D	12	O	O	C	O	13
E	11	O	O	O	C	14
F	10	O	O	O	O	15

-C- = Closed  
-O- = Open

PCBA SWITCH SETTINGS - page 2

IP2 - Switch S2

The IP2 has a 16 bit switch register. S1 on the processor board controls the low order bits and S2 controls the high order bits. Closing a switch gives 0 and leaving it open gives a 1. S1 is connected to the switches on the back panel. The individual switches are anded together, so if either of the switches is closed, the bit will be zero. Leaving all of the switches open on S1 will enable all of the backpanel switches.

The following table shows what S2 controls. Note: "C" means closed and "O" means open.

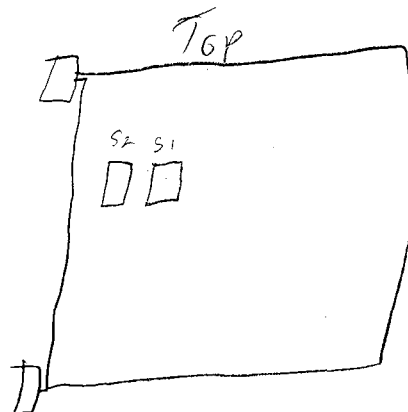
Category	Switch setting								Meaning
	1	2	3	4	5	6	7	8	
master/slave	C	X	X	X	X	X	X	X	Master processor
	O	X	X	X	X	X	X	X	Slave processor
RS232 Speed for ports 1-3	X	X	X	C	C	X	X	X	9600 baud
	X	X	X	C	O	X	X	X	300 baud
	X	X	X	O	C	X	X	X	1200 baud
	X	X	X	O	O	X	X	X	19.2K baud
Display types Primary/Secondary	X	X	X	X	X	C	C	C	NI/NI (DC4:5000-090-03)
	X	X	X	X	X	C	C	O	NI/I (DC4:5000-090-03)
	X	X	X	X	X	C	O	C	NI/RS (DC4:5000-090-02)
	X	X	X	X	X	C	O	O	NI/EU (DC4:5000-090-05)
	X	X	X	X	X	O	C	C	I/NI (DC4:5000-090-03)
	X	X	X	X	X	O	C	O	I/I (DC4:5000-090-03)
	X	X	X	X	X	O	O	C	I/RS (DC4:5000-090-04)
	X	X	X	X	X	O	O	O	I/EU (DC4:5000-090-06)

choice of 2

NI-non-interlaced 60 Hz  
I-interlaced 30 Hz - video  
RS-RS170A - video  
EU-European

any pin # is open = 1

Default all  
Switches closed



PCBA SWITCH SETTINGS - page 3

IP2 - Switch S1

*Default all SWs open*

The following table shows what S1 controls. Remember that S1 and the backpanel switches are wired together. It is recommended that all of the S1 switches be left open. This will give the backpanel switches complete control.

Category	Switch setting								(S1) (BACK)	Meaning
	1 9	2 7	3 6	4 5	5 4	6 3	7 2	8 1		
Boot Environment	X	X	X	X	C	C	C	C		(0) disk boot
	X	X	X	X	C	C	C	O		(1) tape boot
	X	X	X	X	C	C	O	C		(2) floppy boot
	X	X	X	X	C	C	O	O		(3) network boot
	X	X	X	X	C	O	C	C		(4) not used
	X	X	X	X	C	O	C	O		(5) PROM monitor (ie no device)
	X	X	X	X	C	O	O	C		(6) Diagnostic PROM board boot
	X	X	X	X	C	O	O	O		(7) not used
	X	X	X	X	O	C	C	C		(8) not used
	X	X	X	X	O	C	C	O		(9) ip - interphase boot
	X	X	X	X	O	C	O	C		(a) st - storager 2 tape boot
	X	X	X	X	O	C	O	O		(b) sf - storager 2 floppy boot
	X	X	X	X	O	O	C	C		(c) sd - storager 2 disk boot
	X	X	X	X	O	O	C	O		(d) mt - DSD tape boot
X	X	X	X	O	O	O	C		(e) mf - DSD floppy boot	
X	X	X	X	O	O	O	O		(f) md - DSD disk boot	
Autoboot	X	X	X	C	X	X	X	X		don't autoboot
	X	X	X	O	X	X	X	X		autoboot
Quiet	X	X	C	X	X	X	X	X		print hardware information
	X	X	O	X	X	X	X	X		don't print information
Display	X	C	X	X	X	X	X	X		Use primary monitor for display
	X	O	X	X	X	X	X	X		Use secondary monitor

*LOOK AT S2 (6,7,8)*

The recommended switch settings are S1 all open, S2 all closed, and the backpanel switches all closed. This will configure the IP2 to be a master processor with a 60 Hz monitor that will boot from the disk.

*IF S1 sw's are all open Then Backpanel SW's are enabled.*

PCBA SWITCH SETTINGS - page 4

**Disk Drives**

The following illustrations show you where to locate the address jumpers for the different types of disk drives found in SGI workstations. If you replace a disk drive, insure all the other jumpers on the new drive match the drive that you remove.

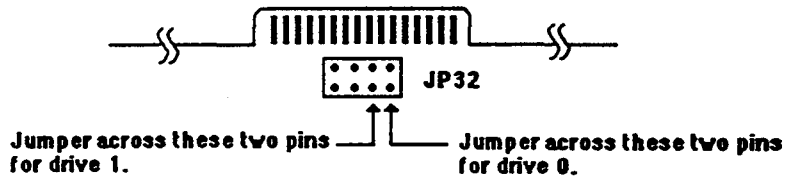
**Vertex 72 MB Drive**

Address selection is controlled by switch pack "J6" on the logic PCB located on the under side of the disk drive. J6 is located near and between the two edge pin connectors.

For drive 0: Switch 4 is on, switch 7 is off, and switch 8 is on.  
For drive 1: Switch 4 is on, switch 7 is on, and switch 8 is off.

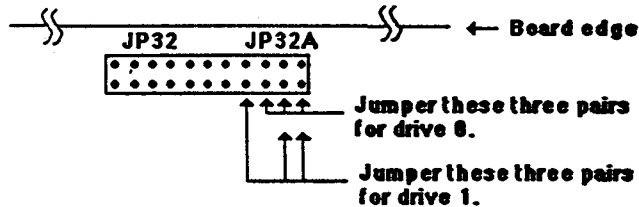
**Hitachi 85 MB Drive**

Address selection is done using jumpers across pins at jumper platform "JP32". This platform is located near the large pin edge connector. The following illustration shows how to set the jumpers.



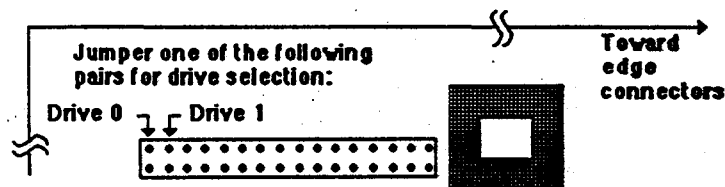
**Hitachi 170 MB Drive**

For this drive address selection is set using jumpers on jumper platform JP32A. It is located along the edge of the logic PCB next to JP32. The following illustration shows what pins are used.



**Fujitsu 170 MB Drive**

For this drive address selection is set using jumpers on a jumper platform that has no name. It has 16 pairs of pins and is located at the end of the logic board that is away from the edge pin connectors. It is next to a large square chip. See following illustration for desired pin connections.



68020 COMPONENTS MATRIX - page 1

Description	SYSTEM TYPE				Part Number
	2400T	3010	3020	3030	
<b>Processor</b>	IP2	IP2	IP2	IP2	5000-561-02
<b>Floating Point</b>	FP1	FP1	FP1	FP1	5000-573
<b>Memory</b>	IM1	IM1	IM1	IM1	5000-532-01 (2MB) 5000-532-02 (4MB)
<b>CONTROLLERS</b>				STOR II	
<b>Disk-Tape-Flpy</b>					026-0005-001
<b>Disk-Tape-Flpy</b>	DSD5217		DSD5217		6100-120-06
<b>Disk-Tape</b>		DSD5214			6100-120-07
<b>Color Printer</b>			X	X	6100-120-26
<b>Tape Drive 1/2"</b>	X		X	X	6100-120-30
<b>COMMUNICATION</b>					
<b>IEEE-488</b>		X			6100-120-10
<b>IEEE-488</b>	X		X	X	6100-120-11
<b>3270 Interface</b>	X	X	X	X	6100-120-27
<b>Enhanced Ethernet 201</b>	X	X	X	X	6100-120-28
<b>GRAPHICS</b>					
<b>Bitplane</b>	BP3	BP3	BP3	BP3	5000-503
<b>Display Control</b>	DC4	DC4	DC4	DC4	
<b>NI-I</b>	↓	↓	↓	↓	6100-100-1
<b>NI-RS170A</b>	↓	↓	↓	↓	6100-1100-31
<b>I-RS170A</b>	↓	↓	↓	↓	6100-100-33
<b>NI-EURO</b>	↓	↓	↓	↓	6100-100-34
<b>I-EURO</b>	↓	↓	↓	↓	6100-100-35
<b>NI-RS170A GENLOCK</b>	↓	↓	↓	↓	6100-100-36
<b>I-RS170A GENLOCK</b>	↓	↓	↓	↓	6100-100-37

68020 COMPONENTS MATRIX - page 2

Description	SYSTEM TYPE				Part Number
	2400T	3010	3020	3030	
NI-EURO GENLOCK	DC4	DC4	DC4	DC4	6100-100-38
I-EURO GENLOCK	DC4	DC4	DC4	DC4	6100-100-39
Update Cntrl	UC4	UC4	UC4	UC4	6100-100-18
Frame Buffer Controller	GF2	GF2	GF2	GF2	
NO-Z Clipping Z Clipping					6100-100-30 6100-100-29
Synchronized Master & Clock Generator	CG1	CG1	CG1	CG1	
RS170A EURO					6100-100-32 6100-100-40
<b>MASS STORAGE</b>					
170 MB Disk				X	Fujitsu 026-0006-2 Hitachi 026-0006-3
340 MB DISK				X	Fujitsu 026-0006-5 Hitachi 026-0006-6
Internal Floppy		X			026-0011-003
Cartridge Tape	X		X	X	026-0011-002
20 MB Disk		X			Tandon 026-0012-1
72 MB Disk	X		X		Vertex 026-0012-2
144 MB Disk	X		X		Vertex 026-0012-3
External Floppy	X		X		6100-110-02
Tape Drive 1/2"	X		X	X	6000-330-01
Color Printer	X	X	X	X	
<b>MONITORS</b>					
15 Inch	X	X	X	X	
19 Inch	X	X	X	X	



## 2400T SPARE PARTS STOCKING GUIDELINES - page 1

Apr 25 1986 REV B

RECOMMENDED SPARES STOCKING GUIDELINES					
Kit type:		Depot	Branch	Service	Unit Price
Part Number	Description	Qty	Qty	Qty	
019-0001-002	Cable, IM1/IP2 40 Cond	2	1	1	\$53.50
019-0002-002	Cable, IM1/IP2 50 Cond	2	1	1	\$63.20
019-0005-001	Cable, FP1/IP2 20 Cond	2	1	1	\$17.00
5000-037-02	IO Panel PCB	2	1	1	\$255.60
5000-069	Backplane, 20 slot	1	1	0	\$1,991.25
5000-090	PC Assy, DC4	2	1	1	\$3,402.70
5000-136-02	Control Panel Assy, J Box	1	1	0	\$415.00
5000-173-3	PC Assy, Disk Controller	2	1	1	\$2,400.00
5000-201-1	ESS Cable, GF to UC	2	1	1	\$45.90
5000-201-3	ESS Cable, GF to PM	2	1	1	\$45.90
5000-204-1	Cable, GF to UC	2	1	1	\$72.10
5000-204-2	Cable, BP Sub 2X	2	1	1	\$108.10
5000-204-3	Cable, BP Sub 3X	2	1	1	\$144.20
5000-204-4	Cable, BP Sub 4X	2	1	1	\$180.20
5000-204-6	Cable, BP Sub 6X	2	1	1	\$108.10
5000-208	ESS Cable, DC-Vid	2	1	1	\$425.90
5000-210-1	ESS Cable, PM-I/O	2	1	1	\$86.40
5000-213-1	Video Monitor Cable	2	1	1	\$156.60
5000-214-1	Cable, I/O-Monitor	2	1	1	\$337.10
5000-224-2	Hrd Disk Cntrl Cable	2	1	1	\$85.90
5000-226-2	Hrd Disk Data Cable	2	1	1	\$59.80
5000-230-2	Tape Dr Cable	2	1	1	\$85.20
5000-503	PC Assy, BP3	2	1	1	\$2,500.00
5000-511	PC Assy, UC4	2	1	1	\$2,904.00
5000-515-08	PC Assy,GF2 125 ns w/z	2	1	1	\$7,500.00
5000-532-01	PC Assy, IM1 4 Meg.	2	1	1	\$10,000.00
5000-532-02	PC Assy, IM1 2 Meg	2	1	1	\$6,000.00
5000-551	IP2 PCB Assy	2	1	1	\$9,450.00
6000-004	I/O Panel Assy, Turbo	2	1	0	\$0.00
6000-02-1	I/O Panel Assy, 20 Slot	1	1	0	\$353.90
6000-165	DC Power Harness	1	1	0	\$6.90
6000-285-01	Assy, Junction Box	0	0	0	\$809.10
7010001	Fuse 5A	20	10	5	\$3.40
7010002	Fuse 2A	20	10	5	\$3.40
9090003	Jumper (for PC Assy)	10	5	5	\$3.80
9110007	Power Switch	2	1	1	\$25.00

**2400T SPARE PARTS STOCKING GUIDELINES - page 2**

RECOMMENDED SPARES STOCKING GUIDELINES					
Kit type:		Depot	Branch	Service	Unit Price
Part Number	Description	Qty	Qty	Qty	
9150001	Mouse	2	1	1	\$573.75
9180002	15A Circuit Breaker	2	1	1	\$31.70
9300002	EMI Filter 20 Slot	1	1	1	\$189.40
9310002	Fan, 20 Slot Chassis	2	1	1	\$164.60
9320002	Caster	4	4	0	\$11.60
9330004	Monitor, 60hz N/I	2	1	0	\$6,000.00
9350001	Fan Cord	2	1	0	\$8.10
9350003	Power Cord, 20 Slot	2	1	1	\$78.30
9410003	72 MB Disk Dr	2	1	0	\$5,000.00
9420001	1/4" Tape Dr	2	1	1	\$2,100.00
9430002	Power Supply, 20 Slot	2	1	1	\$2,416.50
9500001	Keyboard	2	1	1	\$579.95

The total price for the Depot spares kit, Model 2400T is: **\$130107.45.**

The total price for the Branch spares kit, Model 2400T is: **\$66555.15.**

The total price for the Service spares kit, Model 2400T is: **\$52699.60.**

# Contents

<b>LAB 5.1: Component Location</b> .....	1
<b>SECTION 1: Instructional Procedure</b> .....	2
<b>I. OBJECTIVE</b> .....	2
<b>II. PURPOSE</b> .....	2
<b>III. EQUIPMENT.</b> .....	2
<b>IV. PROCEDURE</b> .....	2
<b>SECTION 2: Diagrams</b> .....	7
<b>DIAGRAM 1: 2400T I/O Panel Cable Connections</b> .....	8
<b>DIAGRAM 2: Basic 68020 PCB assignments - page 1</b> .....	9
<b>DIAGRAM 3: Basic 68020 PCB assignments - page 2</b> .....	10
<b>DIAGRAM 4: IP2 interconnect - page 1</b> .....	11
<b>DIAGRAM 5: IP2 interconnect - page 2</b> .....	12
<b>DIAGRAM 6: Ethernet interconnect</b> .....	13
<b>DIAGRAM 7: DSD interconnect</b> .....	14
<b>DIAGRAM 8: Graphics sub-system interconnect</b> .....	15
<b>DIAGRAM 9: PCB switch settings - page 1</b> .....	16
<b>DIAGRAM 10: PCB switch settings - page 2</b> .....	17
<b>DIAGRAM 11: PCB switch settings - page 3</b> .....	18
<b>DIAGRAM 12: PCB switch settings - page 4</b> .....	19
<b>DIAGRAM 13: I/O compartment components location</b> .....	20
<b>DIAGRAM 14: Power compartment components location</b> .....	21
<b>DIAGRAM 15: 2400T AC power distribution</b> .....	22
<b>DIAGRAM 16: 2400T DC power distribution</b> .....	23
<b>DIAGRAM 17: 2400T I/O power distribution</b> .....	24
<b>SECTION 3: Review</b> .....	25



## **LAB 5.1: Component Location**

**This lab project contains three sections:**

- Section 1:** This section is used to teach you where the various components are located and how they are interconnected.
  
- Section 2:** This section contains all the diagrams and tables referred to throughout the lab project.
  
- Section 3:** This is a questionnaire that you will complete and hand in after performing the lab project.

## **SECTION 1: Instructional Procedure**

### **I. OBJECTIVE**

When you finish this lab project you should be able to:

- A. Correctly indicate where a specific optional PCBA would plug into the card cage.
- B. Correctly state the PCBA switch settings for any system configuration.
- C. Locate and remove any component in the workstation and then correctly re-install the part.
- D. Given a workstation with all the PCBA and I/O interconnect cables removed, correctly re-install all the cables within 45 minutes.

### **II. PURPOSE**

Teach the student how the 3x30 Workstations are configured and allow time for each student to locate and remove specific components from the workstation.

### **III. EQUIPMENT.**

- A. A functional IRIS 3030 or 3130 workstation.
- B. Classroom tool kit.
- C. ESD wrist straps

### **IV. PROCEDURE**

Two students will be assigned one IRIS workstation. The two students will work as a team.

Follow the steps of the lab project carefully. When you are through, power-on your workstation and insure that it functions correctly.

**STEP 1: Perform a reboot command.**

**STEP 2: Power off the workstation.**

**Step 3: Remove the power cord from the back of the workstation.**

**Step 4: Remove the following panels:**

- **Front panel:**  
Removed by inserting a screw driver or other thin object into the seam that separates the front panel and side panels, and twisting screw driver slightly. The front panel should pop off.
- **Top panel:**  
Removed by firmly grasping the front edge of the panel and lifting up. Like the front panel, this panel is held in place by "snap in" type connectors.
- **Side panels:**  
These panels are held in place by phillip head screws that are easy to spot.
- **Card cage panel:**  
This panel is on the front of the workstation and held in place by phillip head screws. You do not have to completely remove the screws.
- **I/O compartment panel:**  
This panel is located on top of the workstation, it is the larger of two panels on top and held in place by phillip head screws. Like the card cage panel, its screws do not have to be completely removed to remove the panel.
- **Power compartment panel:**  
This is the smaller panel on top of the workstation. It too is held in place by phillip head screws.

**STEP 5: Rear panel connections.**

Using diagram #1, verify the connections at the rear of the workstation.

You will notice that your Ethernet *Drop cable* attaches to a box called the *Multiport Transceiver*. This box contains an amplifier for each connection. Only the box is connected to the main Ethernet cable. (Later in the course you will be shown how to connect to the main Ethernet cable)

Two notes of interest:

- The monitor used on the classroom 2400 Turbos is different from the monitor used on the 30X0 workstations. The RGB cable connections are clearly called out on the 30X0 monitors, but the way each line is terminated is different.

On the old style monitors (2400T) the line termination is controlled by switches (which you will see) on the back panel of the monitor, whereas, terminators on the new style monitors (30X0) are attached to the monitor using a beaded chain.

The principle of termination remains the same for both type of monitors; **only the last monitor in a string of monitors will be terminated.**

The line termination must be 75 ohms for normal sync and color drive.

- The keyboard used with the old style monitors needs the *Junction Box*. The keyboard and mouse plug into the "J-box" and the "J-box" is connected to port 1 of the workstation.

The new style keyboards contain the logic that is contained in the "J-box", therefore, the mouse plugs into the keyboard and the keyboard is connected to port 1. (The new style keyboards allow the mouse to plug into the *right* or *left* end of the keyboard)

You will also discover two additional connectors on ports 2 and 3 of the workstation not shown by diagram 1. These are the two ASCII terminals you have been using during the UNIX portion of the class.

Now, check the cable connections and monitor terminator switches. When you are satisfied continue with the project.

**STEP 6: Card cage configuration.**



Using diagrams 2-12, remove all the cards from your workstation.

**NOTE:**

**DO NOT** handle the cards in a manner that brings your body in contact with the card edge connectors. **DAMAGE** may occur caused by *STATIC DISCHARGE* into the card. When you remove or move cards about, use the **ESD wrist straps** provided at each workstation and use the special bags that cards are shipped in.

- Verify switch settings on cards with switch packs.
- Look at other function jumpers on the cards, but **DO NOT** move any of the jumpers. Remember, if you replace a card in the field, insure that all the jumpers on the replacement card match the card removed.
- Look for part numbers, serial numbers, and revision levels on the cards.
  - SGI cards will have their numbers along the edge somewhere.
  - Cards purchased from vendors will not have an SGI part number on them, i.e. DSD, Ethernet, Storager II.
- Re-install the cards and cable together per diagrams 2-12.

**STEP 7: Removing mass storage devices.**

Using diagrams 7 and 13 and a *long neck* screw driver, remove the disk and tape drives.

**NOTE:**

When you shut down your workstation the heads of the disk drives were positioned off the recording surface. No special *PARK* command is required.

- Using diagram 12, verify the address switch settings on the disk PCB's.
- Re-install the drives and cables per diagrams 7 and 17.

*LED's on processor.*

**STEP 8: Power compartment components.**

Using diagrams 14-17, verify the location and cabling of the various components in the power compartment.

**DO NOT** remove the power supply (or supplies if you have the additional supply), but locate the screws that fasten them. You can loosen the screws just to verify that you have located the correct ones.

Also, **DO NOT** remove any of the power supply harness wires, you will get to do this some day in the field, just verify their routes.

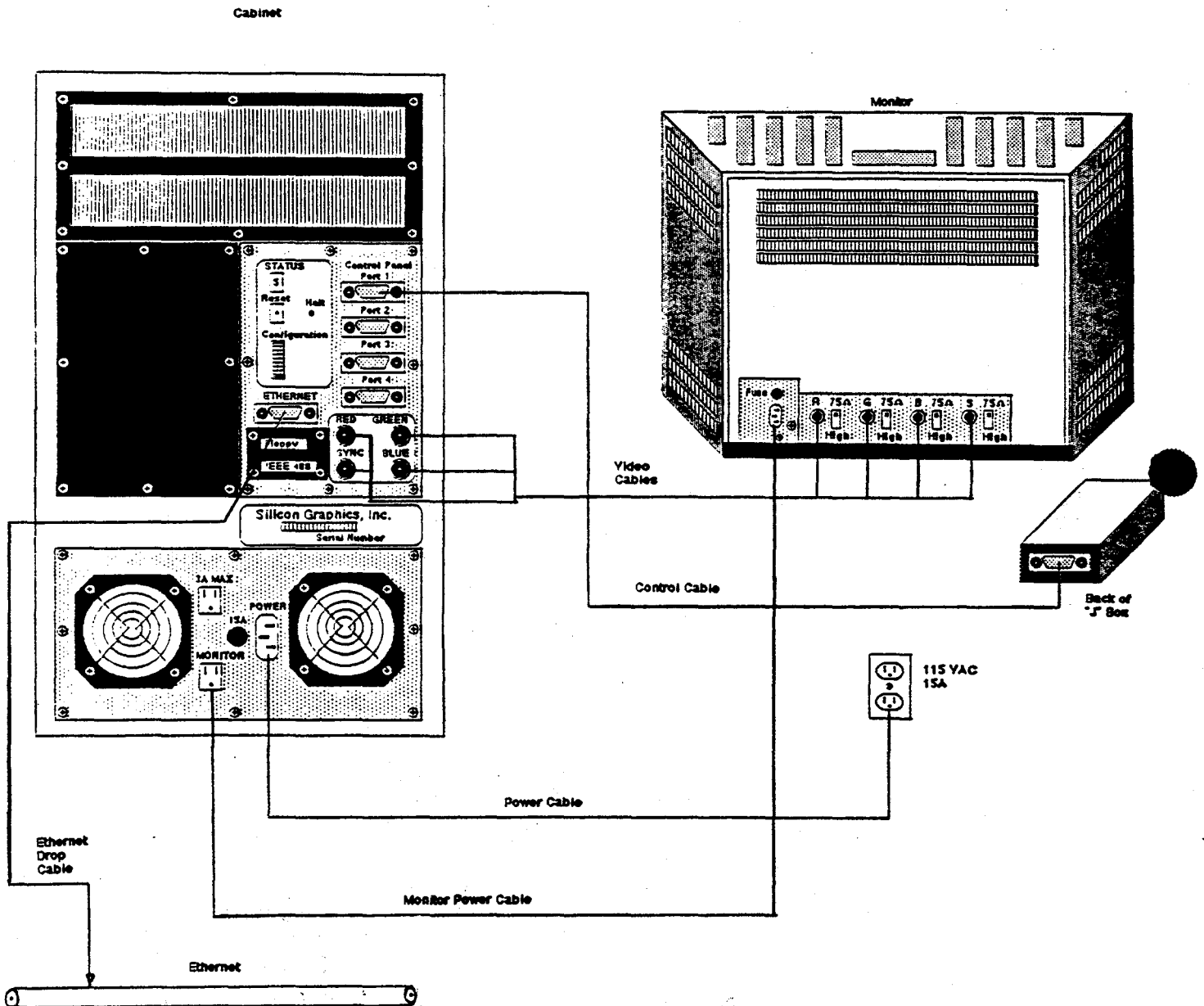
**STEP 9: Clean up and test.**

Re-install all parts and panels and power on your workstation and verify that it functions O.K.

When you are satisfied that your workstation is functioning correctly, complete section 3 and hand it into your instructor.

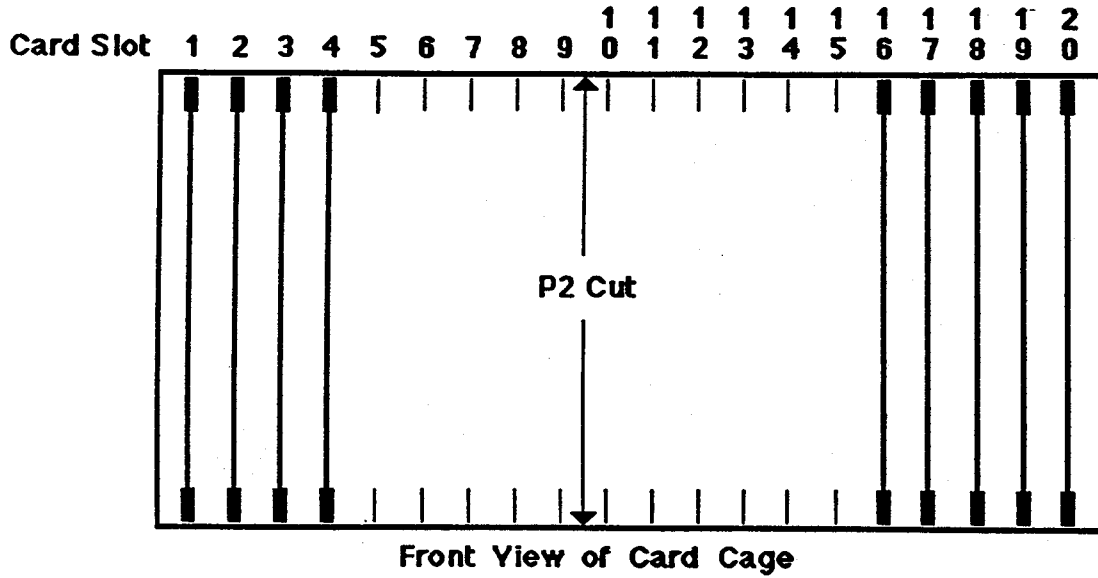
**SECTION 2: Diagrams**

DIAGRAM 1: 2400T I/O Panel Cable Connections

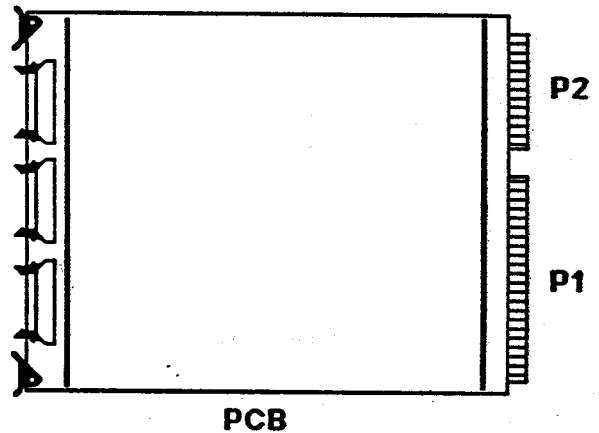


**DIAGRAM 2: Basic 68020 PCB assignments - page 1**

4 Meg, No FPA



SLOT	PCB	DESCRIPTION
1	IM1	Memory Board
2	IP2	Processor
3	Enet	Ethernet
4	DSD	Disk / Tape Controller
5	Option	For Location of Options, See Next Page.
6	Option	
7	Option	
8	Option	
9	Option	
10	Option	
11	Option	
12	Option	
13	Option	
14	Option	
15	Option	
16	BP3	Bit Plane
17	BP3	Bit Plane
18	DC4	Display Controller
19	UC4	Update Controller
20	GF2	Frame Buffer



**Basic PCB Slot Assianments**

**DIAGRAM 3: Basic 68020 PCB assignments - page 2**

**Processor Side  
(without Floating Point Accelerator)**

SLOT	1	2	3	4	5	6	7	8	9
	IM1	IP2	ENET	DSD	(IEEE)				(CG1)
	(IM1)	IM1	IP2	ENET	DSD	(IEEE)			(CG1)
	(IM1)	(IM1)	IM1	IP2	ENET	DSD	(IEEE)		(CG1)
	(IM1)	(IM1)	(IM1)	IM1	IP2	ENET	DSD	(IEEE)	(CG1)

( ) Indicates an Option

**Processor Side  
(with Floating Point Accelerator)**

SLOT	1	2	3	4	5	6	7	8	9
	IM1	(FP1)	IP2	ENET	DSD	(IEEE)			(CG1)
	(IM1)	IM1	(FP1)	IP2	ENET	DSD	(IEEE)		(CG1)
	(IM1)	(IM1)	IM1	(FP1)	IP2	ENET	DSD	(IEEE)	(CG1)
	(IM1)	(IM1)	(IM1)	IM1	(FP1)	IP2	ENET	DSD	(CG1)

( ) Indicates an Option

**Graphics Side**

Slot	10	11	12	13	14	15	16	17	18	19	20
							BP3	BP3	DC4	UC4	GF2
						(BP3)	BP3	BP3	DC4	UC4	GF2
					(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2
			(BP3)	(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2
		(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2
	(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2
	(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	(BP3)	BP3	BP3	DC4	UC4	GF2

DIAGRAM 4: IP2 interconnect - page 1

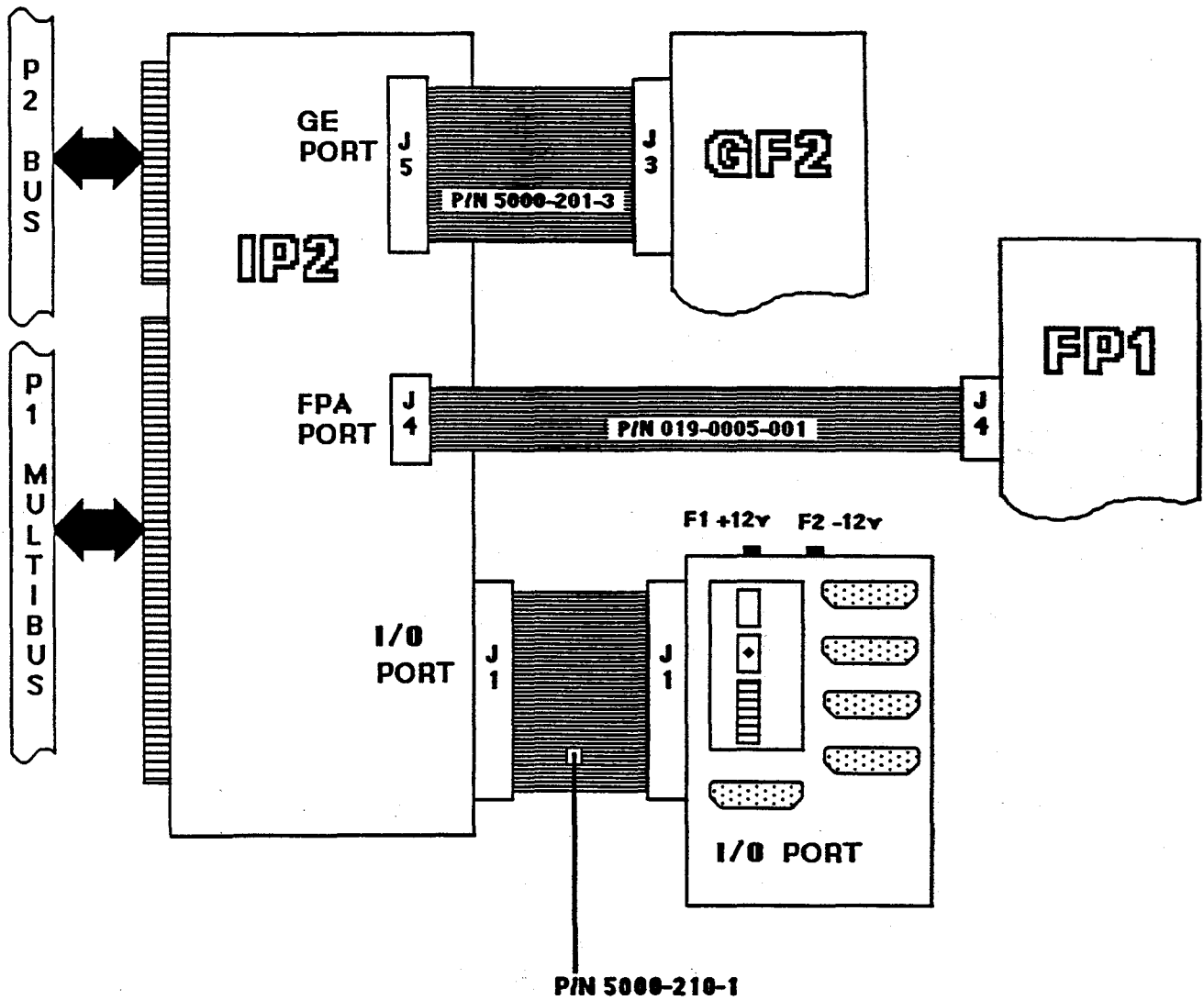
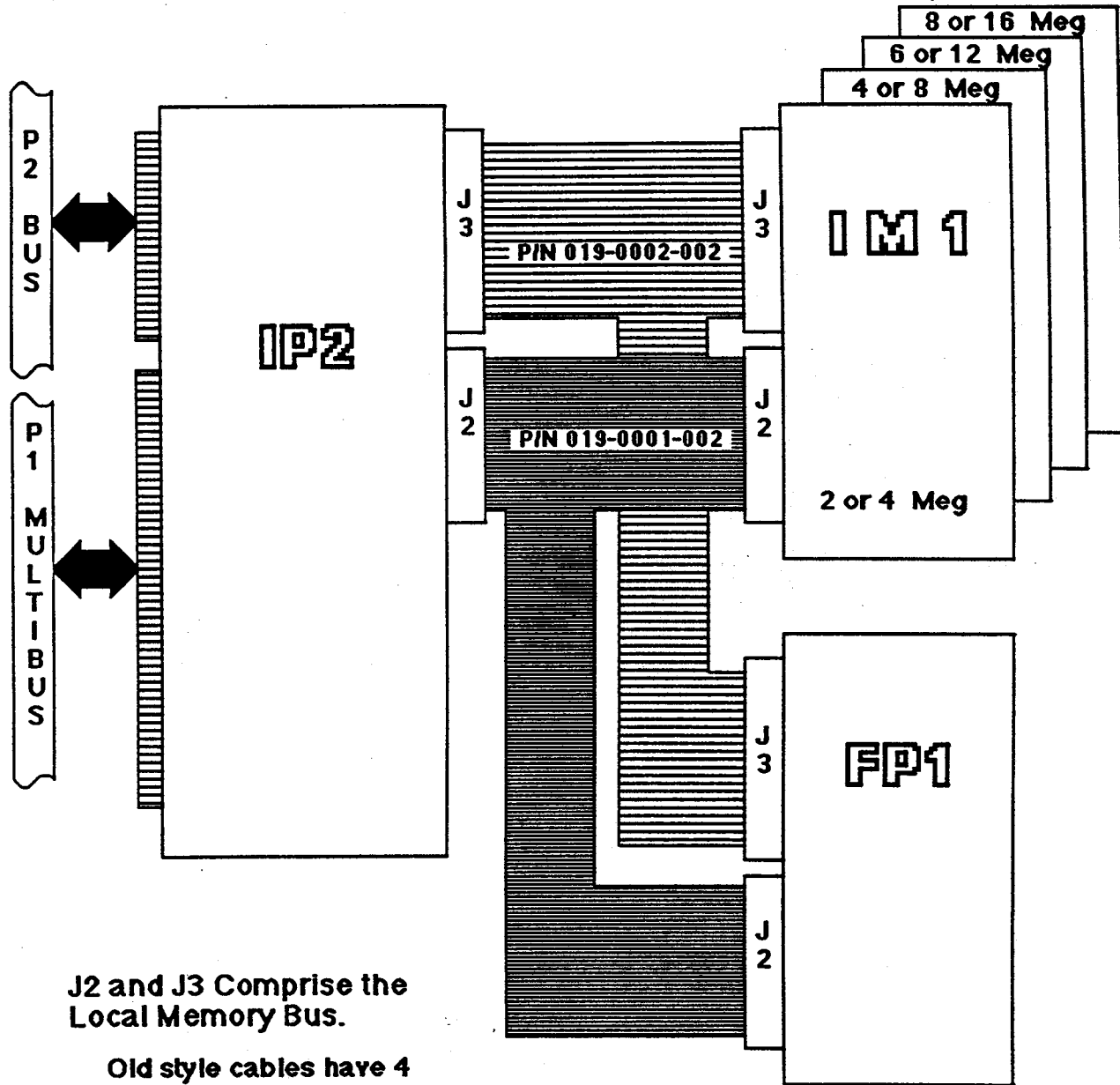


DIAGRAM 5: IP2 interconnect - page 2



J2 and J3 Comprise the Local Memory Bus.

Old style cables have 4 connectors.

New style cables have 6 connectors.

Note: Only a system with 4 memory boards and an FP1 will use all connectors.



DIAGRAM 6: Ethernet interconnect

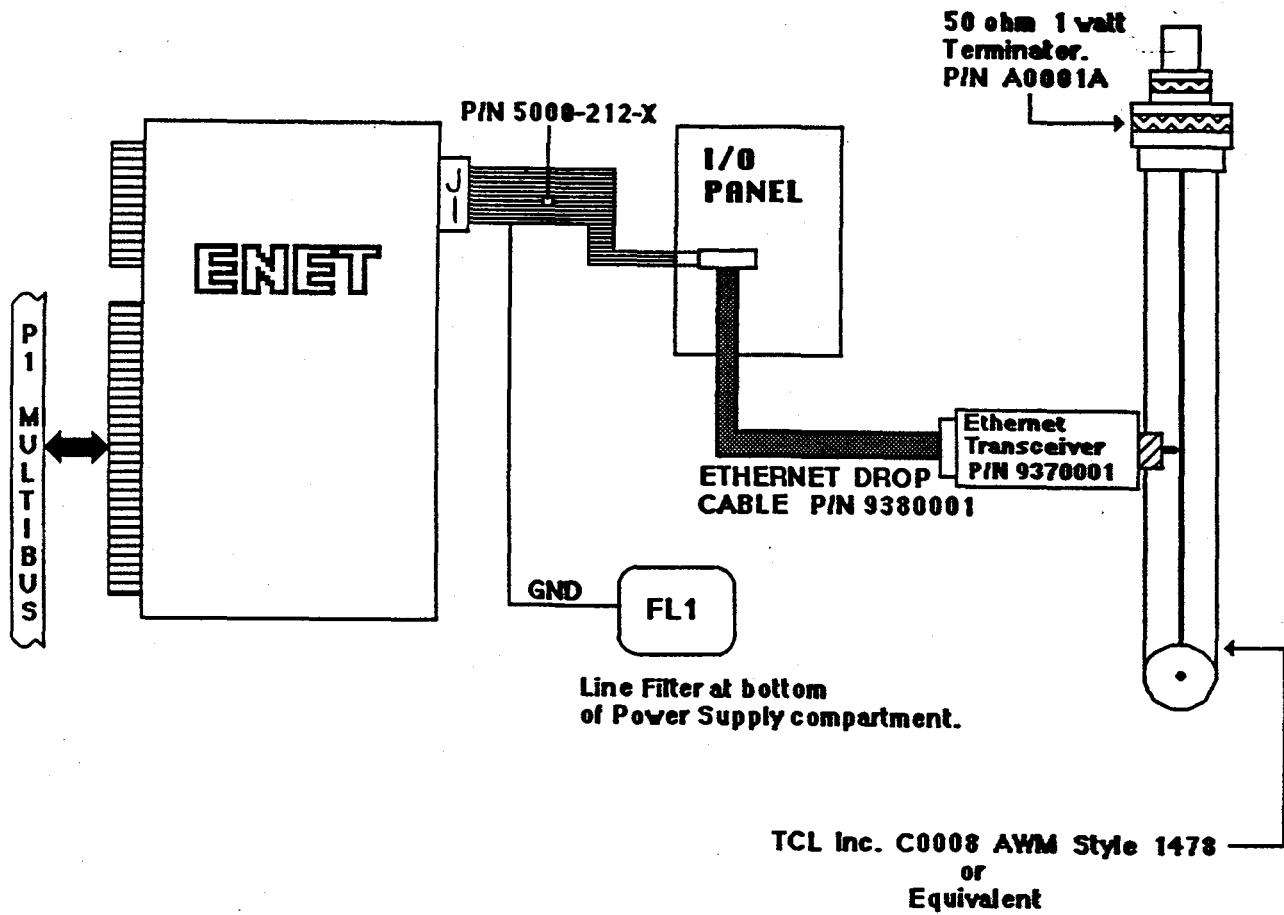


DIAGRAM 7: DSD interconnect

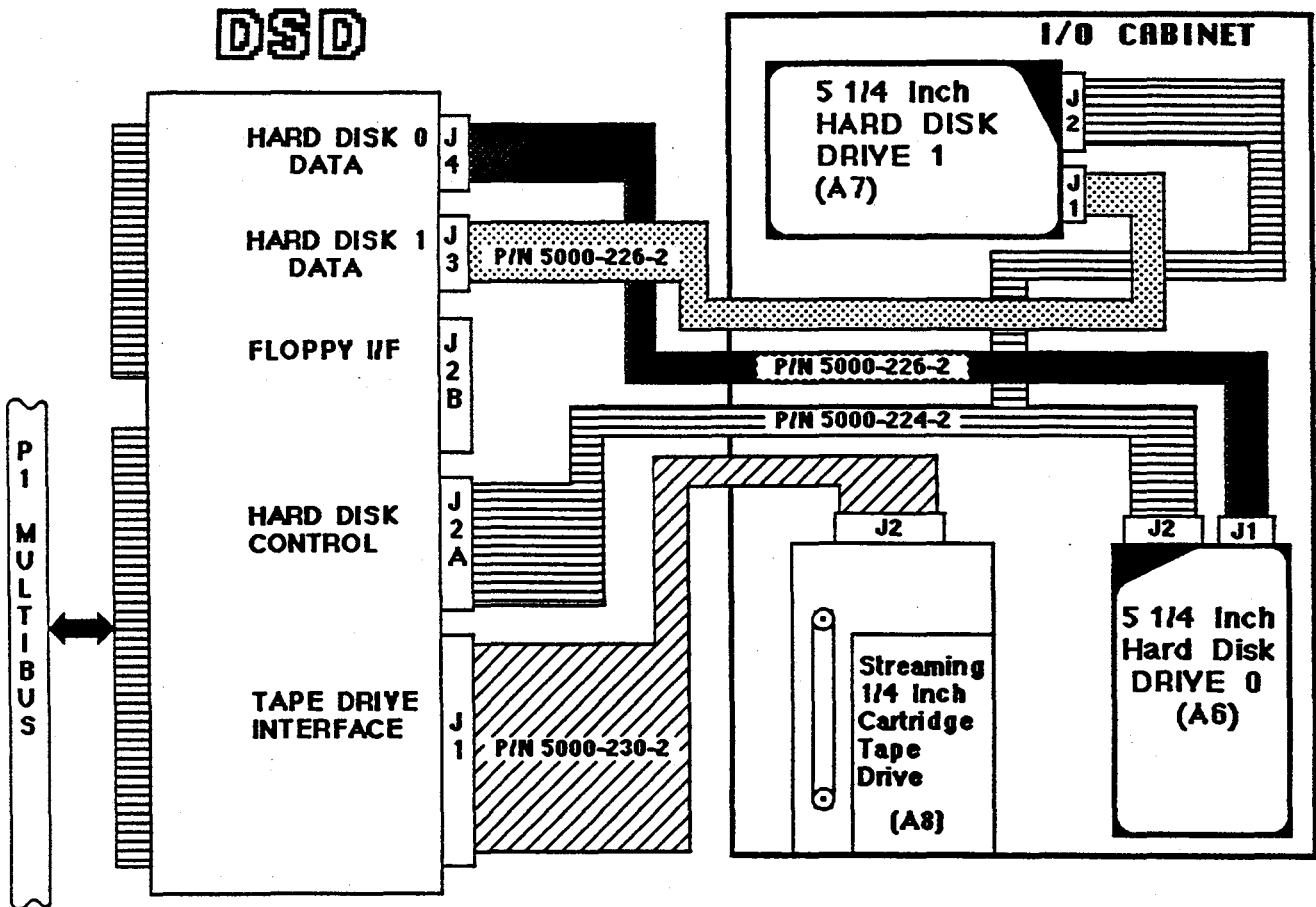
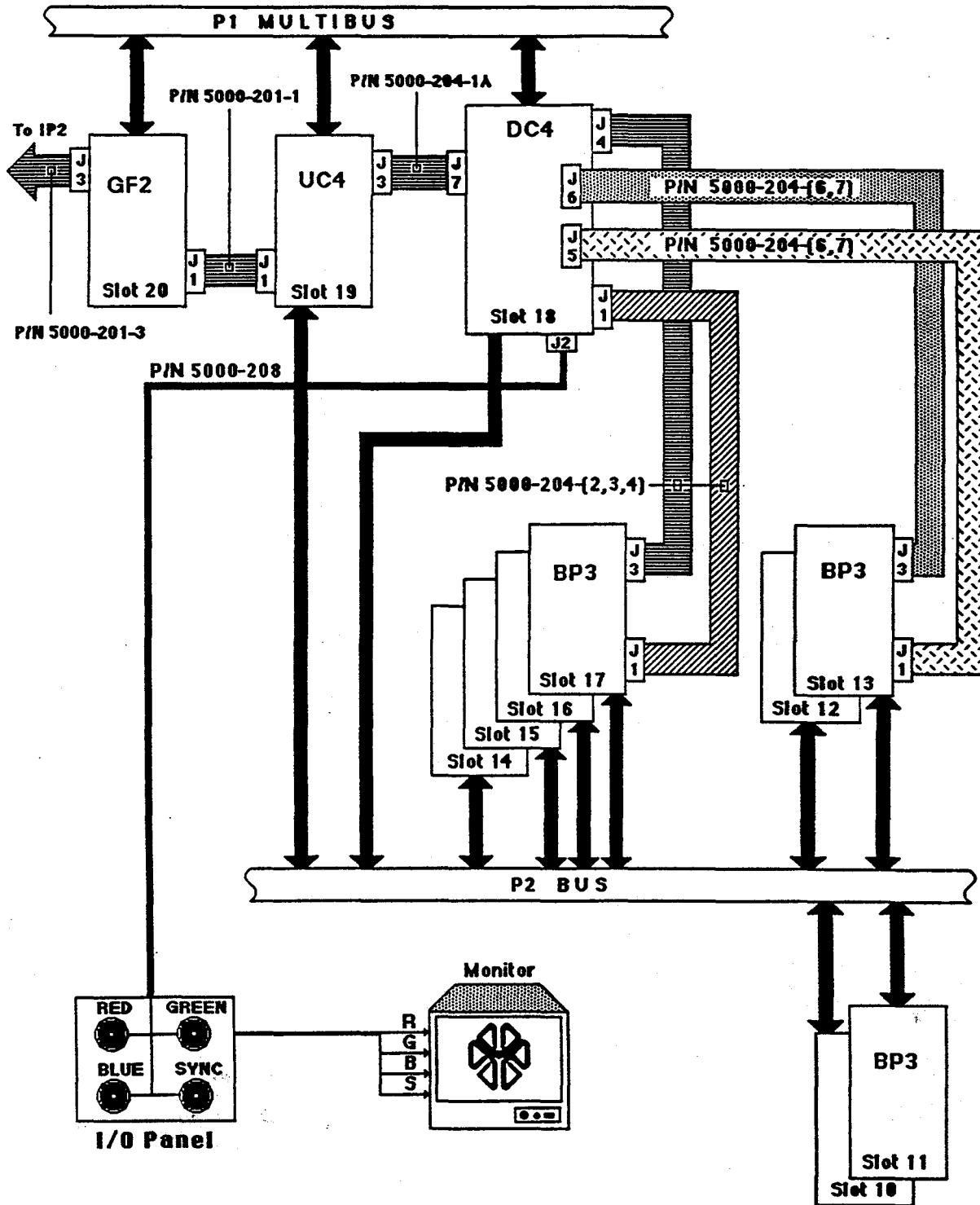


DIAGRAM 8: Graphics sub-system interconnect



**DIAGRAM 9: PCB switch settings - page 1**

**Memory Switches:** The switch pack is located on the front edge of each board. Each pack contains four switches.

		Switch			
		1	2	3	4
B o a r d	1	C	C	C	C
	2	C	C	C	O
	3	C	C	O	C
	4	C	C	O	O

"C" = Closed  
"O" = Open

**Bit Plane Switches:** The switch pack is located on the front edge of each board. There are four switches per pack, and the setting produces a hex address starting at 8.

		Switch			
Addr	Slot	1	2	3	4
8	17	O	C	C	C
9	16	O	C	C	O
A	15	O	C	O	C
B	14	O	C	O	O
C	13	O	O	C	C
D	12	O	O	C	O
E	11	O	O	O	C
F	10	O	O	O	O

8  
9  
10  
11  
12  
13  
14  
15

"C" = Closed  
"O" = Open *Open = 1*

**DIAGRAM 10: PCB switch settings - page 2**

**IP2 - Switch S2**

The IP2 has a 16 bit switch register. S1 on the processor board controls the low order bits and S2 controls the high order bits. Closing a switch gives 0 and leaving it open gives a 1. S1 is connected to the switches on the back panel. The individual switches are anded together, so if either of the switches is closed, the bit will be zero. Leaving all of the switches open on S1 will enable all of the backpanel switches.

The following table shows what S2 controls. Note: "C" means closed and "O" means open.

Category	Switch setting								Meaning
	1	2	3	4	5	6	7	8	
master/slave	C	X	X	X	X	X	X	X	Master processor
	O	X	X	X	X	X	X	X	Slave processor
RS232 Speed for ports 1-3	X	X	X	C	C	X	X	X	9600 baud
	X	X	X	C	O	X	X	X	300 baud
	X	X	X	O	C	X	X	X	1200 baud
	X	X	X	O	O	X	X	X	19.2K baud
Display types Primary/Secondary	X	X	X	X	X	C	C	C	NI/NI (DC4:5000-090-03)
	X	X	X	X	X	C	C	O	NI/I (DC4:5000-090-03)
	X	X	X	X	X	C	O	C	NI/RS (DC4:5000-090-02)
	X	X	X	X	X	C	O	O	NI/EU (DC4:5000-090-05)
	X	X	X	X	X	O	C	C	I/NI (DC4:5000-090-03)
	X	X	X	X	X	O	C	O	I/I (DC4:5000-090-03)
	X	X	X	X	X	O	O	C	I/RS (DC4:5000-090-04)
	X	X	X	X	X	O	O	O	I/EU (DC4:5000-090-06)

*all closed*

NI-non-interlaced 60 Hz  
 I-interlaced 30 Hz  
 RS-RS170A  
 EU-European

*a*

DIAGRAM 11: PCB switch settings - page 3

IP2 - Switch S1

The following table shows what S1 controls. Remember that S1 and the backpanel switches are wired together. It is recommended that all of the S1 switches be left open. This will give the backpanel switches complete control.

Category	Switch setting								Meaning
	1 8	2 7	3 6	4 5	5 4	6 3	7 2	8 1 (S1) (BACK)	
Boot Environment	X	X	X	X	C	C	C	C	(0) disk boot
	X	X	X	X	C	C	C	O	(1) tape boot
	X	X	X	X	C	C	O	C	(2) floppy boot
	X	X	X	X	C	C	O	O	(3) network boot
	X	X	X	X	C	O	C	C	(4) not used
	X	X	X	X	C	O	C	O	(5) PROM monitor (ie no device)
	X	X	X	X	C	O	O	C	(6) Diagnostic PROM board boot
	X	X	X	X	C	O	O	O	(7) not used
	X	X	X	X	O	C	C	C	(8) not used
	X	X	X	X	O	C	C	O	(9) ip - interphase boot
	X	X	X	X	O	C	O	C	(a) st - storager 2 tape boot
	X	X	X	X	O	C	O	O	(b) sf - storager 2 floppy boot
	X	X	X	X	O	O	C	C	(c) sd - storager 2 disk boot
	X	X	X	X	O	O	C	O	(d) mt - DSD tape boot
X	X	X	X	O	O	O	C	(e) mf - DSD floppy boot	
X	X	X	X	O	O	O	O	(f) md - DSD disk boot	
Autoboot	X	X	X	C	X	X	X	X	don't autoboot
	X	X	X	O	X	X	X	X	autoboot
Quiet	X	X	C	X	X	X	X	X	print hardware information
	X	X	O	X	X	X	X	X	don't print information
Display	X	C	X	X	X	X	X	X	Use primary monitor for display
	X	O	X	X	X	X	X	X	Use secondary monitor

The recommended switch settings are S1 all open, S2 all closed, and the backpanel switches all closed. This will configure the IP2 to be a master processor with a 60 Hz monitor that will boot from the disk.

**DIAGRAM 12: PCB switch settings - page 4**

**Disk Drives**

The following illustrations show you where to locate the address jumpers for the different types of disk drives found in SGI workstations. If you replace a disk drive, insure all the other jumpers on the new drive match the drive that you remove.

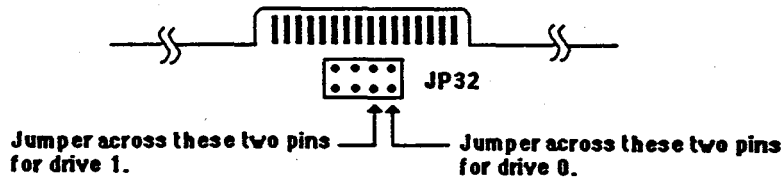
**Vertex 72 MB Drive**

Address selection is controlled by switch pack "J6" on the logic PCB located on the under side of the disk drive. J6 is located near and between the two edge pin connectors.

- For drive 0: Switch 4 is on, switch 7 is off, and switch 8 is on.
- For drive 1: Switch 4 is on, switch 7 is on, and switch 8 is off.

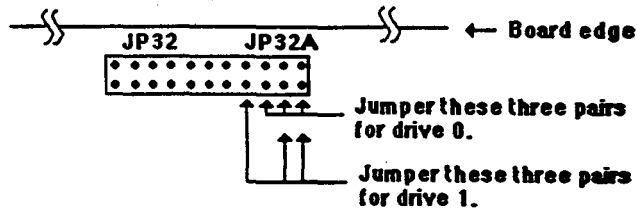
**Hitachi 85 MB Drive**

Address selection is done using jumpers across pins at jumper platform "JP32". This platform is located near the large pin edge connector. The following illustration shows how to set the jumpers.



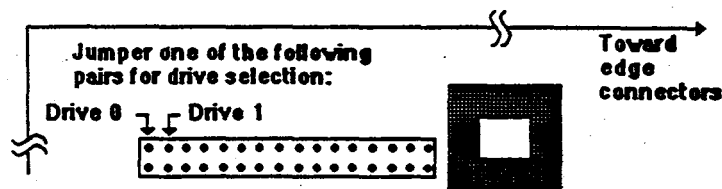
**Hitachi 170 MB Drive**

For this drive address selection is set using jumpers on jumper platform JP32A. It is located along the edge of the logic PCB next to JP32. The following illustration shows what pins are used.

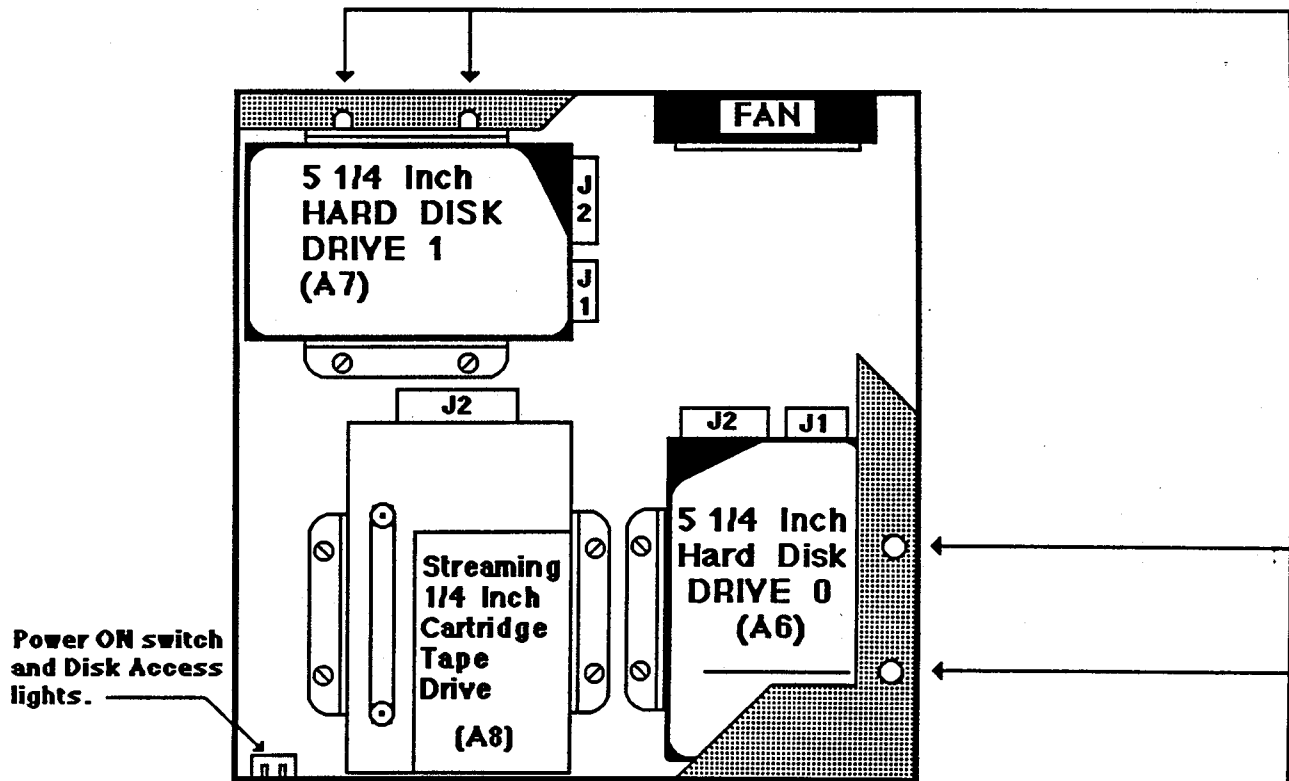


**Fujitsu 170 MB Drive**

For this drive address selection is set using jumpers on a jumper platform that has no name. It has 16 pairs of pins and is located at the end of the logic board that is away from the edge pin connectors. It is next to a large square chip. See following illustration for desired pin connections.



**DIAGRAM 13: I/O compartment components location**

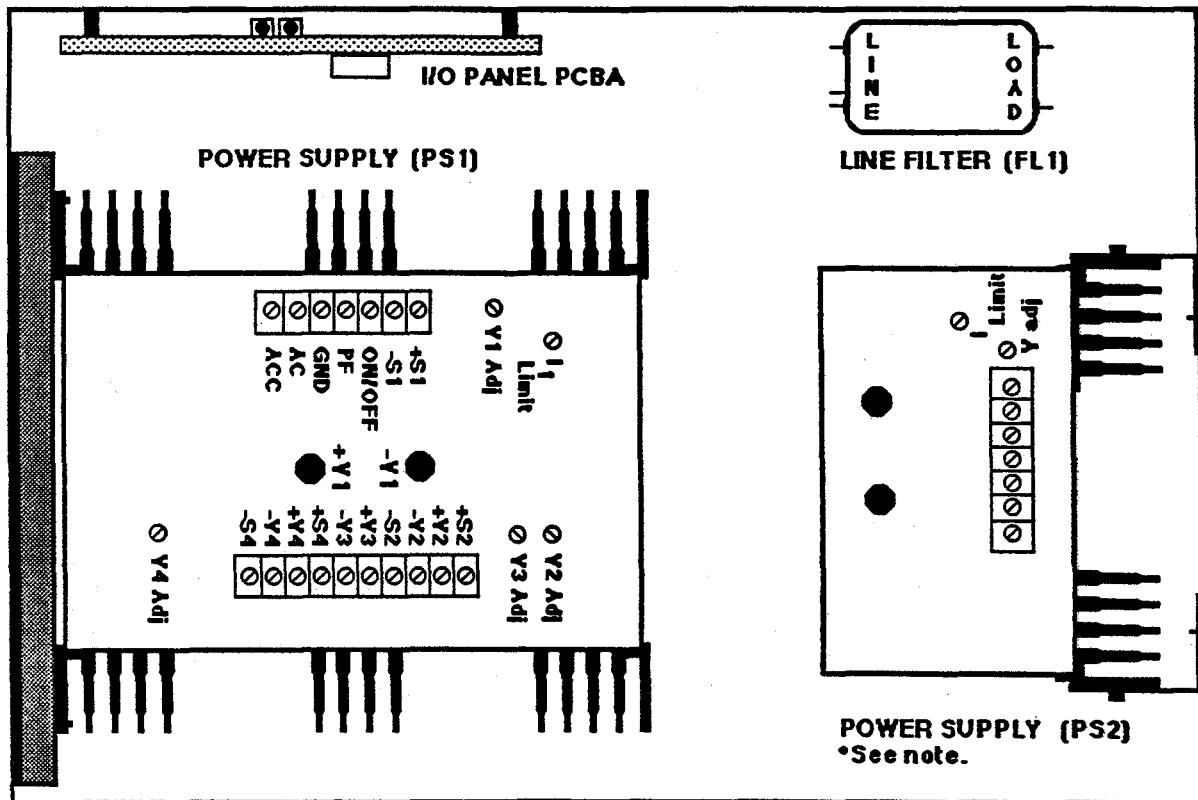


**NOTE:**

When removing the Hard Disk Drives, you will need a screw driver with a long neck that will allow you access to the two screws that are obstructed by the flange that is part of the sheet metal that comprises the workstation housing. Part of the flange showing access holes for the disk fastening screws is shown and highlighted by the drawing.



DIAGRAM 14: Power compartment components location



**NOTE:**

This power supply is added to 2400 workstations when they are upgraded to a 2400 Turbo. The existing supply (an LM-54) does not have enough current drive to supply all the options. The additional supply (an LM-11) *CAN ONLY* be paralleled with an LM-54. For installation instructions, see document # 003-0002.

DIAGRAM 15: 2400T AC power distribution

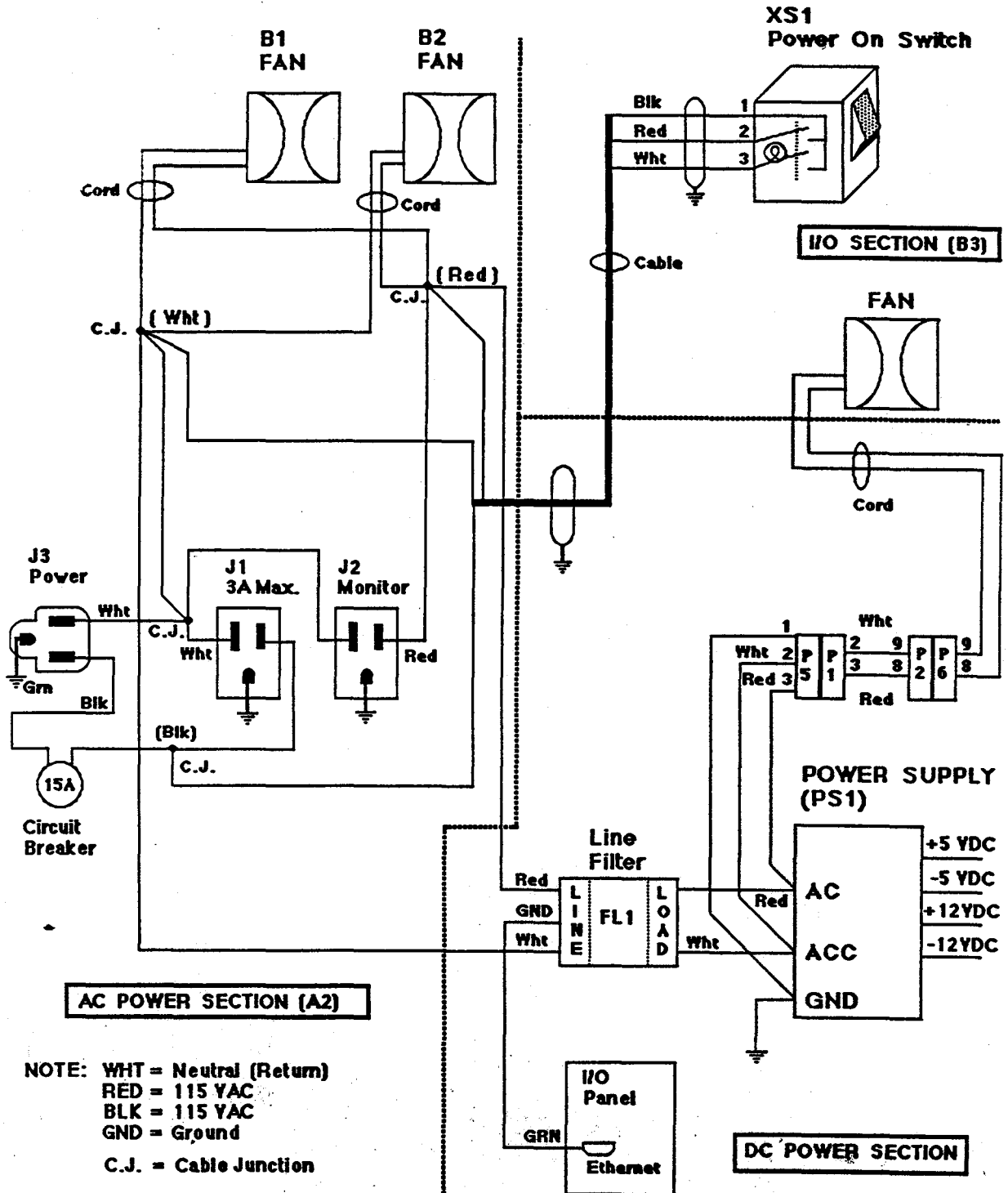


DIAGRAM 16: 2400T DC power distribution

Student Workbook

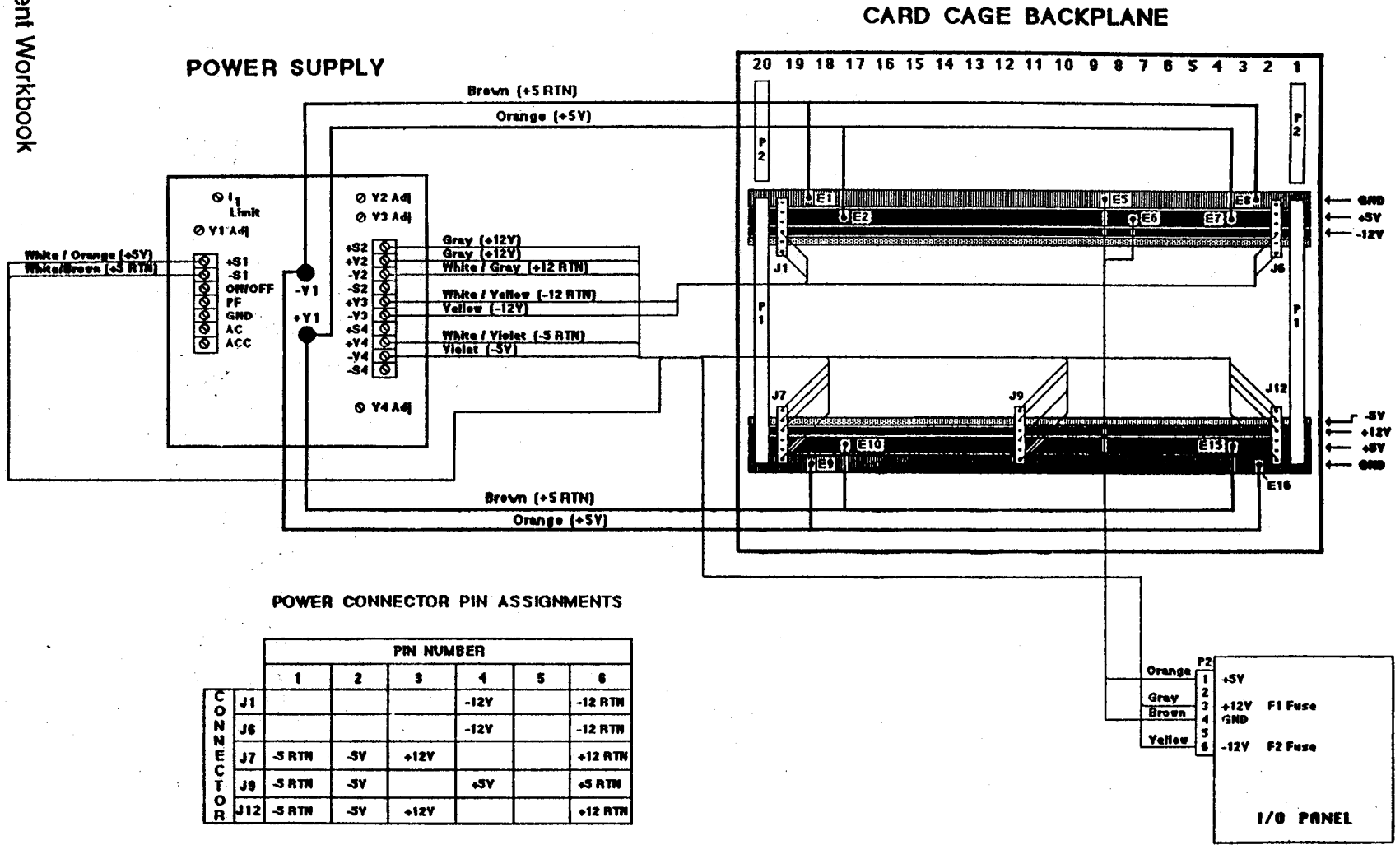
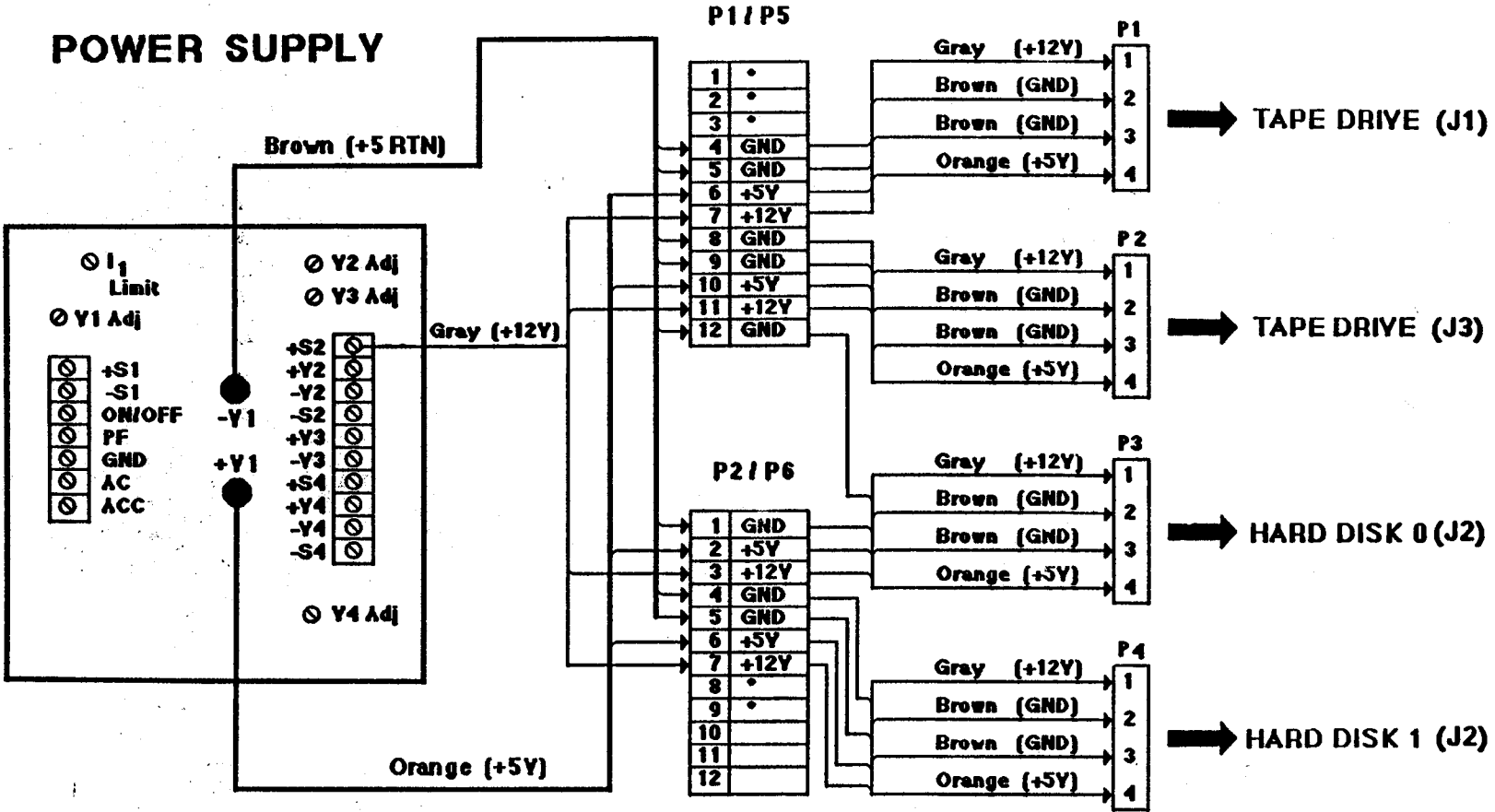


DIAGRAM 17: 2400T I/O power distribution



NOTE: (•) See AC Distribution Diagram for I/O Fan.

**SECTION 3: Review**

Student Name:

Please complete the review and hand it into your instructor.

1. Where are the RGB video and sync lines terminated?

on Display Back

2. If your workstation contains 8 bit planes, show the state of the address switches for the bit plane in card slot 13.

None 01 0000

3. What is the recommended state of all the switches in switch packs S1 and S2 located on the IP2 board?

S1 all open S2 all closed.

4. What slot will the Ethernet board occupy for a 12 MB system with a Floating Point board?

slot 5

5. Where does the cable from J1 of the Ethernet board connect to?

I/O Panel

6. J3 of the UC4 board connects to what?

J3 → J7 PC4

7. Where could you monitor the -12 voltage?

Power Supply

# Contents

<b>LESSON 6: DATA FLOW: Theory of Operation</b> .....	1
SYSTEM BLOCK DIAGRAM .....	2
PRODUCING GRAPHIC COLORS .....	3
ADDITIVE COLOR PRODUCING METHOD .....	4
MIXING COLORS .....	5
PIXEL .....	6
PIXEL ORGANIZATION .....	7
RASTER SCAN: 60 Hz MONITOR .....	8
RASTER SCAN: 30 Hz INTERLACED MONITOR .....	9
DISPLAY CONTROLLER (DC4): D-to-A CONVERTERS .....	10
A SINGLE BIT PLANE .....	11
BIT PLANE BOARD: BP3 .....	12
BIT PLANE DATA .....	13
COLOR MAP .....	14
FRAME BUFFER STRUCTURE: COLOR MAP MODE .....	15
BP3 PHYSICAL ORGANIZATION .....	16
COLOR MAP MODE: Single Buffer 2 Bit Planes .....	17
COLOR MAP MODE: Single Buffer 3 Bit Planes .....	18
COLOR MAP MODE: Double Buffer 2 Bit Planes .....	19
COLOR MAP MODE: Double Buffer 3 Bit Planes .....	20
COLOR MAP MODE: Double Buffer 4 Bit Planes .....	21
COLOR MAP MODE: Double Buffer 5 Bit Planes .....	22
COLOR MAP MODE: Double Buffer 6 Bit Planes .....	23
RGB MODE .....	24
Z-BUFFERING: 7 Bit Planes .....	25
Z-BUFFERING: 8 Bit Planes .....	26
BIT PLANE to DC4 CABLE CONNECTIONS .....	27
BIT PLANE MAPPING and CABLE CONNECTION TABLES .....	28
BIT PLANE BLOCK DIAGRAM .....	29
DISPLAY CONTROLLER .....	30
DISPLAY CONTROLLER: Basic Block Diagram .....	31
DISPLAY CONTROLLER: Detailed Block Diagram - page 1 .....	32
DISPLAY CONTROLLER: Detailed Block Diagram - page 2 .....	33
GRAPHICS SYSTEM DATA FLOW .....	34
IP2 PROCESSOR .....	35
GRAPHICS PROCESSOR: GF2 .....	36
GRAPHICS SYSTEM FLOW CHART - page 1 .....	37

GRAPHICS SYSTEM FLOW CHART - page 2 .....	38
FRAME BUFFER CONTROLLER .....	39
GRAPHICS SYSTEM FLOW CHART - page 3 .....	40
UPDATE CONTROLLER: UC4 .....	41
UPDATE CONTROLLER BLOCK DIAGRAM .....	42
NTSC VIDEO .....	43
Non-GENLOCKED NTSC .....	44
GENLOCKED NTSC .....	45
GENLOCKED NTSC OPERATION .....	46
GENLOCKED RS-170A OPERATION .....	47

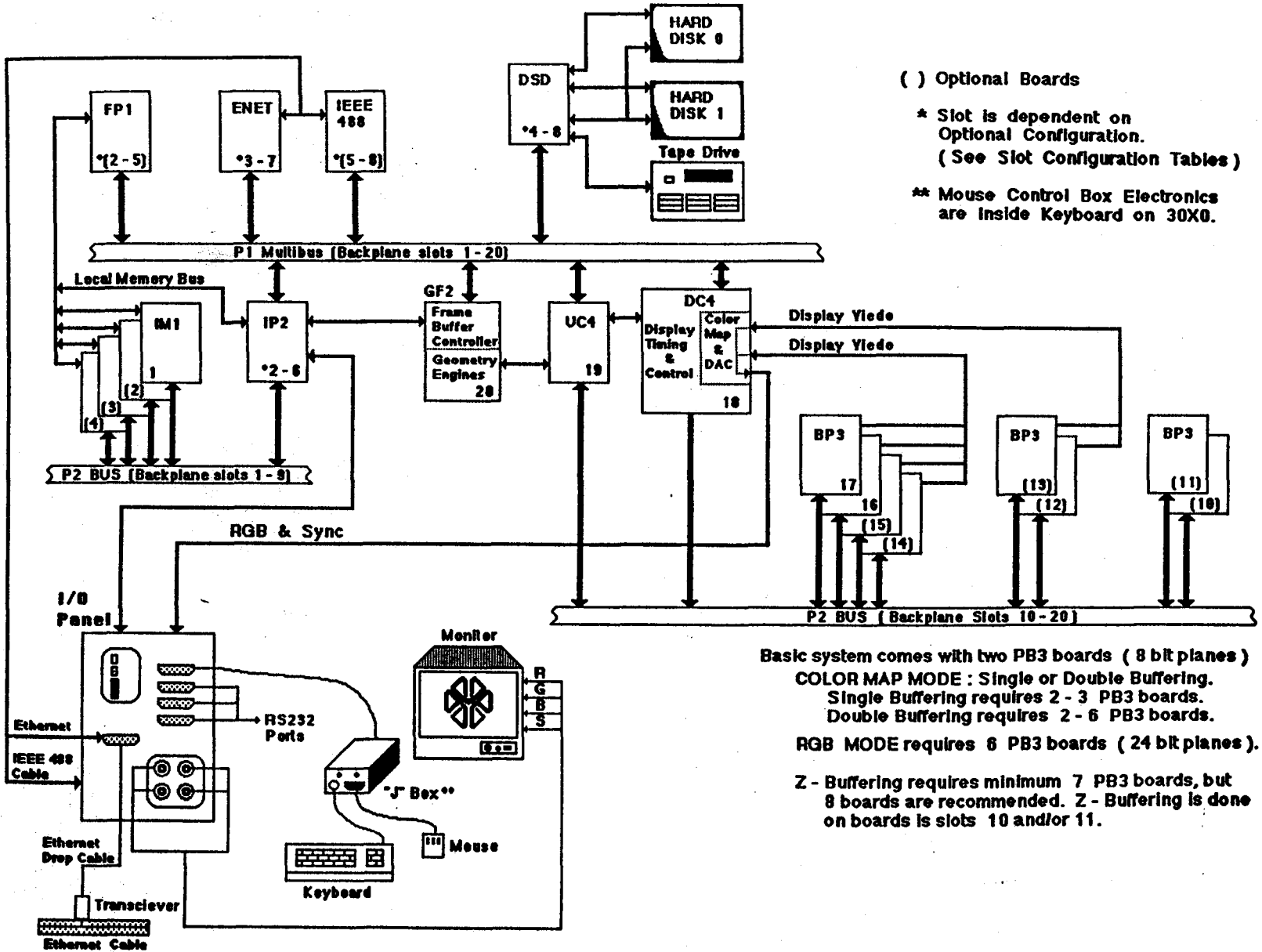
## **LESSON 6: DATA FLOW: Theory of Operation**

Upon completion of this lesson, the student, using available documentation will be able to:

- Name each one of the boards in the IRIS workstation; stating the purpose of each board and correctly indicate where that board fits into the overall block diagram.
- Be able to name the major subsystems of the IRIS workstation.



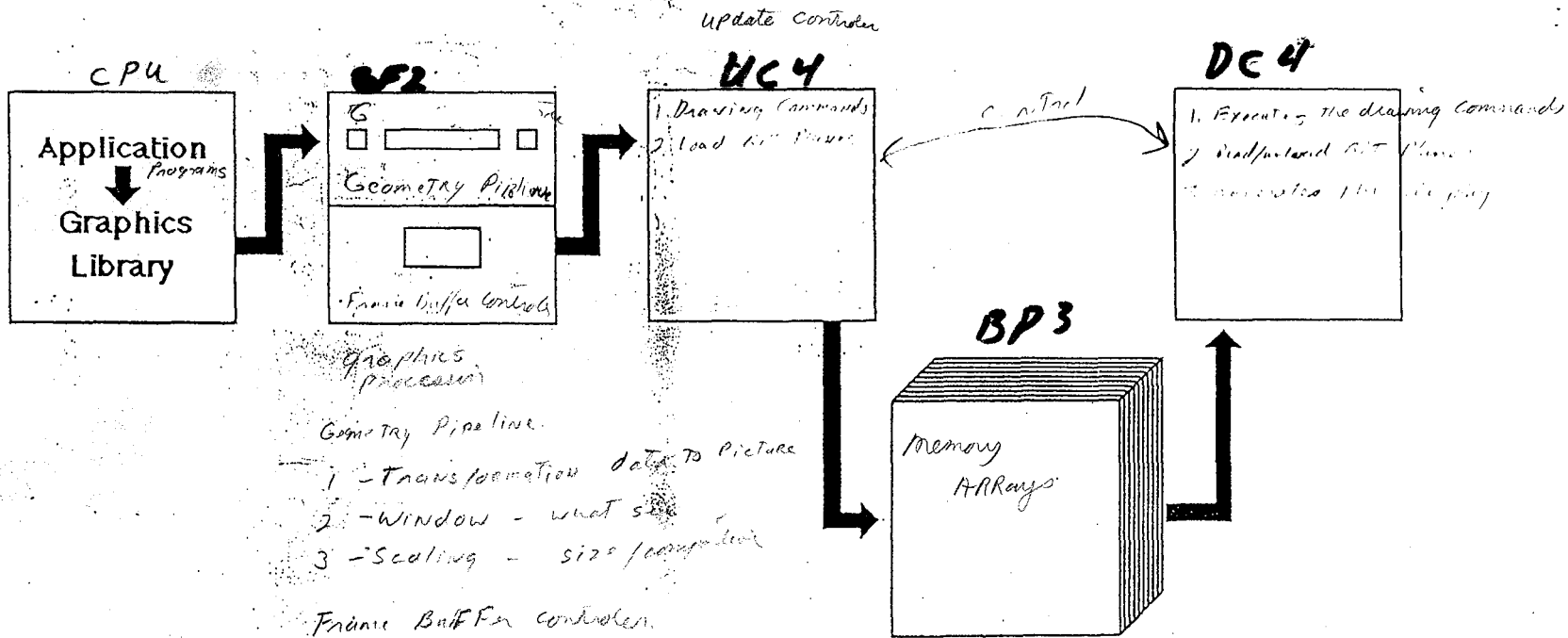
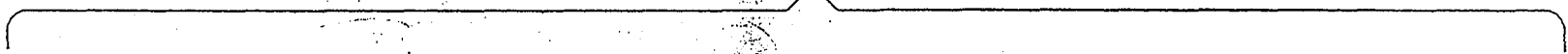
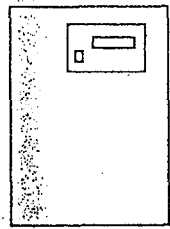
**SYSTEM BLOCK DIAGRAM**



( ) Optional Boards  
 \* Slot is dependent on Optional Configuration. ( See Slot Configuration Tables )  
 \*\* Mouse Control Box Electronics are inside Keyboard on 30X0.

Basic system comes with two PB3 boards ( 8 bit planes )  
**COLOR MAP MODE** : Single or Double Buffering.  
 Single Buffering requires 2 - 3 PB3 boards.  
 Double Buffering requires 2 - 6 PB3 boards.  
**RGB MODE** requires 6 PB3 boards ( 24 bit planes ).  
**Z - Buffering** requires minimum 7 PB3 boards, but 8 boards are recommended. Z - Buffering is done on boards in slots 10 and/or 11.

# Graphics Data Flow



Graphics processor

Geometry Pipeline

- 1 - Transformation data to picture
- 2 - Window - what see
- 3 - Scaling - size/compute

Frame Buffer controller

- 1 - Z Buffer 3 dimensioning
- 2 - Gouraud shading - The way light falls on objects and how it looks on solid objects
- 3 - depth Culling - on WIRE FRAME models
  - The closer to you the lighter color of wire and gets darker as it goes away from you

## PRODUCING GRAPHIC COLORS

### Primary Colors

The light of a few distinct colors, called *Primary Colors*, is mixed together in various proportions and amounts to form a desired color.

If the primary colors are chosen carefully, the potential to form many colors will exist.

The selection of primary colors is somewhat arbitrary, and there are several choices which will perform as desired, to some extent. The only requirement is that no primary can be matched by any combination of the other primaries.

### SGI Primary Colors

The universally accepted primary colors, when mixed together in equal amounts produce *white light*, are **Red, Green & Blue**. These colors are chosen because of their ability to produce a virtually complete array of colors.

## **ADDITIVE COLOR PRODUCING METHOD**

Graphically speaking, there are two ways to create different colors. In one method, the light of the primary colors is mixed together in various proportions and amounts to form the desired color. This method is called the *Additive Method*.

The second method used is referred to as the *Subtractive Method*. You start with white light and subtract out certain primaries to produce the desired color.

**SGI uses the Additive method.**

The idea of adding colors is shown by the figure on the next page. There are three circles in Red, Green and Blue which overlap. Where the circles are superimposed, the color shown is the mixture produced by adding equal amounts of the primary colors. At the center, all three color circles overlap, resulting in white.

Where only green and blue add, the resulting color is a greenish blue mixture called *Cyan*. The red purple color shown by the addition of red and blue is called *Magenta*, or violet. Where red and green overlap is called *Yellow*.

### **NOTE:**

Printers call cyan, magenta and yellow the primary colors.

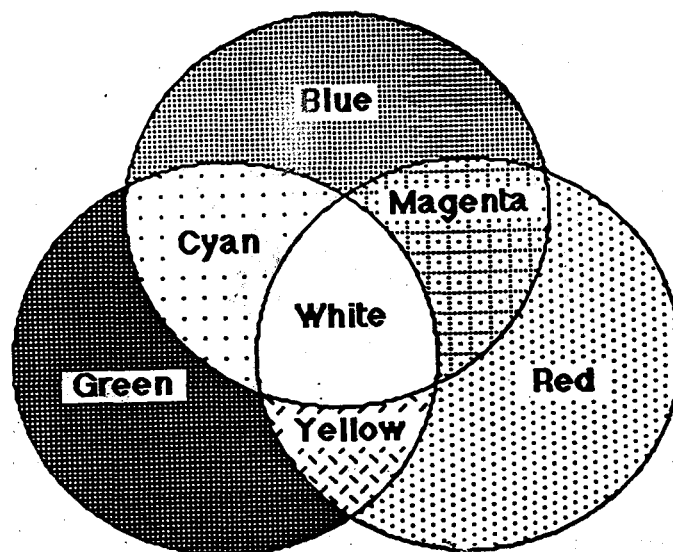
## MIXING COLORS

To produce any color, you must begin with the primary colors. The diagram below shows the three primary colors of red, green and blue producing what printers call the primary colors of cyan, magenta and yellow.

Now, this is only six colors, how are the thousands of other colors produced?

The primary colors of the chart are assumed to be at maximum intensity or *true* red, green, and blue.

By mixing the three primary colors together using different levels of intensity for each primary, thousands of colors can be produced. (i.e., maximum and equal amounts of red and blue produce magenta, but more blue with less red produces purple or, more red with less green produces orange and so on.



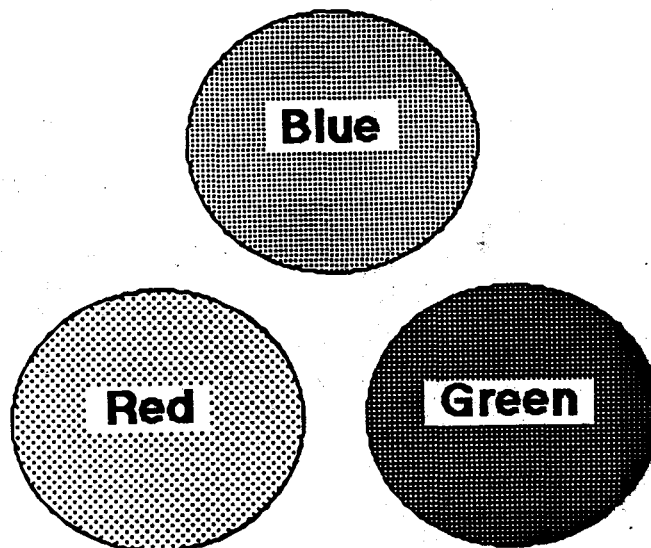
## PIXEL

Graphic displays (the CRT or monitor) are solid objectives and not cans of paint pigments that can be freely mixed. How then, are the colors we see at the monitor produced?

Monitor colors are produced by exciting *light producing* phosphor with an electronic beam. Actually, there are three beams and three colors of phosphor: Red, Green and Blue.

The inside face of the CRT is coated with tiny triplets of the colored phosphor; one of each color type to a triplet. Each triplet will become a point or *dot* on the screen. The dot is called a *PIXEL*. (see diagram below)

Just as any color of paint pigment can be made from mixing the three primary pigments together, so can any color be produced by the pixels when exciting them with varying intensity levels of the electronic beams - the stronger the beam, the more intense the individual phosphor will glow.



**PIXEL ORGANIZATION**

The pixels are arranged in lines on the face of the CRT.

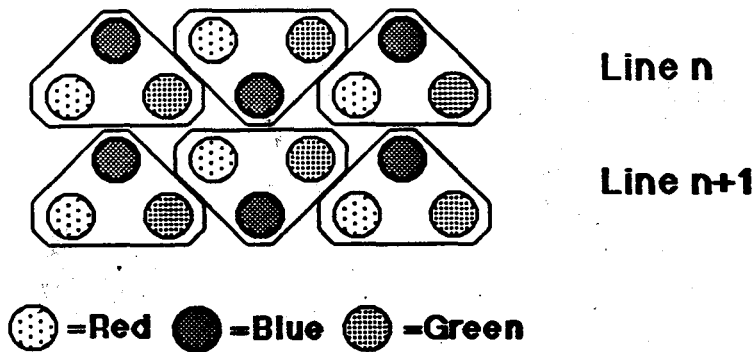
Monitor resolution (how sharp a picture will be) becomes a function of how closely the pixels are packed together on an individual line and how many lines there are on the CRT screen.

**SGI MONITOR RESOLUTIONS**

- 60 Hz non-interlaced ) *High resolution* 1024 pixels wide X 768 pixels high
- 33 Hz interlaced ) 1024 x 768
- Broadcast TV* / • 30 Hz interlaced *RS170A* 636 x 485
- 25 Hz interlaced *European* 768 x 575

The following diagram illustrates how the pixels are arranged.

**Pixel Organization**



### **RASTER SCAN: 60 Hz MONITOR**

The three electronic beams are driven together across the face of the CRT beginning at the upper left hand corner. The beams sweep the first line of pixels and when they reach the end of the line, they are returned to the beginning of the next line of pixels to again be swept across the face of the CRT.

The guns are aligned and the beams are controlled so that they sweep only the pixels on an individual line.

While the beams are sweeping over the individual pixels, beam intensity modulation will cause the pixels to glow desired colors. While the beams are being repositioned at the beginning of the next line (this is called horizontal retrace), video modulation of the beams is blanked (shut off), insuring that "picture painting" only occurs on a video sweep (beam moving left to right).

When the beams reach the end of the last line of pixels, they are again blanked and retraced to the beginning of the first line (vertical retrace). The process is then repeated.

From the beginning of the first line to the end of the last line is called a *Picture Frame*.

A 60 Hz monitor will draw **60 frames** per second.



**RASTER SCAN: 30 Hz INTERLACED MONITOR**

There are two fields, or sets of pixel lines, which are interlaced. The beams begin at the upper left corner and sweep out all the even-numbered lines, or the *even field*, and then the odd-numbered lines, or the *odd field* is swept.

Each field takes  $1/60$  of a second to complete (60 Hz), therefore it takes  $1/30$  of a second (30 Hz) to complete the entire picture. A complete picture is called a *frame*.

**DISPLAY CONTROLLER (DC4): DIGITAL-to-ANALOG CONVERTERS**

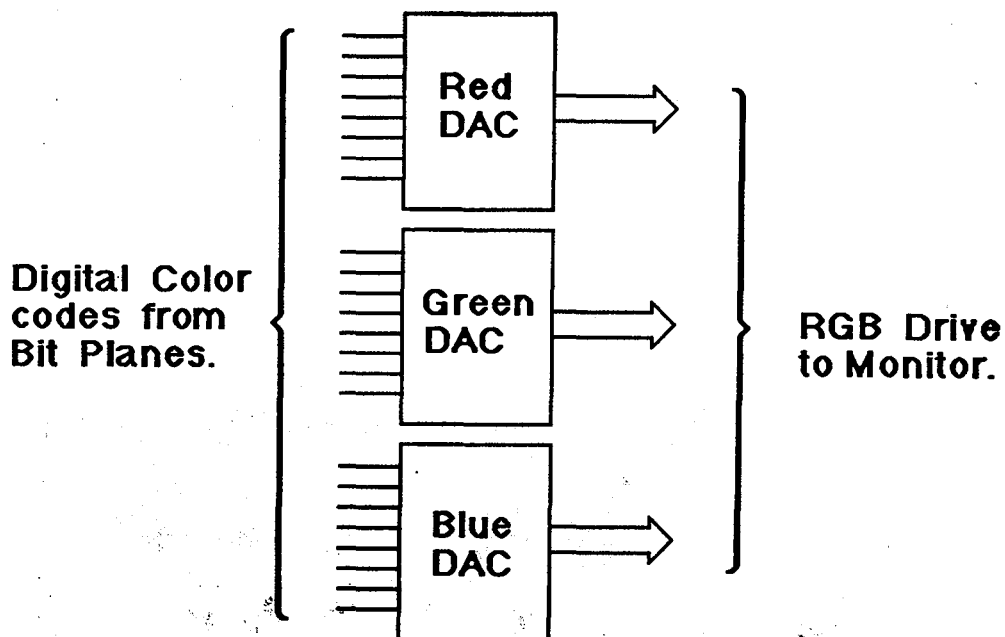
SGI utilizes an 8-bit *Digital-to-Analog Converter* (DAC) to drive the monitor beam guns.

By using 8-bit DAC's, 256 levels (intensities or shades) for each primary color can be produced at the monitor.

**16,777,216 colors** are possible using three 8-bit DAC's.

$$256 \times 256 = 65,536 \times 256 = 16,777,216$$

The DAC's, located on the *Display Controller (DC4)* receive the 24 bits of digital coded color data from the Bit Plane Boards (BP3).



**A SINGLE BIT PLANE****QUESTION:**

What is a *Bit Plane*?

**Answer:**

It is a Random Access Memory (RAM). *64K Bg 16*

*1024 / 1024 / 1*

**Question:**

How large is a single BP3 Bit Plane?

**Answer:**

Each IRIS bit plane is **1024 x 1024 x 1**.

*Bit of info for every pixel*

**Question:**

What is the bit plane used for?

**Answer:**

Storage of a single bit of digital coded color data for each one of the pixels on the face of the CRT.

- The bit stored for each pixel carries the same relative bit weight within the 8-bit color code, in other words - the bits in the plane represent one of the bit weights (1, 2, 4, 8, 16, 32, 64 or 128) of the 8-bit color code.
- Only **1024 x 768** locations are used of the possible **1024 x 1024**.

*Need 24 BIT To make Red Blue Green*

**Question:**

Why are only **1024 x 768** locations used?

*768 21*

**Answer:**

The maximum monitor resolution is **1024 x 768**.

- The bit plane is an exact copy of the pixel arrangement on the face of the CRT.
  - **1024 pixels wide.**
  - **1024 pixels high (lines), of which, only 768 are used.**

**BIT PLANE BOARD: BP3**

**Question:**

A Bit Plane board (BP3) is comprised of how many bit planes?

**Answer:**

4 bit planes.

**Question:**

Are the bit planes identical?

**Answer:** Yes.

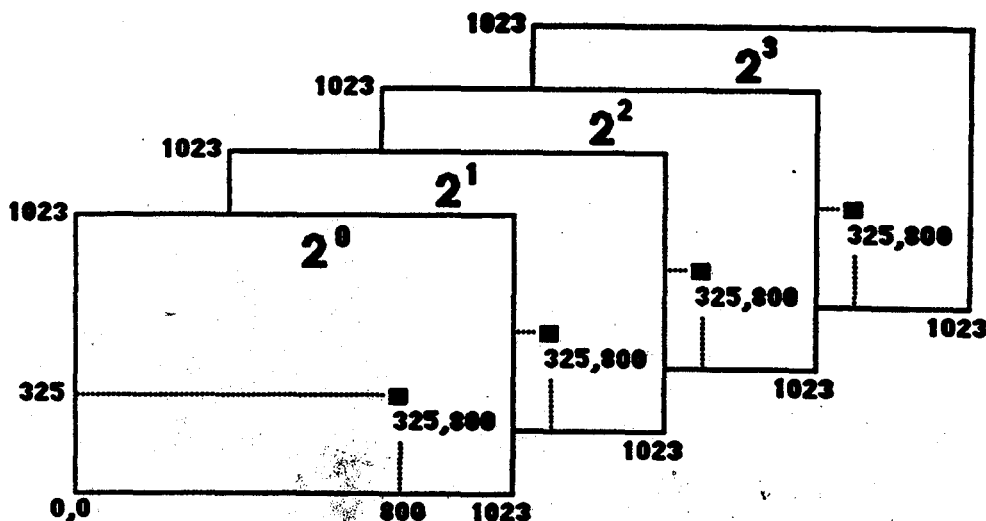
- The same address lines drive all four bit planes on BP3, and any other BP3 boards that may be installed.
- The same relative location in each bit plane stores a single bit of the 8-bit color code for the same relative pixel on the CRT screen.

**Question:**

Are the bit weights (for the color code) of each plane the same?

**Answer:**

NO - Each bit plane will store a different bit weight of the color code. (see diagram)



**NOTE: The system numbers the pixels from the lower left corner,**

**BIT PLANE DATA****Question:**

Do the bit planes store more than one type of data?

**Answer:**

YES - Three types of data can be stored in the bit planes depending on the programming mode: **RGB MODE**, **COLOR MAP MODE**, **Z-BUFFER MODE**.

- If in *RGB MODE*, the bit planes hold digital color codes for each of the primary colors.

RGB Mode requires a minimum of 6 BP3 boards.

- Each color is defined by an 8-bit code. *Need*
- $3 \times 8 = 24$  bits.
- Each BP3 will hold 4 bits, therefore,  
 $24 \text{ divide by } 4 = 6 \text{ BP3's (more later)}$

- If in *COLOR MAP MODE*, the bit planes will hold an address (index) into another RAM, called the *COLOR MAP*. (more later) *Faster + Run with less hardware → 1317 HW added*
- If in *Z-BUFFER MODE*, some of the bit planes will hold Z-coordinate (depth information) values for *Hidden Surface Removal* functions. (more later)

While in Z-Buffer Mode, 12 Bits planes will also hold Color Map Mode data.

## COLOR MAP

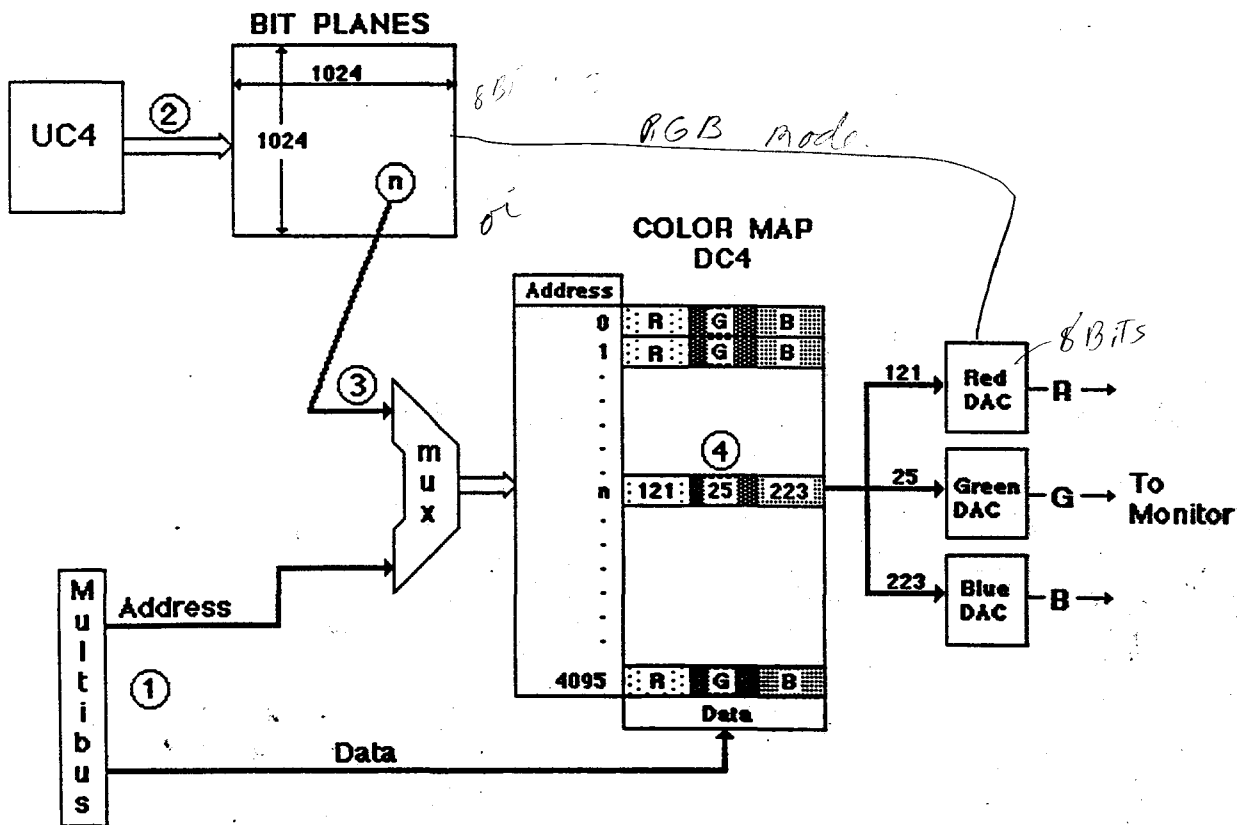
- The *Color Map* is located on the *Display Controller* (DC4).
- It is a 4096 x 24 bit RAM.
  - Addressed with 12 bits which produces 4096 locations.
  - The 24 data bits store three 8-bit color codes - one for each primary.
  - At any point in time, the Color Map can define (map) 4096 colors out of the possible 16.7 million.
- The color map is addressed from two places: **Bit Planes & Multibus**.
  - **Bit Planes**  
When in *Color Map Mode*, the bit planes supply the **12-bit address**, which is an index into a specific location. The three 8-bit (24 bits) color codes are output to the *DAC's* - This is a table-look-up function.
  - **Multibus**  
When the programmer wants to change the color codes in the color map, the system will load (map some new colors) the color map with new color codes.

See next page for Block Diagram of the Color Map.

**FRAME BUFFER STRUCTURE: COLOR MAP MODE**

The diagram below has four (4) numbers that are circled. Each number corresponds to one of the following paragraphs.

1. The color map is loaded with the digital values of the desired colors.
2. The Update Controller (UC4) loads pixel data into the Bit Planes.
3. The Display Controller (DC4) accesses the Bit Planes, at a rate that is synchronized to the monitor raster scanning, fetching pixel data sequentially - the data becoming index addresses into the color map. Point three freezes the scan at a point (a location) called *n* for illustration.
4. *N* addresses location *n* in the color map, from which 24 bits of color code are fetched. The three color codes are feed to the DAC's, and pixel *n* on the screen is painted the defined color.

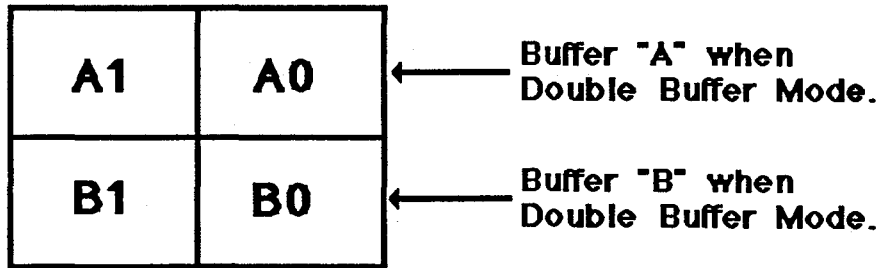


### BP3 PHYSICAL ORGANIZATION

The Bit Plane boards are comprised of four (4) identical bit planes. The planes are given names and the board is divided into two halves: A & B.

The names are *generic* and the actual configuration of the BP3 boards is illustrated by several of the diagrams that follow this diagram.

### BP3 Generic Names



Remember, the bit planes hold an address index when in *Color Map Mode*. The generic address bit weights are as follows:

- **Single Buffer Mode**

BP3 bit	B1	A1	B0	A0	
Bit weight	8	4	2	1	= 16

- **Double Buffer Mode**

BP3	A1/B1	A0/B0	
Bit weight	2	1	= 4



**COLOR MAP MODE: Single Buffer 2 Bit Planes**

**2 Boards (8-bit deep pixel)**

**BP3<sub>1</sub>**

Slot 16

A3	A2
B3	B2

**BP3<sub>0</sub>**

Slot 17

A1	A0
B1	B0

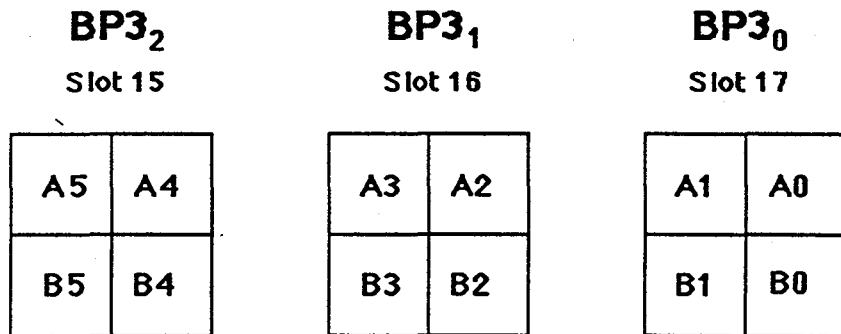
DC4 Color Map Address Bit	7	6	5	4	3	2	1	0
BP3 Bits	B3	A3	B2	A2	B1	A1	B0	A0
Color Map Address Weight	128	64	32	16	8	4	2	1

Using an 8-bit deep pixel,  
256 colors could be mapped.

Address bits 8-11 are set = 0.

**COLOR MAP MODE: Single Buffer 3 Bit Planes**

**3 Boards (12-bit deep pixel)**

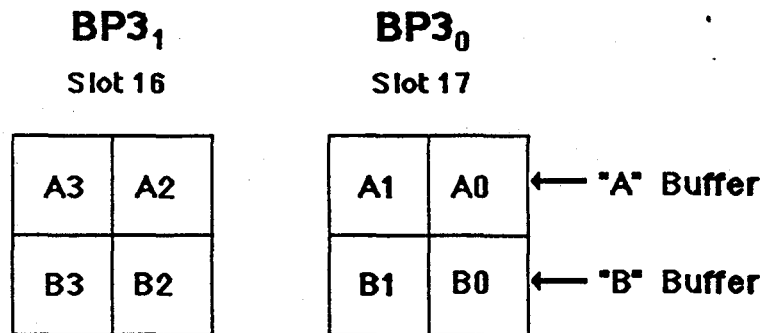


<b>DC4 Color Map Address Bit</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>BP3 Bits</b>	<b>B5</b>	<b>A5</b>	<b>B4</b>	<b>A4</b>	<b>B3</b>	<b>A3</b>	<b>B2</b>	<b>A2</b>	<b>B1</b>	<b>A1</b>	<b>B0</b>	<b>A0</b>
<b>Color Map Address Weight</b>	<b>2048</b>	<b>1024</b>	<b>512</b>	<b>256</b>	<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>

Using a 12-bit deep pixel, 4096 colors could be mapped.

**COLOR MAP MODE: Double Buffer 2 Bit Planes**

**2 Boards (4-bit deep pixel)**



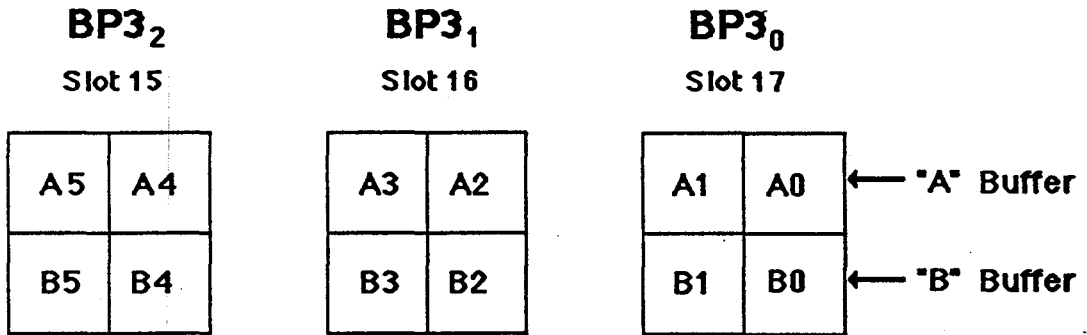
DC4 Color Map Address Bit	3	2	1	0
BP3 Bits ( A Buffer )	A3	A2	A1	A0
BP3 Bits ( B Buffer )	B3	B2	B1	B0
Color Map Address Weight	8	4	2	1

Using a 4-bit deep pixel in each buffer,  
16 colors could be mapped.

Address bits 4-11 are set = 0.

**COLOR MAP MODE: Double Buffer 3 Bit Planes**

**3 Boards (6-bit deep pixel)**



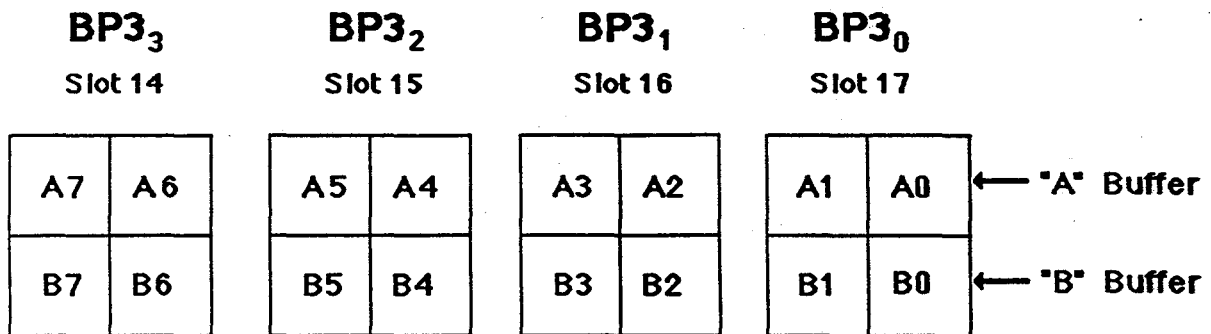
DC4 Color Map Address Bit	5	4	3	2	1	0
BP3 Bits ( A Buffer )	A5	A4	A3	A2	A1	A0
BP3 Bits ( B Buffer )	B5	B4	B3	B2	B1	B0
Color Map Address Weight	32	16	8	4	2	1

Using a 6-bit deep pixel in each buffer,  
64 colors could be mapped.

Address bits 6-11 are set = 0

**COLOR MAP MODE: Double Buffer 4 Bit Planes**

**4 Boards (8-bit deep pixel)**



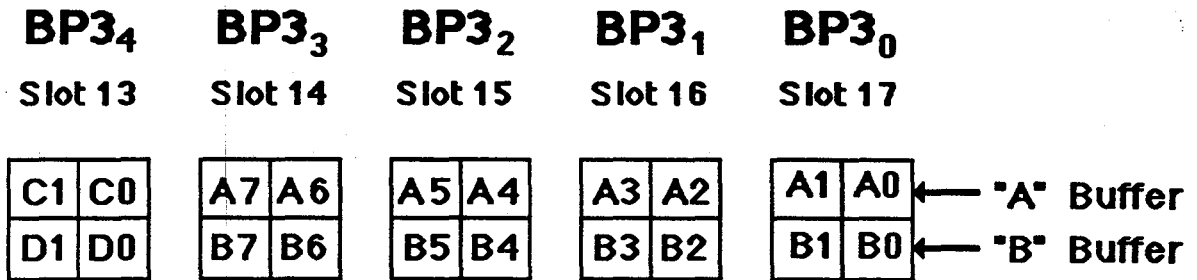
DC4 Color Map Address Bit	7	6	5	4	3	2	1	0
BP3 Bits ( A Buffer )	A7	A6	A5	A4	A3	A2	A1	A0
BP3 Bits ( B Buffer )	B7	B6	B5	B4	B3	B2	B1	B0
Color Map Address Weight	128	64	32	16	8	4	2	1

Using an 8-bit deep pixel in each buffer,  
256 colors could be mapped.

Address bits 8-11 are set = 0

**COLOR MAP MODE: Double Buffer 5 Bit Planes**

**5 Boards (10-bit deep pixel)**



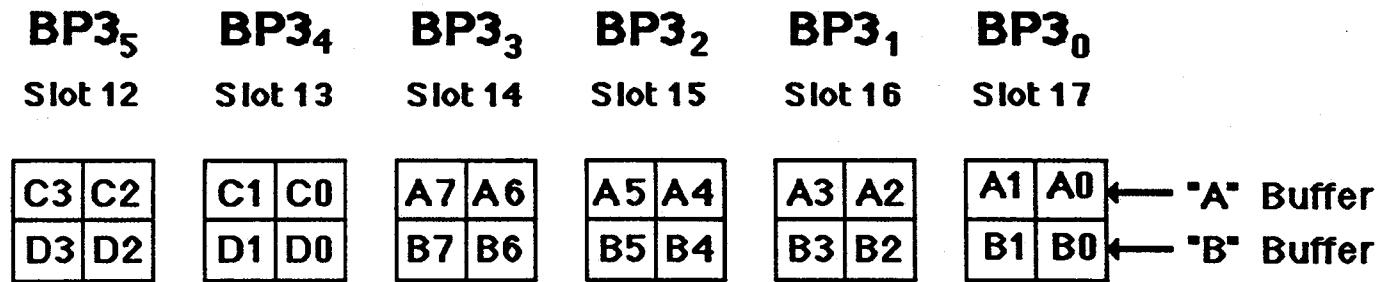
<b>DC4 Color Map Address Bit</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>BP3 Bits ( A Buffer )</b>	<b>C1</b>	<b>C0</b>	<b>A7</b>	<b>A6</b>	<b>A5</b>	<b>A4</b>	<b>A3</b>	<b>A2</b>	<b>A1</b>	<b>A0</b>
<b>BP3 Bits ( B Buffer )</b>	<b>D1</b>	<b>D0</b>	<b>B7</b>	<b>B6</b>	<b>B5</b>	<b>B4</b>	<b>B3</b>	<b>B2</b>	<b>B1</b>	<b>B0</b>
<b>Color Map Address Weight</b>	<b>512</b>	<b>256</b>	<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>

**Using a 10-bit deep pixel in each buffer,  
1024 colors could be mapped.**

**Address bits 10-11 are set to zeros.**

**COLOR MAP MODE: Double Buffer 6 Bit Planes**

**6 Boards (12-bit deep pixel)**



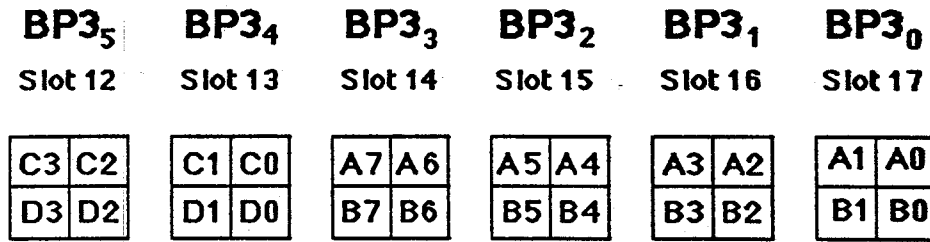
<b>DC4 Color Map Address Bit</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>BP3 Bits ( A Buffer )</b>	<b>C3</b>	<b>C2</b>	<b>C1</b>	<b>C0</b>	<b>A7</b>	<b>A6</b>	<b>A5</b>	<b>A4</b>	<b>A3</b>	<b>A2</b>	<b>A1</b>	<b>A0</b>
<b>BP3 Bits ( B Buffer )</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>	<b>B7</b>	<b>B6</b>	<b>B5</b>	<b>B4</b>	<b>B3</b>	<b>B2</b>	<b>B1</b>	<b>B0</b>
<b>Color Map Address Weight</b>	<b>2048</b>	<b>1024</b>	<b>512</b>	<b>256</b>	<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>

**Using a 12-bit deep pixel in each buffer,  
4096 colors could be mapped.**

**RGB MODE**

**RGB MODE**

Must have 6 boards (24 bits)



Bit Plane bits to DAC input bits								
	7	6	5	4	3	2	1	0
Red	A7	A6	A5	A4	A3	A2	A1	A0
Green	B7	B6	B5	B4	B3	B2	B1	B0
Blue	D3	C3	D2	C2	D1	C1	D0	C0
	128	64	32	16	8	4	2	1
	Color Weight							

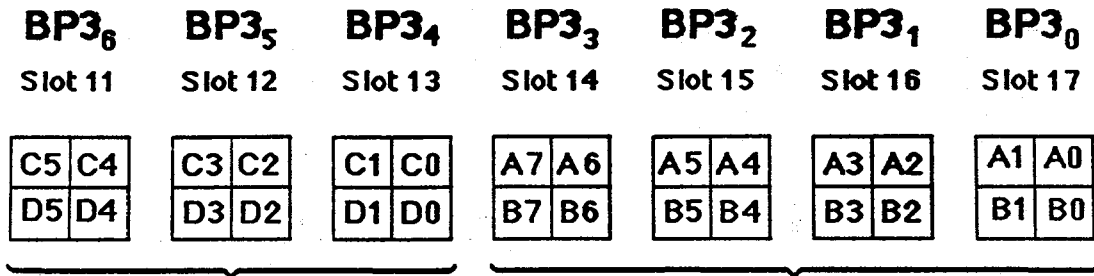
- Red = 256 shades (intensity)
- Green = 256 shades (intensity)
- Blue = 256 shades (intensity)

$$256 \times 256 = 65,536 \times 256 = 16,777,216$$

**In RGB mode, 16,777,216 colors are possible.**



**Z-BUFFERING: 7 Bit Planes**



**Z - Buffer Data**

- 12 planes

**Pixel Color Data**

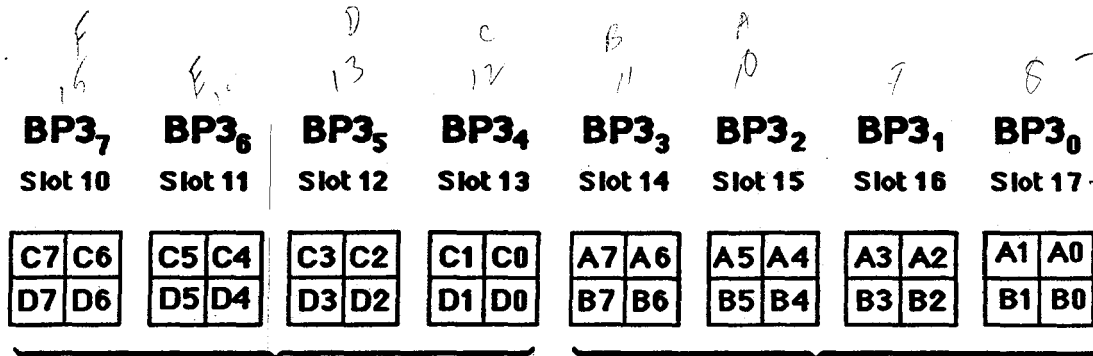
- Single buffer mode only
- 12 bit-planes store pixel color code to produce 4096 possible mapped colors.
- 4 bit planes un-used.

- Z-Buffer data is not used to drive the DAC's to produce video. It is used for *Hidden Surface Removal*.

Hidden surface removal is used when two or more objects occupy the same pixel. The Z coordinate for each pixel on the screen is stored in the bitplanes. When a pixel is to be drawn, it's new Z value is compared to the existing Z value. If the new Z value is less than the existing Z value (i.e., closer to the viewer), the new color value for that pixel is written into the bitplanes along with the new Z value. Otherwise, the color and Z value remain unchanged. The result is only the parts of the image visible to the viewer are displayed on the screen.

- Z-Buffering could be accomplished by programming if the customer does not purchase the optional bit planes, but it would take more time to perform the operation.

**Z-BUFFERING: 8 Bit Planes**



**Z - Buffer Data**  
• 16 planes

**Pixel Color Data**

- Single buffer mode only
- 12 bit-planes store pixel color code to produce 4096 possible mapped colors.
- 4 bit planes un-used.

*IF you change address switch 8 to 9 + problem stay the same it is D then u4 if it change say A to A3 then its BP3 Board.*

*Pop up menu + window IDs*

**QUESTION:**

Why are 4 bit planes lost?

**ANSWER:**

Because of the method used to write data into the bit planes. The bit planes are accessed on a word boundary - a word is comprised of 16 bits.

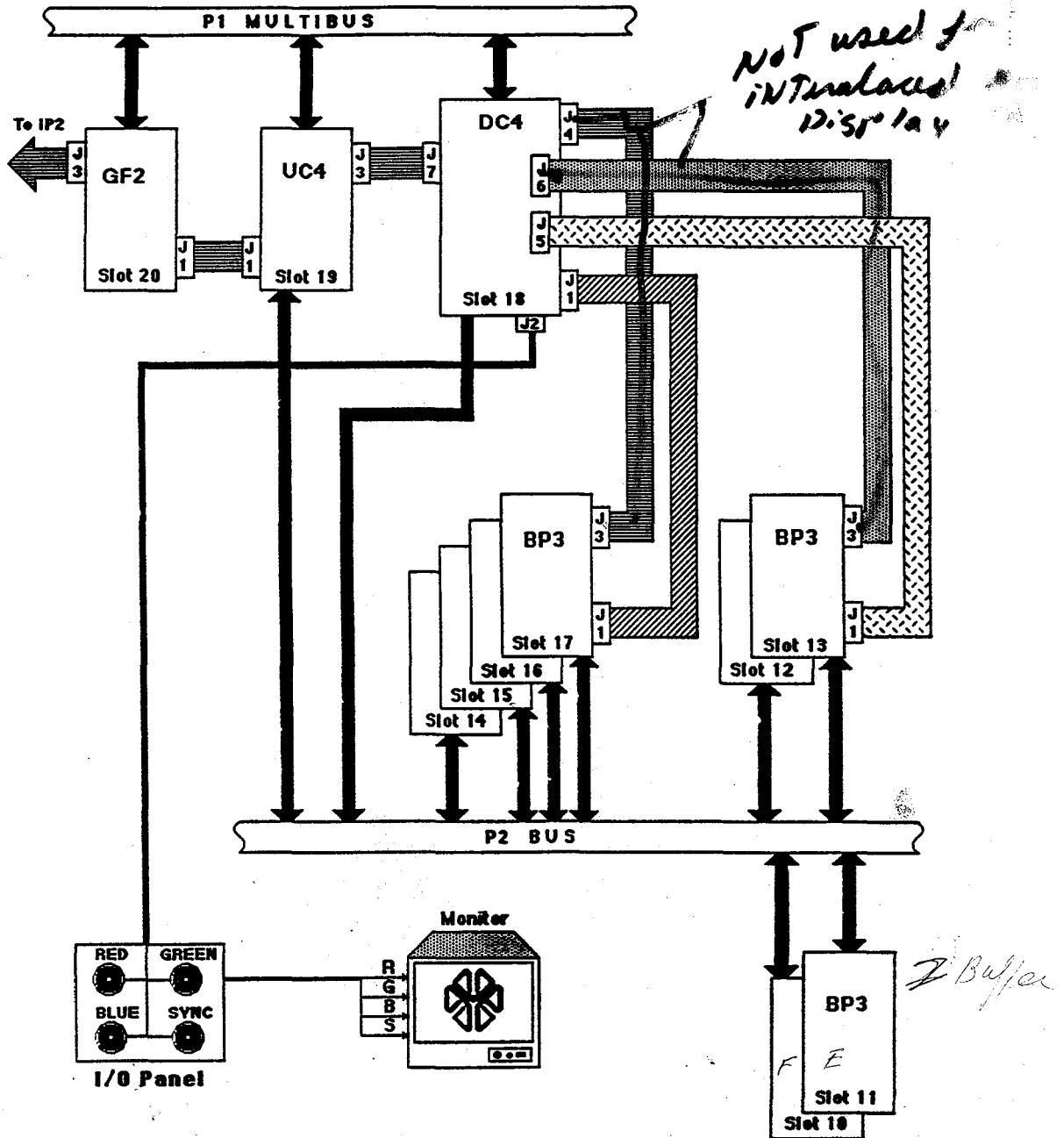
**Bit Plane Write Access**

- Single Buffer Mode: Write to A & B.
- Double Buffer Mode: Write to A & C or B & D.
- Z-Buffer Mode: Write to A & B for color.  
C & D for depth (Z-coordinate).

Since 12 bits is the maximum color pixel depth, 4 bit planes in the A & B double-word are un-used.

**BIT PLANE to DC4 CABLE CONNECTIONS**

This diagram and the next illustrate how the Bit Planes are cabled to the Display Controller (DC4).



**BIT PLANE MAPPING and CABLE CONNECTION TABLES**

**Mapping System Bitplane Names to BP3's**

Bitplane		Bitplane		Bitplane Names for each BP3			
number	name	number	name	Slot Number of BP3's L R			
				17/13	16/12	15/11	14/10
0	A0	10	C0	A0			
1	A1	11	C1	A1			
2	A2	12	C2		A0		
3	A3	13	C3		A1		
4	A4	14	C4			A0	
5	A5	15	C5			A1	
6	A6	16	C6				A0
7	A7	17	C7				A1
8	B0	18	D0	B0			
9	B1	19	D1	B1			
A	B2	1A	D2		B0		
B	B3	1B	D3		B1		
C	B4	1C	D4			B0	
D	B5	1D	D5			B1	
E	B6	1E	D6				B0
F	B7	1F	D7				B1

*Left column  
Left #*

**BP3 to DC4 Cable Connections**

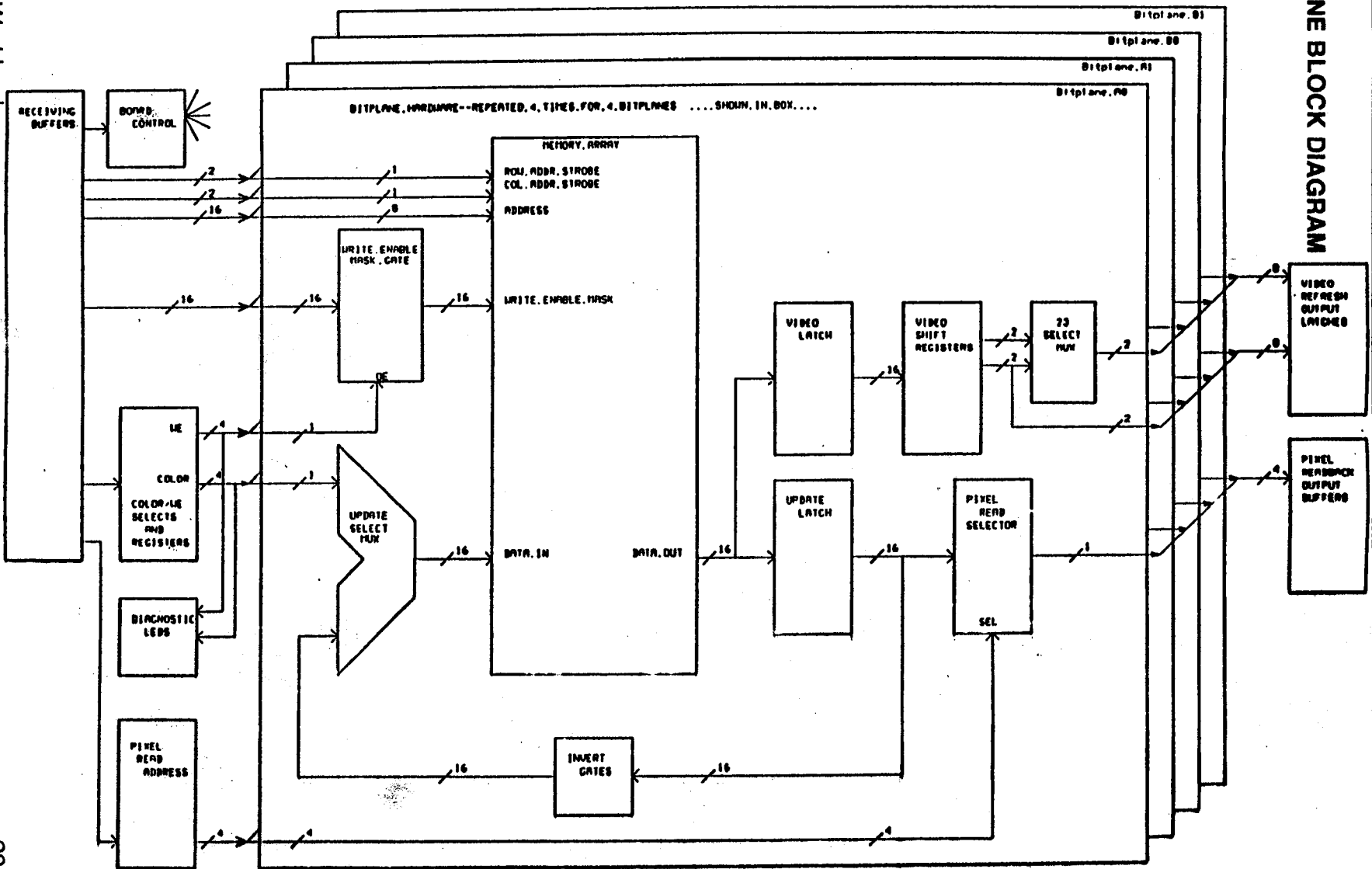
Bd #	Bitplanes	BP3	DC4	BP3	DC4 Non-Int
0	A0,A1,B0,B1	J1	J1	J3	J4
1	A2,A3,B2,B3	J1	J1	J3	J4
2	A4,A5,B4,B5	J1	J1	J3	J4
3	A6,A7,B6,B7	J1	J1	J3	J4
0	C0,C1,D0,D1	J1	J5	J3	J6
1	C2,C3,D2,D3	J1	J5	J3	J6
2	C4,C5,D4,D5	-	-	-	-
3	C6,C7,D6,D7	-	-	-	-

*To Back*

*Slot  
17  
16  
15  
14  
13  
12  
11  
10*

**Note: In non-interlaced mode, use both DC4 and DC4 Non-Int cable connections.**

BIT PLANE BLOCK DIAGRAM



**DISPLAY CONTROLLER**

DC4

Look at  
LC P7

High Resolution  
(625x 768)

Interactively Switchable 30/60hz or 30hz/NTSC or 60hz/NTSC depending on installed bootprom

30hz Interlaced  
Faster single buffer mode

60hz Non-Interlaced (Standard Monitor for all 2000 systems)  
Clearer, brighter display without flicker and with anti-glare  
Single buffer 4 times slower than 30hz  
(due to shorter retrace time available for writing)

Non-Interlaced  
TV Monitors  
Standard Color

NTSC - Non-Broadcast Quality  
636 by 485 pixels  
Spec number (RS170A)

30Hz 636x485 - Low Resolution

Broadcast-quality color registration (subcarrier frequency) requires a tolerance of ~.0003%. Our crystal has a tolerance of ~.005% for non-broadcast quality with the IRIS as the sync master. Though this is just within the RS170A spec, Color registration using our standard sync drifts sufficiently to be unacceptable in a studio environment.

Requires additional hardware:

1) Sync Generator (3rd party)  
Provides sync, subcarrier, and blank.

RS170A  
2) NTSC encoder (3rd party)  
Blends the sync generator output with the IRIS RGB signals to produce a composite NTSC video signal (video, sync, blank).

RS170A A

**NTSC Performance**

Lower resolution (fewer pixels)

Slower pixel clock, lower speed due to the longer update time (with fewer pixels, this loss is not so bad)

Poorer color resolution due to color bandlimiting during encoding (fuzzy or discolored edges - Inherent with NTSC)

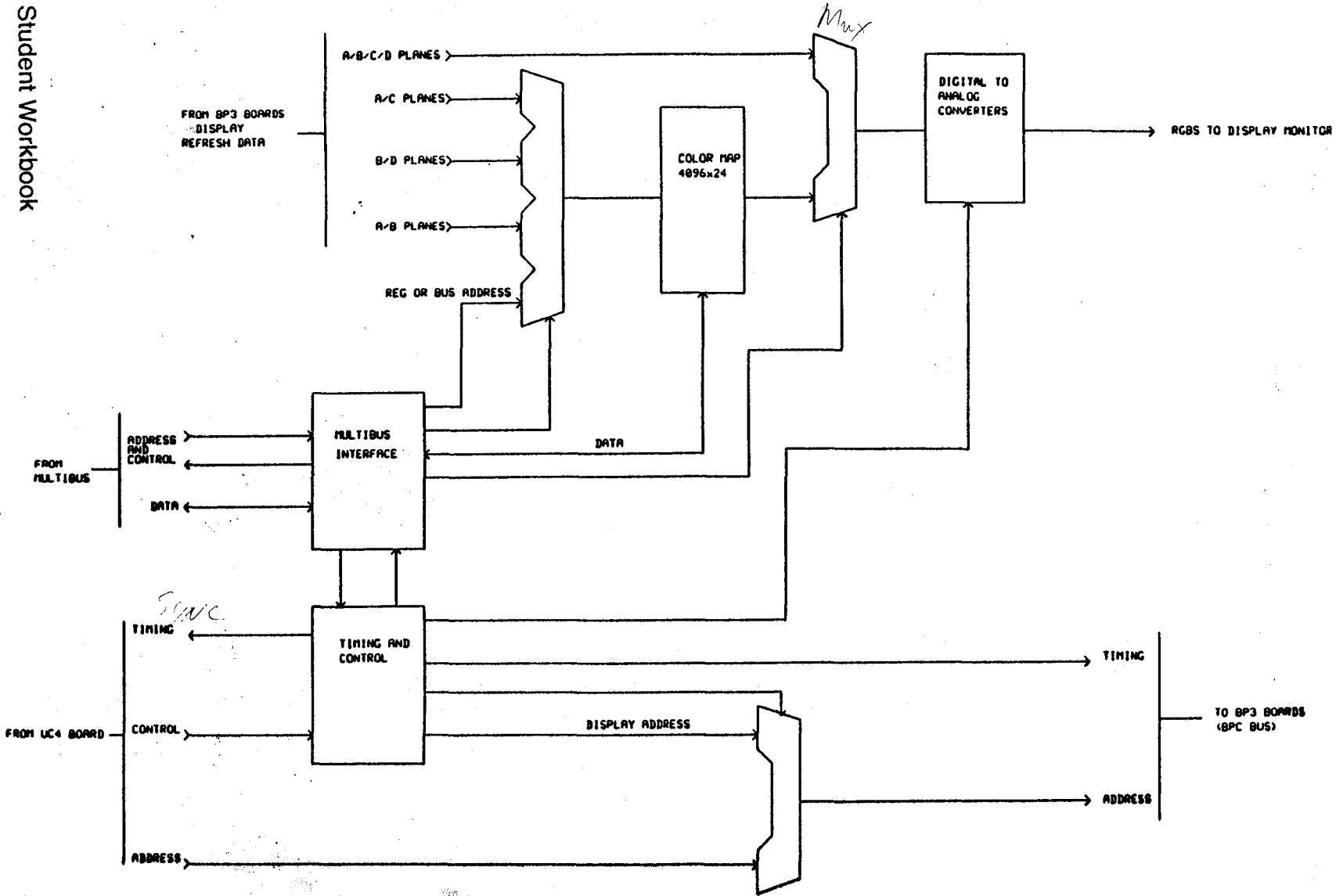
EU, European

25Hz Interlaced

*P 25-10*

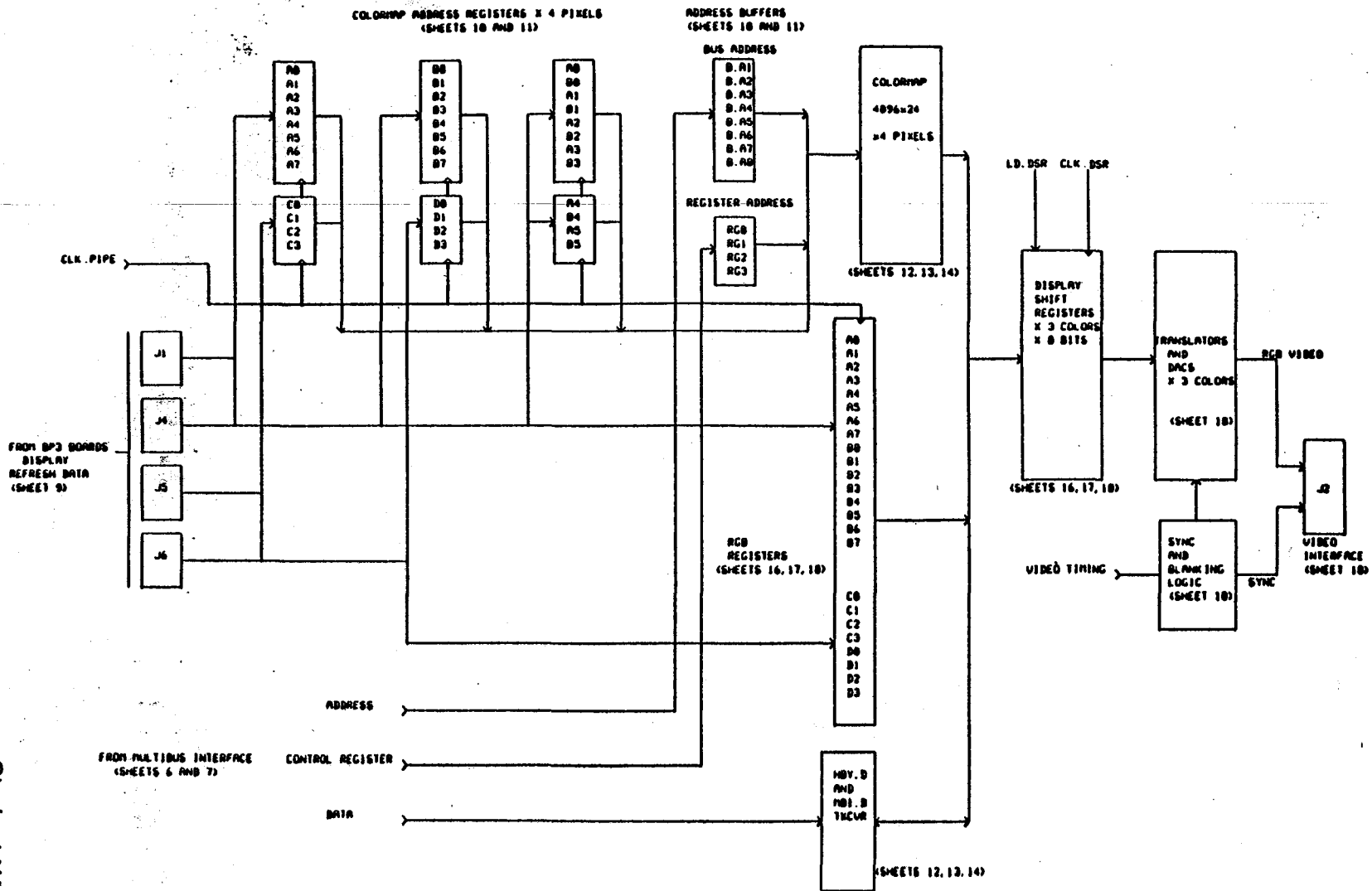
DC4 PROM CONFIGURATIONS FOR DISPLAY OPTIONS			
Part Number	Location	Pri/Sec	Display Configuration
5000-395-01	U16D	60/30	60 Hz non-interlaced display 30 Hz interlaced display  PCA PN# 5000-090-03
5000-396-01	U17D	60/30	
5000-397-01	U18D	60/30	
5000-398-01	U14B	60/30	
5000-399-03	U15E	60/30	
5000-400-03	U14F	60/30	
5000-395-02	U16D	60/RS170A	60 Hz non-interlaced display RS170A (Color TV format)  PCA PN# 5000-090-02
5000-396-02	U17D	60/RS170A	
5000-397-02	U18D	60/RS170A	
5000-398-02	U14B	60/RS170A	
5000-399-02	U15E	60/RS170A	
5000-400-02	U14F	60/RS170A	
5000-395-03	U16D	30/RS170A	30 Hz interlaced display RS170A (Color TV format)  PCA PN# 5000-090-04
5000-396-03	U17D	30/RS170A	
5000-397-03	U18D	30/RS170A	
5000-398-03	U14B	30/RS170A	
5000-399-04	U15E	30/RS170A	
5000-400-04	U14F	30/RS170A	
5000-395-04	U16D	60/PAL	60 Hz non-interlaced display PAL (European) display  PCA PN# 5000-090-05
5000-396-07	U17D	60/PAL	
5000-397-07	U18D	60/PAL	
5000-398-04	U14B	60/PAL	
5000-399-05	U15E	60/PAL	
5000-400-05	U14F	60/PAL	
5000-395-05	U16D	30/PAL	30 Hz interlaced display PAL (European) display  PCA PN# 5000-090-06
5000-396-08	U17D	30/PAL	
5000-397-08	U18D	30/PAL	
5000-398-05	U14B	30/PAL	
5000-399-06	U15E	30/PAL	
5000-400-06	U14F	30/PAL	

DISPLAY CONTROLLER: Basic Block Diagram

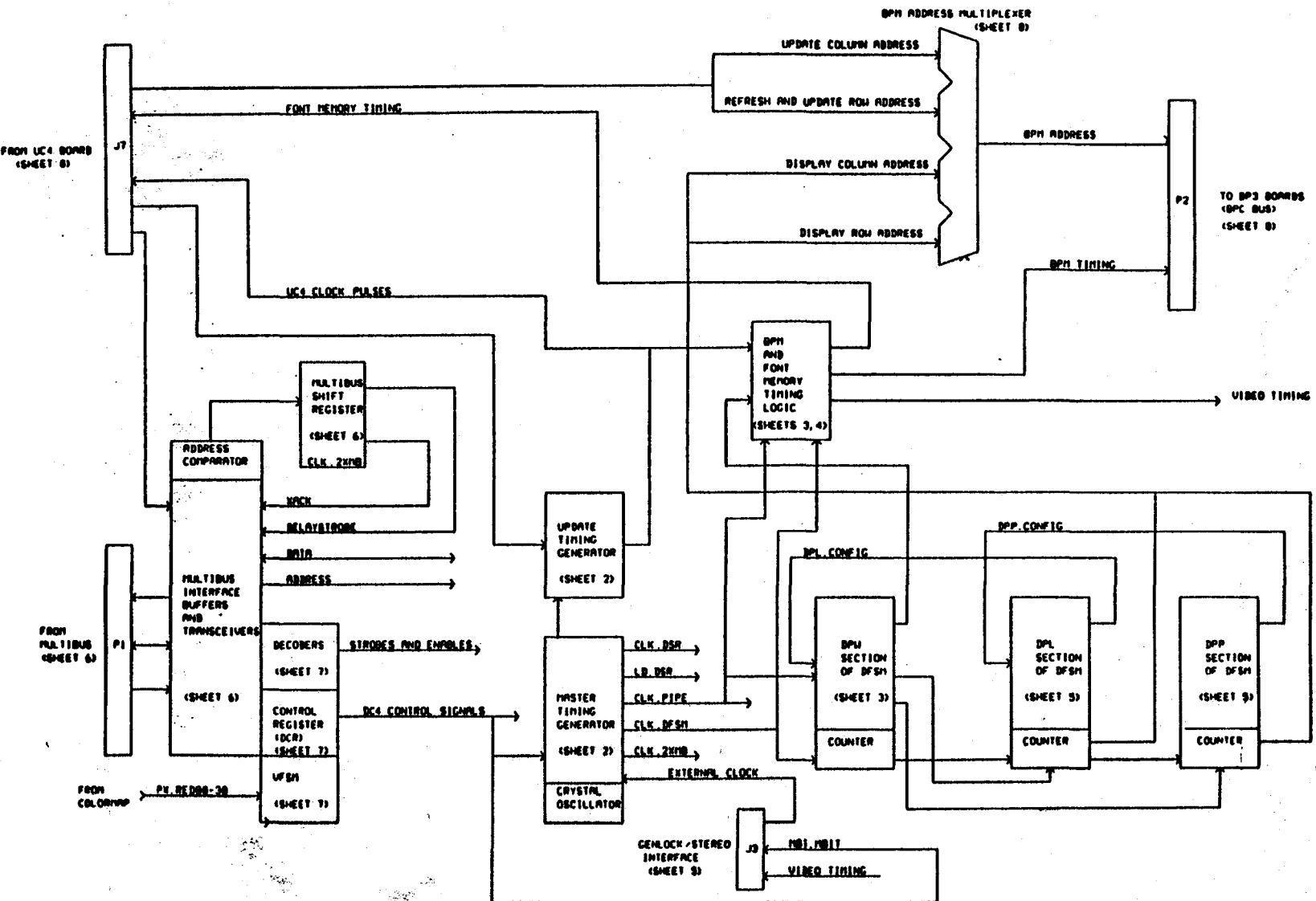




DISPLAY CONTROLLER: Detailed Block Diagram - page 1



DISPLAY CONTROLLER: Detailed Block Diagram - page 2



## GRAPHICS SYSTEM DATA FLOW

1. Applications program calls graphics subroutines.
2. Graphics Library interprets command, converting it into a token and fields of data. This information is passed on to the display list for storage OR executed (immediate mode) by command macros.

If the data had been stored in the display list, it is executed by the command macros when called.

3. The macros execute the display list commands and feeds the tokens and any data fields into the pipe.
4. The GF card contains the GA's, GE's, and the FBC. The Geometry Accelerators convert the IEEE or DEC fp into GE fp. The Geometry Engines perform the transformation, clipping, and viewport scaling against the coordinate data. The (now screen) coordinate data is passed to the FBC.
5. The frame buffer controller converts screen coordinates into commands to the Update controller for subsequent drawing. In the case of Gouraud Shading, the FBC calculates the polygon edge intensities and generates a series of depthque commands to the UC. Depth cuing is handled by the UC directly. The FBC handles all of the zbuffering operations.
6. The update controller does final scan conversion (including polygon fill & line generation), draws raster characters, and enters the pixel information into the bitplanes.
7. The display controller fetches the pixel values from the bitplanes, indexes the color map (if mapmode) and sends the resultant RGB values to the RGB DACs.

**IP2 PROCESSOR**

- **Processor Functions:** *68020*

Operating System *UNIX*

Pipe Feeding *Geometry*

Communications *4-port*

Application Programs *- GRAP*

Display List Control *- memory & graphics*

- **IP1 Features:**

True 32-bit addressing, data, and I/O

68010 instruction set compatibility

Direct access to all 16Mbytes memory via dedicated buss

Real-time clock with battery backup

Four RS-232 serial ports, two with complete modem control

Fast Memory mapped port to the Graphics Subsystem

Fast Proprietary buss to FPA

*- Floating Point*

16Mbyte physical and 64Mbyte virtual address space

Mouse interface

*on board mouse*

16Mhz clock - 3 to 4 times faster than IP1

*10 MHz*

*Compatible  
with  
IP2*

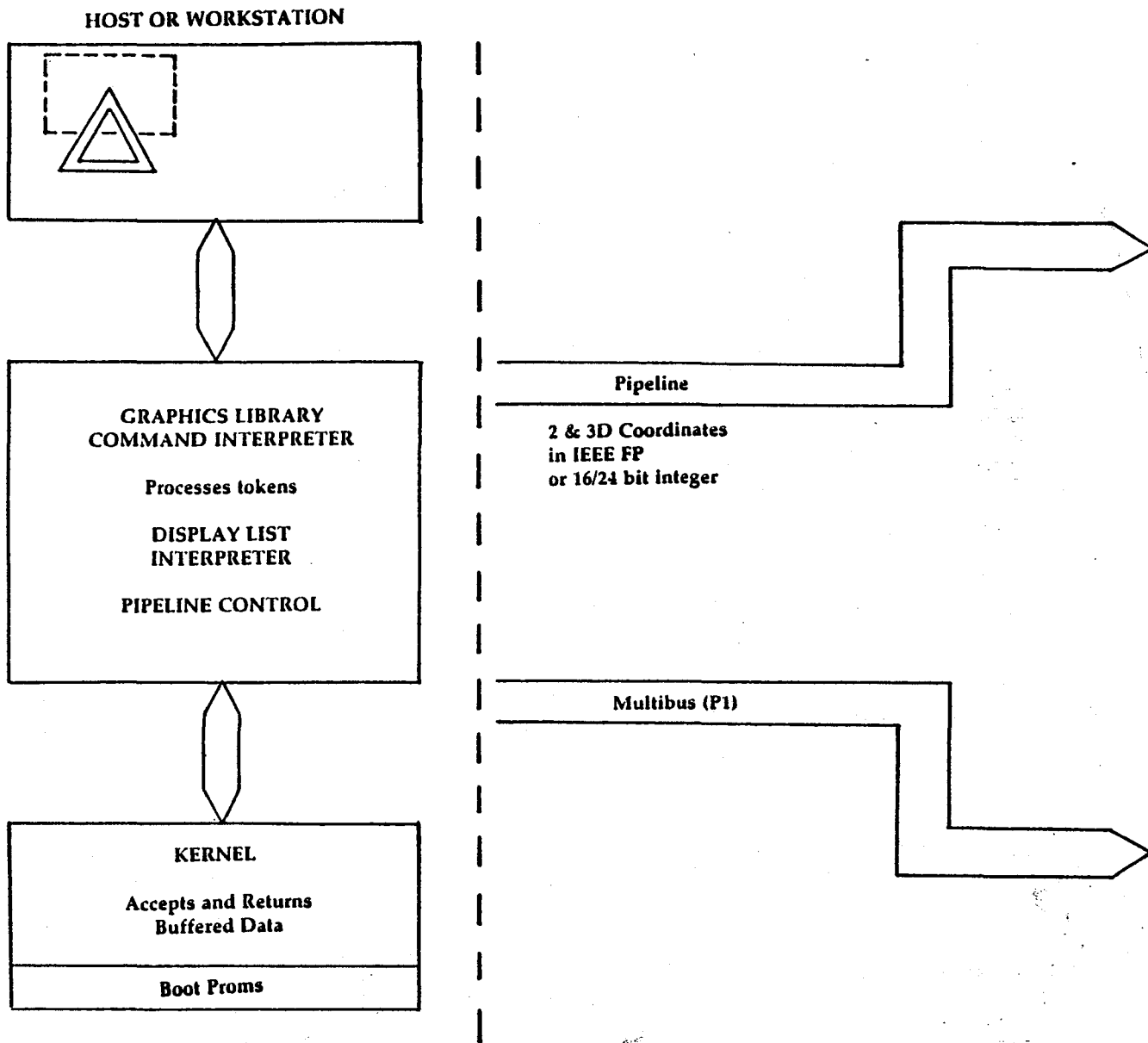
## GRAPHICS PROCESSOR: GF2

The GF2 graphics board performs several functions detailed on following pages.  
It's features include:

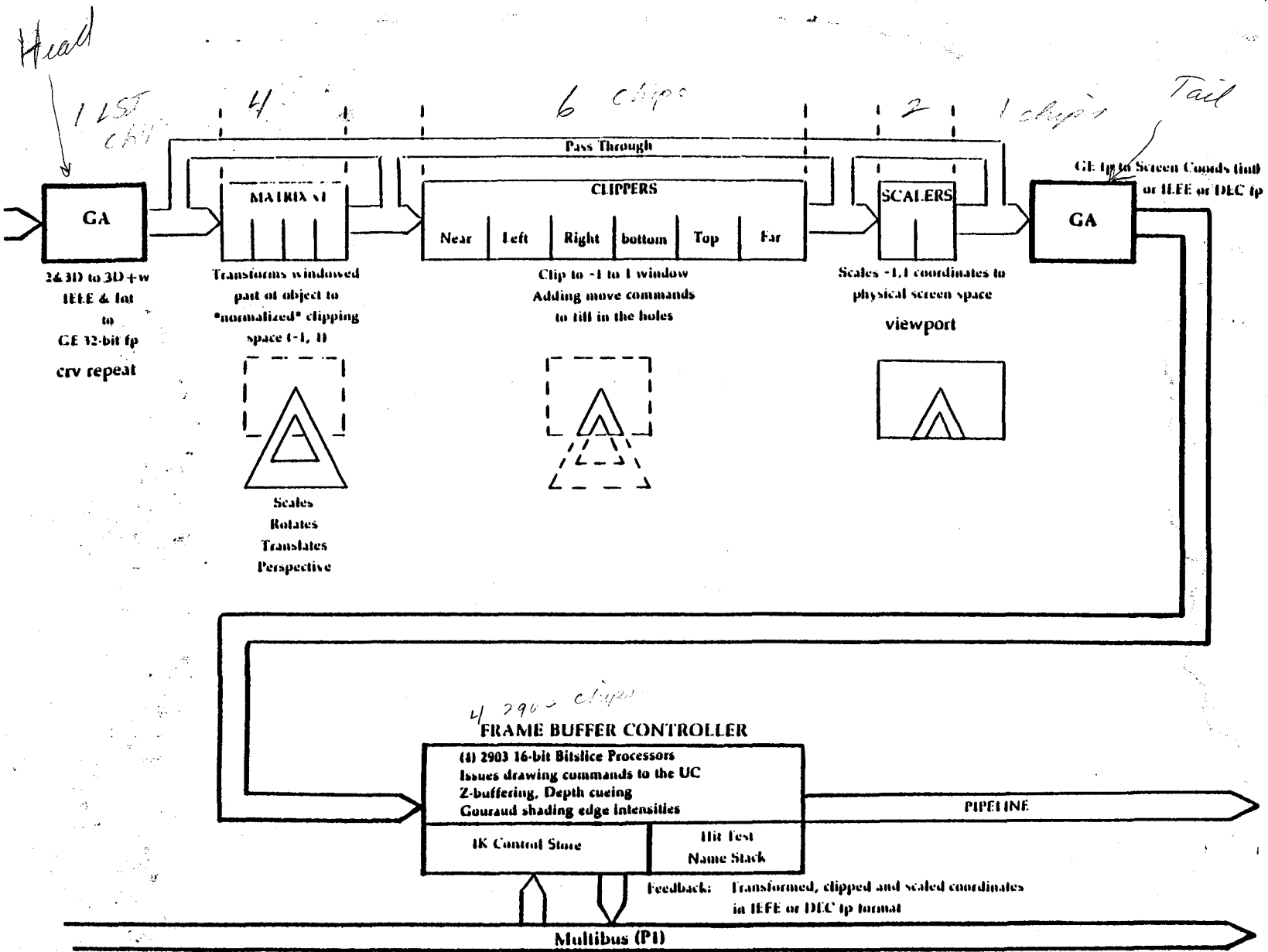
- 125ns clock speed
- **Geometry Accelerators** *Main Function*  
Buffering, Floating Point Conversion, Curve Generation
- **Geometry Engines**  
4x4 Matrix processors - Coordinate Transformations  
Clipping processors - Clip points, lines, or polygons to window  
Scaling processors - Scale image (2D) to viewport
- **Update Controller** *Frame Buffer Controller*  
Feedback - *Multi Bus*  
~~Color Conversion~~  
Z-buffering  
Gouraud Shading  
Depth Cuing

*Geometry Accelerators*

GRAPHICS SYSTEM FLOW CHART - page 1



GRAPHICS SYSTEM FLOW CHART - page 2



**FRAME BUFFER CONTROLLER****Features:**

- 4K writable control store  
Alias Research Microcode Compiler
- Faster execution (187ns clock)

**Function:**

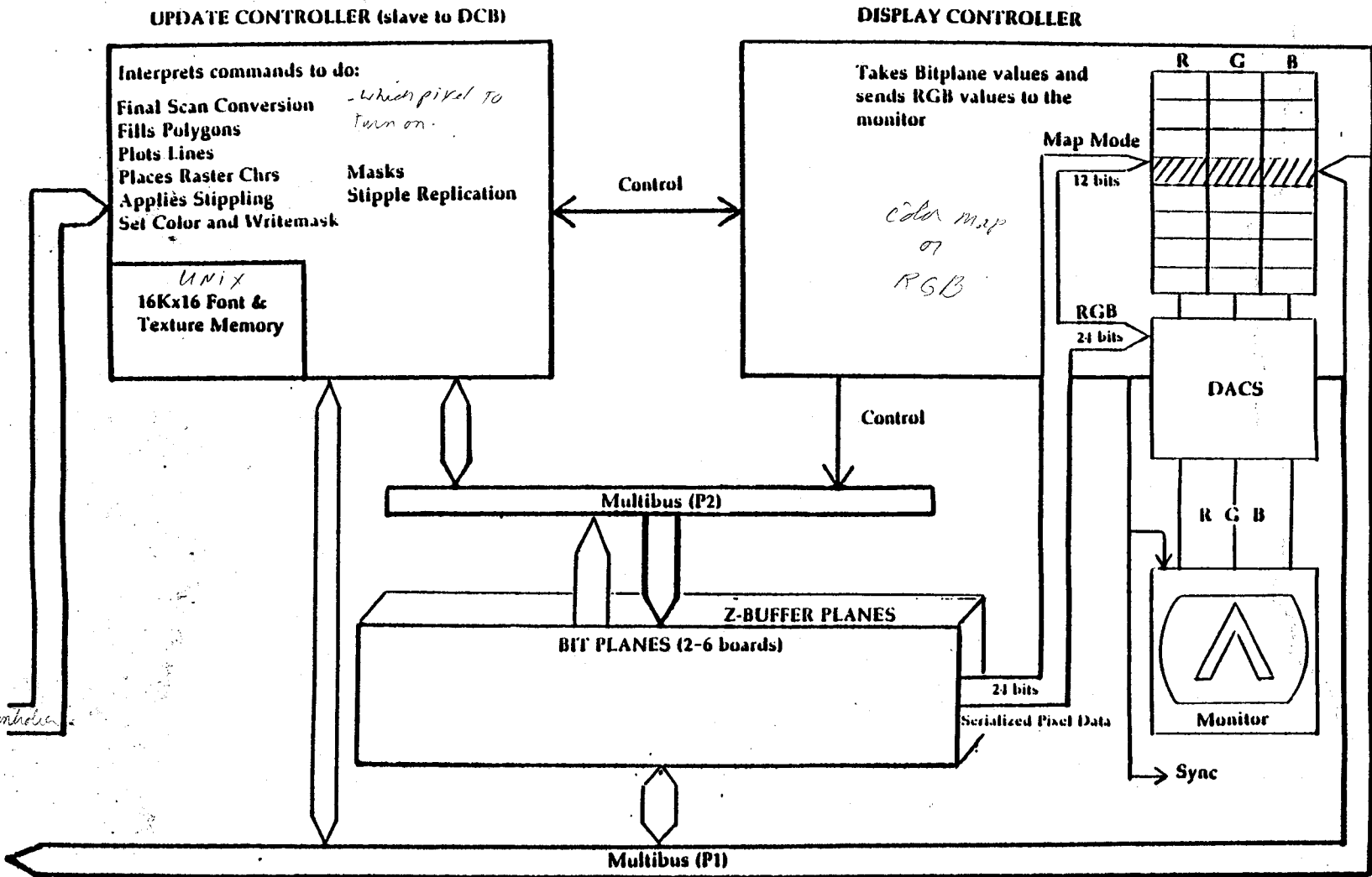
- Generates setup commands & modes for UC
- Performs Zbuffer arithmetic
- Calculates color range values for depth-queued lines
- Calculates shade ranges for Gouraud shading
- Passes Feedback to Multibus (appx 8000 vectors/sec)



*Drawing Command.*

*UDC*

*From Buffer controller*



---

**UPDATE CONTROLLER: UC4**

- Concurrent command setup and execution (executes current command while setting up next command)
- High-Speed Disk transfer to and from image memory through the UC using Fast random pixel access (2-4us/pixel) Warnings
- Support for depth-cued lines and Gouraud shaded polygons by color value increment passed from FBC
- Trapezoid Fill. This reduces the time to compute the left and right boundaries of a scan
- Supports the FBC in testing the Z-buffer bits
- Line stipple replication. This supports the scaling of the 16-bit stipple by  $n$ , where  $n$  is  $\leq 255$

ex: 0101 = 00110011

- Multiple square mask patterns (16, 32, or 64 squared) for screen masks
- Larger font memory width (16k by 16 bits)

Faster filling

Wider stipples and masks

Could be expanded to 64k by 16 chips to quadruple memory size for fonts

- Invert Mode. A drawing mode that compliments the color bits before writing to the bitplanes

**Performance:**

Clear fill rate = 44 million pixels/sec

With mask = 22 million pixels/sec (Trapezoid fill)

Arbitrary line fill = 3 million pixels/sec

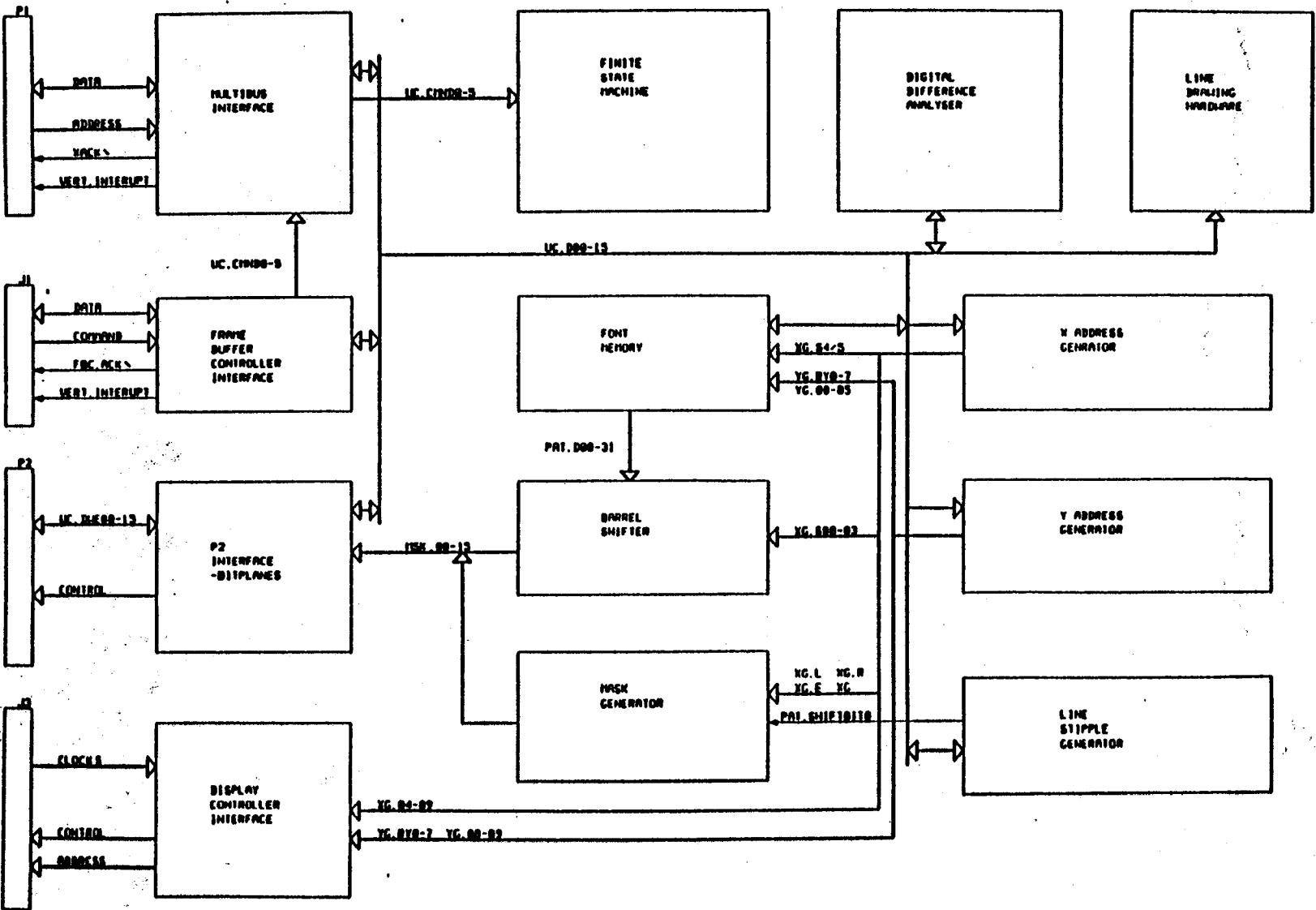
Horizontal line fill = 44 million pixels/sec (scan rate)

Gouraud Shading line fill = 3 million pixels/sec

Random Pixel R/W = 200k - 500k pixels/sec

Intensity modulated line fill (depth-cuing) = 1.5 million pixels/sec

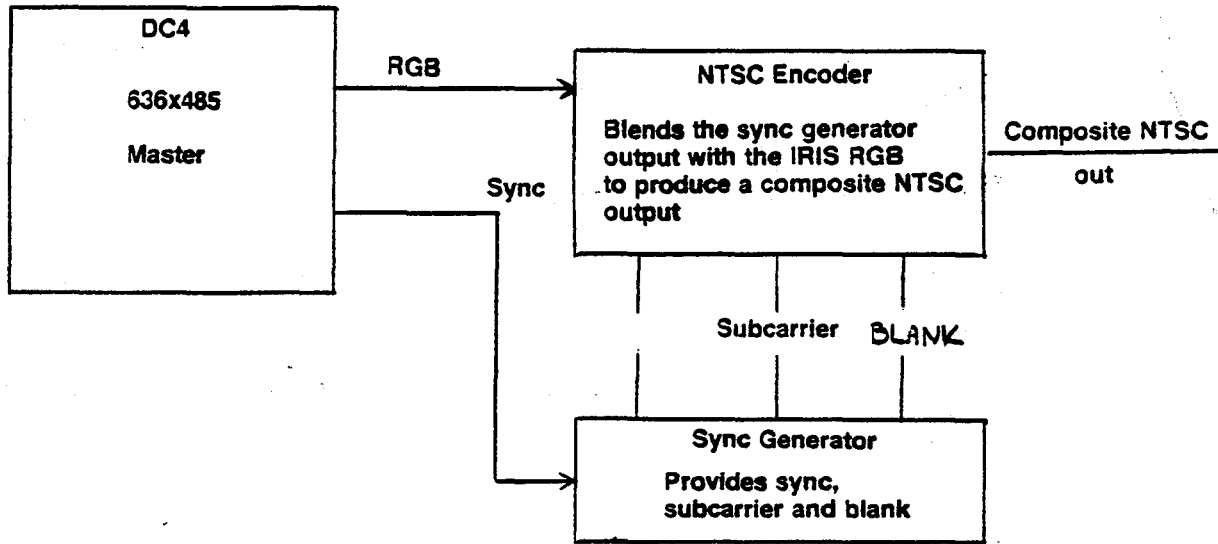
UPDATE CONTROLLER BLOCK DIAGRAM



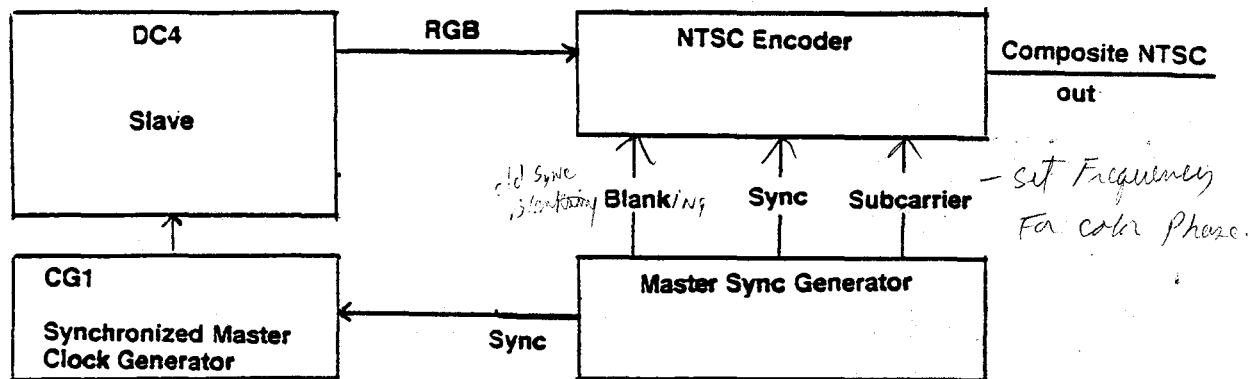
NTSC VIDEO

*CG1 Genlock*

NON-BROADCAST CONFIGURATION

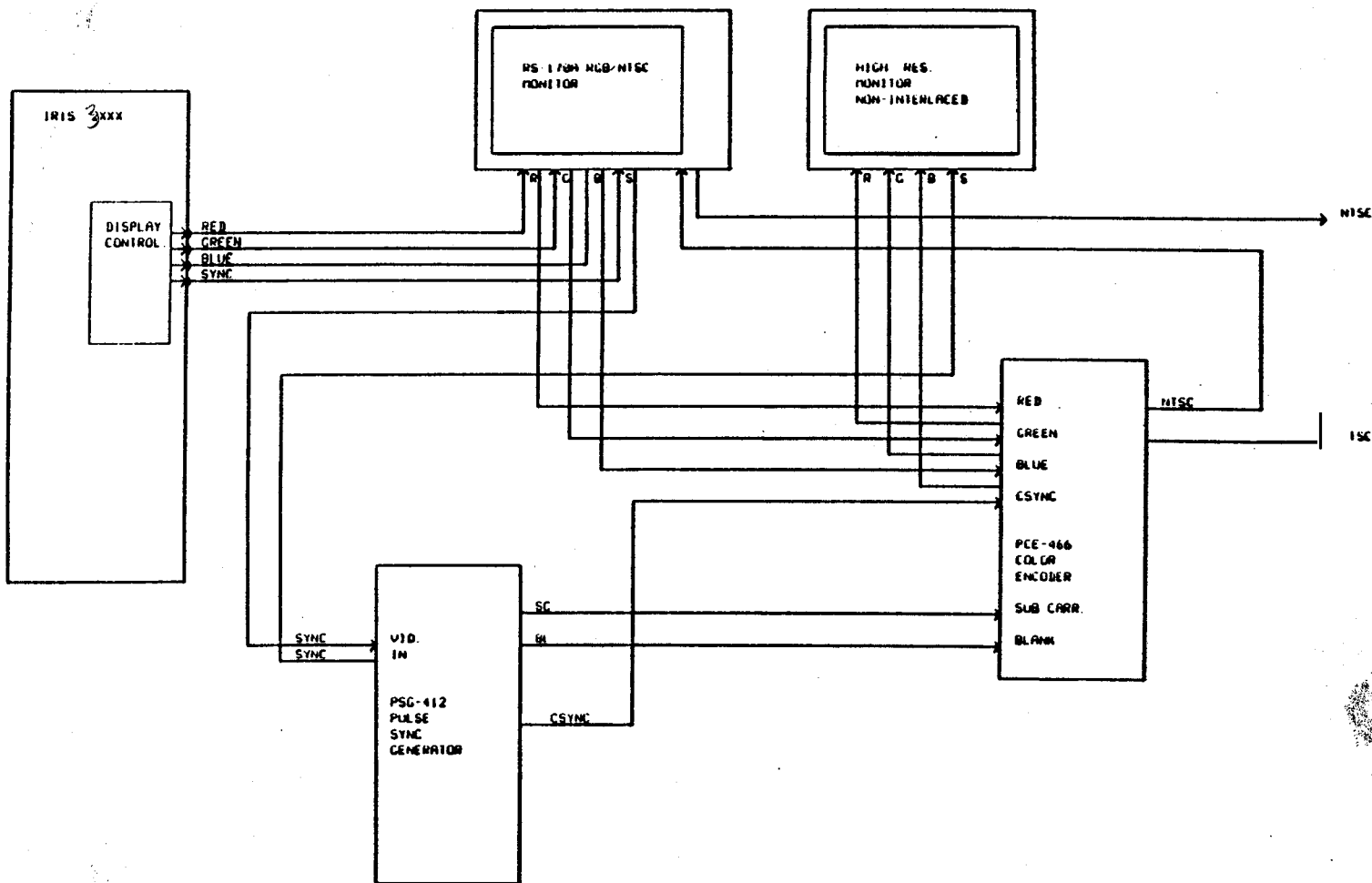


BROADCAST QUALITY CONFIGURATION



**Non-GENLOCKED NTSC**

**Non-Genlocked NTSC Operation**



**GENLOCKED NTSC****Requires:**

New SGI Synchronized Master Clock Generator (CG1)

Plugs into Multibus on either side of the partition.

2xxx series machines only.

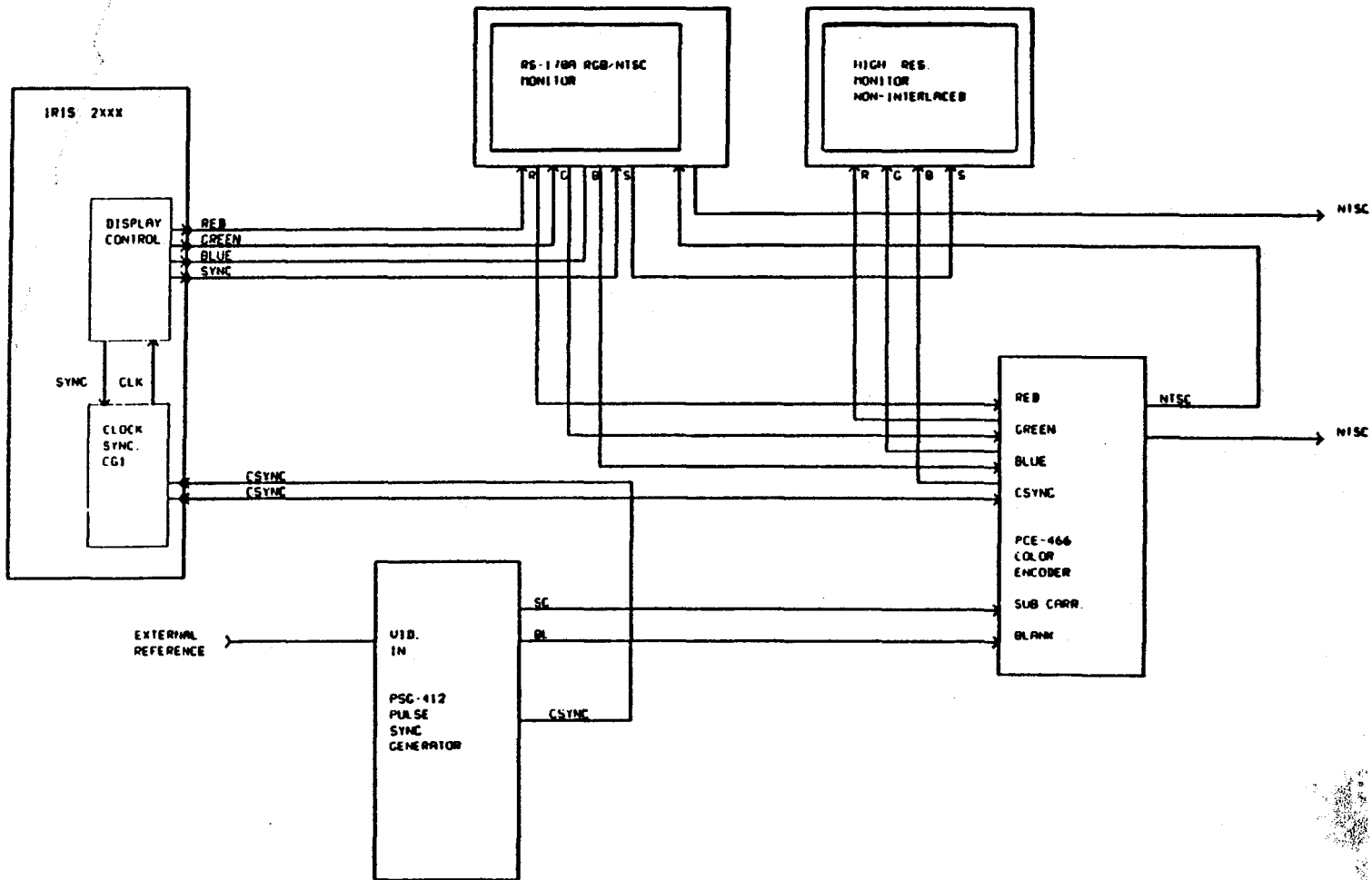
**Description:**

In this arrangement the IRIS is a slave locked (Genlocked) to the timing of a studio master sync generator. The master sync generator provides a composite sync to the CG1, which in turn controls the DC4 timing to produce RGB signals properly registered to the vertical and horizontal timing of the master. This permits the master to properly mix the IRIS output with whatever other devices the master is controlling.

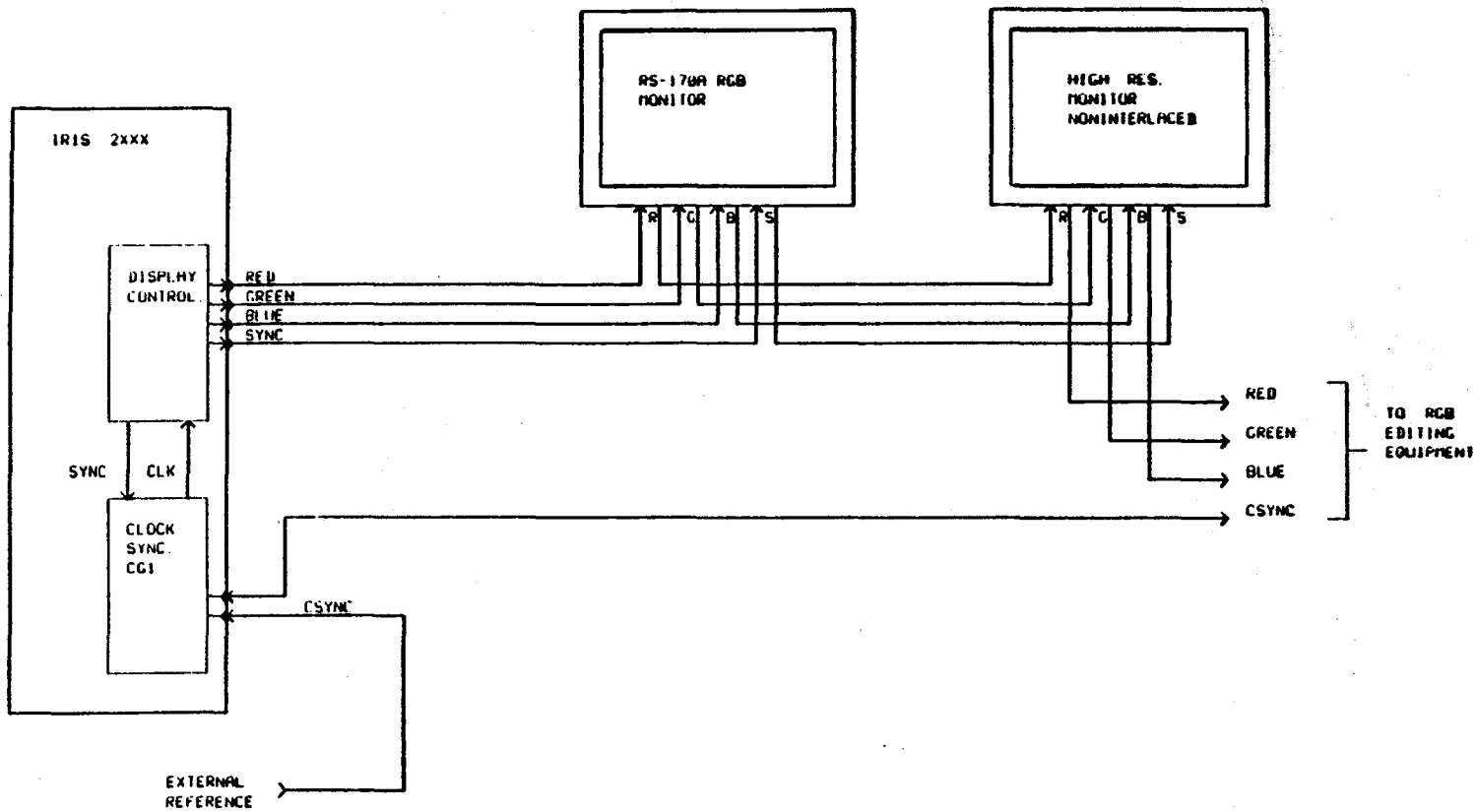
**Note:**

*CG1 on IRIS*  
Without the CG1 card, the IRIS does not have the ability to accept external control. The IRIS would, therefore, have to be the master in a studio environment. This is unacceptable to most studios.

GENLOCKED NTSC OPERATION



GENLOCKED RS-170A OPERATION





## Review Quiz 2: Lesson 6

Revision A 10/23/86

### INSTRUCTIONS:

1. You have 75 minutes to complete this quiz.
2. **DO NOT** spend too much time on any one question.
3. **DO NOT ASSUME** any facts or conditions about any question. If you do not understand the question, ask your instructor for help before you attempt to answer the question.
4. Some of the questions may have more than one answer. Select the answer that is most correct.
5. The objective of this quiz is to review the lesson information. Some of the questions you should be able to answer from recall, while others require that you refer to your class notes and/or supporting documentation.

While most of the questions will be on covered material, there will be some that require you to search your documents for the correct answer. For these questions, you will have to use the documents that you have been given, with your ability to extract the correct answer from them.

6. You may use your workstation and UNIX to help you determine the answer to any question.
7. When you complete this quiz, you may leave the room.

QUIZ 2

NAME:

1. How is it, that SGI can claim it's workstations can produce 16.7 million different colors?

256 x 256 x 256                      3 / RGB Dacs                      P 10  
1 Dac Has 256 level of intensity

2. What is the size of a single bit plane?

1024 x 1024 x 1                      11

3. What three types of data can be stored in an IRIS bit plane?

- 1. digital color code                      RGB mode
- 2. Address - color map mode                      13
- 3. Z coordinate                      14

4. What does the color map hold?

color codes that are address by Bit Plane #  
Multiplies                      24 BITS

5. What board has the responsibility for loading color codes (pixel data) into the bit planes?

UC4                      15

11. What board calculates points in space (coordinate data) for objects that are to be displayed?

GF2

12. What board maintains the UNIX character font memory?

UC4

40

13. What board has the responsibility for synchronizing the IRIS to a studio master sync generator, which supplies a composite sync signal to it?

CG1

Genlock

14. What cable carries RGB color drive to the I/O panel? (give DC4 board connector number and the cable part number)

J2

5000-208

15. Why don't the BP3 boards in slots 10 and 11 have external data cables installed at their edge connectors like the other bit plane boards?

They are used for 2 Buffering + 109 pixels

---

## **Appendix 6.1: Field Engineering Technical Aids**

This appendix supplies you with the most recent Field Engineering Technical Aids.

These aids contain information concerning all Silicon Graphics product lines.

The information is categorized into five (5) sections: **General, Interface, Diagnostics, Software, and Host**. Following this cover page is an index for the complete set of tech aids. Each section also has it's own separate index.

It is recommended that the student take about two hours and scan read each tech aid in order to familiarize himself/herself with the information supplied. It may come in handy at some future date.

Not all of the tech aids will apply to every Field Engineer, but knowing what is contained by this document cannot hurt.



## **Appendix 6.1: Field Engineering Technical Aids**

This appendix supplies you with the most recent Field Engineering Technical Aids.

These aids contain information concerning all Silicon Graphics product lines.

The information is categorized into five (5) sections: **General, Interface, Diagnostics, Software, and Host**. Following this cover page is an index for the complete set of tech aids. Each section also has it's own separate index.

It is recommended that the student take about two hours and scan read each tech aid in order to familiarize himself/herself with the information supplied. It may come in handy at some future date.

Not all of the tech aids will apply to every Field Engineer, but knowing what is contained by this document cannot hurt.

## TECHNICAL AIDS

## INDEX - TECHAID

Updated: 8/4/86

## GENERAL:

GEN-001	Tech Aids definition
GEN-002	Status indicator code definitions
GEN-003	Mouse cleaning procedure
GEN-004	Backplane cuts & jumpers
GEN-005	512K Multibus Memory addressing
GEN-006	GF1 board cooling
GEN-007	GF1 board static sensitivity
GEN-008	PM1-I/O Cable abrasion
GEN-009	PM1 board switch settings
GEN-010	Backplane voltage limiter for GF1 bd.
GEN-011	Excelan jumper configurations
GEN-012	Keyboard lockup
GEN-013	Ethernet taps, changes in
GEN-014	vi editor
GEN-015	175 ns jumper on GF1
GEN-016	Testing hard disk drives
GEN-017	Tape drives and Release 1.7
GEN-018	Tape head cleaning procedure
GEN-019	"Auto reboot in progress..."
GEN-020	High pitched noise from hard disk
GEN-021 R1	How to create a disk file system (mkfs).
GEN-022	Demo 'cclock' hang
GEN-023	Floating Point Installation
GEN-024 R1	Tape Cartridges
GEN-025	Bootable Tapes
GEN-026	2400 - New procedure for booting from floppies
GEN-027	GL2 - GF2 Boards
GEN-028	GL2 - Multibus Memory Board Modification
GEN-029	Tape drive errors
GEN-030 R1	Disk Drive Restoration Procedure
GEN-031 R1	Hard Disk Bad Blocks
GEN-032 R1	New IP1 board and PM2M1
GEN-033	Lines at bottom of Monitor
GEN-034	Tape/Floppy head cleaning procedure
GEN-035	2500 Power Receptacle requirements
GEN-036	2400/2400T - Tape and Disk errors after upgrade
GEN-037	Configuring a Second Disk as One File System

## INTERFACE:

I/O-001	Configuring DSD 5215 Disk Controller
I/O-002	Serial Port (DTR)

**TECHNICAL AIDS**

**INDEX - TECHAID**

Updated: 8/4/86

---

I/O-003	DSD Disk Controller cable chart
I/O-004 R1	DSD board jumper locations
I/O-005	Serial Interface, host (R1400)
I/O-006	Vertex V100 Drive Configuration Dipswitch
I/O-007	Vertex V100 Noise Problem
I/O-008 R1	Modems, 1400's & cu
I/O-009 R4	Connecting a printer to a 1400.
I/O-010	Unable to access tape drive
I/O-011	Bad Tape Recovery
I/O-012 R1	ttyd1 gets assigned as syscon
I/O-013 R1	Connecting a printer to a 2400.
I/O-014	XNS between two workstations
I/O-015	NTSC problems
I/O-016	Interphase SMD Controller upgrade before install of 3.4
I/O-017	Euro Video Display Problem
I/O-018 R1	IEEE-488 programming tips
I/O-019	RS-232 devices are hanging the system

**DIAGNOSTICS:**

DIAG-001	Sky Board Diagnostics
----------	-----------------------

**SOFTWARE:**

SOFT-001	Software Release R1a Overview
SOFT-002	XNSD start up (disable xns)
SOFT-003	TPE (Terminal Programming Environment).
SOFT-004	Fex - new release for 2.4/3.4
SOFT-005	GL2-W2.3 Release Notes

**HOST:**

HOST-001	Null Modem
HOST-002	DELNI Multiplexer



TECHNICAL AIDS

INDEX - GENERAL

Updated: 8/4/86

---

GEN-001	Tech Aids definition
GEN-002	Status indicator code definitions
GEN-003	Mouse cleaning procedure
GEN-004	Backplane cuts & jumpers
GEN-005	512K Multibus Memory addressing
GEN-006	GF1 board cooling
GEN-007	GF1 board static sensitivity
GEN-008	PM1-I/O Cable abrasion
GEN-009	PM1 board switch settings
GEN-010	Backplane voltage limiter for GF1 bd.
GEN-011	Excelan jumper configurations
GEN-012	Keyboard lockup
GEN-013	Ethernet taps, changes in
GEN-014	vi editor
GEN-015	175 ns jumper on GF1
GEN-016	Testing hard disk drives
GEN-017	Tape drives and Release 1.7
GEN-018	Tape head cleaning procedure
GEN-019	"Auto reboot in progress..."
GEN-020	High pitched noise from hard disk
GEN-021 R1	How to create a disk file system (mkfs).
GEN-022	Demo 'cclock' hang
GEN-023	Floating Point Installation
GEN-024 R1	Tape Cartridges
GEN-025	Bootable Tapes
GEN-026	2400 - New procedure for booting from floppies
GEN-027	GL2 - GF2 Boards
GEN-028	GL2 - Multibus Memory Board Modification
GEN-029	Tape drive errors
GEN-030 R1	Disk Drive Restoration Procedure
GEN-031 R1	Hard Disk Bad Blocks
GEN-032 R1	New IP1 board and PM2M1
GEN-033	Lines at bottom of Monitor
GEN-034	Tape/Floppy head cleaning procedure
GEN-035	2500 Power Receptacle requirements
GEN-036	2400/2400T - Tape and Disk errors after upgrade
GEN-037	Configuring a Second Disk as One File System

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-001  
Model: All IRIS  
Subject: Tech Aids Definition  
Date: 1/18/84  
From: Product Support Center

---

### Problem Definition:

No formal distribution vehicle exists for the dissemination of technical information to the field.

### Solution:

TECH AIDS are born! Hence forth, TECHNICAL AIDS will serve as the means of distributing technical information such as diagnostic aids, troubleshooting techniques and problem identification/resolution aids. The Product Support Center will be responsible for generation, distribution and maintenance of the program. However, you can contribute by submitting your own Tech Aids to the Product Support Center for approval and inclusion to the distribution.

The following is a definition of the document:

1. Number ..... defines category and sequential number within that category. There are 5 categories which are:

GEN ..... General IRIS  
I/O ..... I/O devices  
DIAG ..... Diagnostics  
SOFT ..... Software  
HOST ..... Host computers

2. Model ..... defines SGI Model affected.

3. Subject .... defines subject matter.

4. Date ..... defines composition date.

5. From ..... defines originator.

6. Problem definition ..... defines problem or failure symptom.

7. Solution ... defines corrective action or method used for problem resolution.

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-002  
Model: IRIS 1000 & 1200  
Subject: Status Indicator code definitions  
Date: 1/24/84  
From: Roger Edgar

---

### Problem Definition:

Failures resulting from power up or reset initiated diagnostics are hex coded and displayed at the Status indicator on the I/O panel. Error codes are displayed regardless of the configuration dip switch 9 setting which controls the display of 'progress' codes only.

### Solution:

A Status indicator code of 0,1 signifies diagnostics completed with no detected errors. Use the following chart to interpret the Status Indicator error codes:

Failure	Error Code	Description
Unexpected exception	0,B	Be prepared to catch certain exceptions on exit of boot state.
Local memory	0,C	Tests the minimum processor memory size needed to operate (first 128K). Random value write/verify.
Refresh timer too short	0,D	Test timing and exception handling for the memory refresh timer (arm timer, buzz loop and check count).
Refresh timer too long	0,E	This is the other extreme of the refresh timer test. An exception doesn't happen or it doesn't happen soon enough.
Context register	0,6	Write/read all possible values (16).
Segment map	0,7	Write/read random values to all segment map registers.
Page map	0,8	Write/read random values to all page map registers.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-003  
Model: ALL IRIS  
Subject: Mouse Cleaning Procedure  
Date: 1/18/83  
From: Ron Ludwig

---

#### Problem Definition:

Mouse bearings have tendency to become 'gummy' and bind after several hours of intensive use.

#### Solution:

Cleaning the mouse should be a normal procedure during PM. Be sure to have the following items on hand prior to starting; small standard screwdriver, a lint free cloth, isopropyl alcohol, and light instrument oil. The procedure is as follows:

1. Remove the mouse from the monitor control panel.
2. Remove the two screws from the bottom of the mouse.
3. While holding the mouse upside down, push on the three buttons and remove the base from the housing. Do not use the cable as a leverage to remove base. The pushbutton caps are not secure and should fall out freely.
4. Remove the two screws holding the center ball housing.
5. Place the base onto a flat surface and remove the circuit board.  
< Caution >.....Be careful not to damage the contact arms!
6. Remove the three housings from the center and two small balls.
7. Clean the three balls with alcohol.
8. Clean the center balls washer & spring washer with alcohol.
9. Lubricate the three shaft bearings located on the center ball's plastic housing with one drop of light oil each.
10. Clean the two red contact wheels with alcohol.
11. Insert the two small balls and their plastic housings into the base.
12. Insert the center ball spring washer, flat washer and ball into the base in that order.

Number: GEN-003

---

13. Place the center ball plastic housing over the ball, slide the circuit board back into position, and insert the two mounting screws.

14. Place the mouse housing upside down on a flat surface and insert the three pushbutton caps so that the plastic tips are facing the outer edge.

15. Insert the base into the mouse housing and secure with the two mounting screws.

16. Reconnect the mouse to the monitor control panel.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-004  
Model: IRIS 1000  
Subject: Backplane Cuts and Jumpers  
Date: 1/23/84  
From: Ron Ludwig

---

#### Problem Definition:

In 10 card slot systems, the use of slots 4 & 5 are configuration dependent via backplane jumpers and potentially a board inserted into either of these two slots may not work correctly if the jumpers are not installed specifically to match that configuration. This is done to make 10 slots accommodate various system configurations. Care should be exercised when board swapping is done or configuration changes are made in the field to insure that the backplane configuration jumpers support the intended use of either card slot. Alterations to the backplane jumpering scheme may be required when changing the original system configuration.

#### Solution:

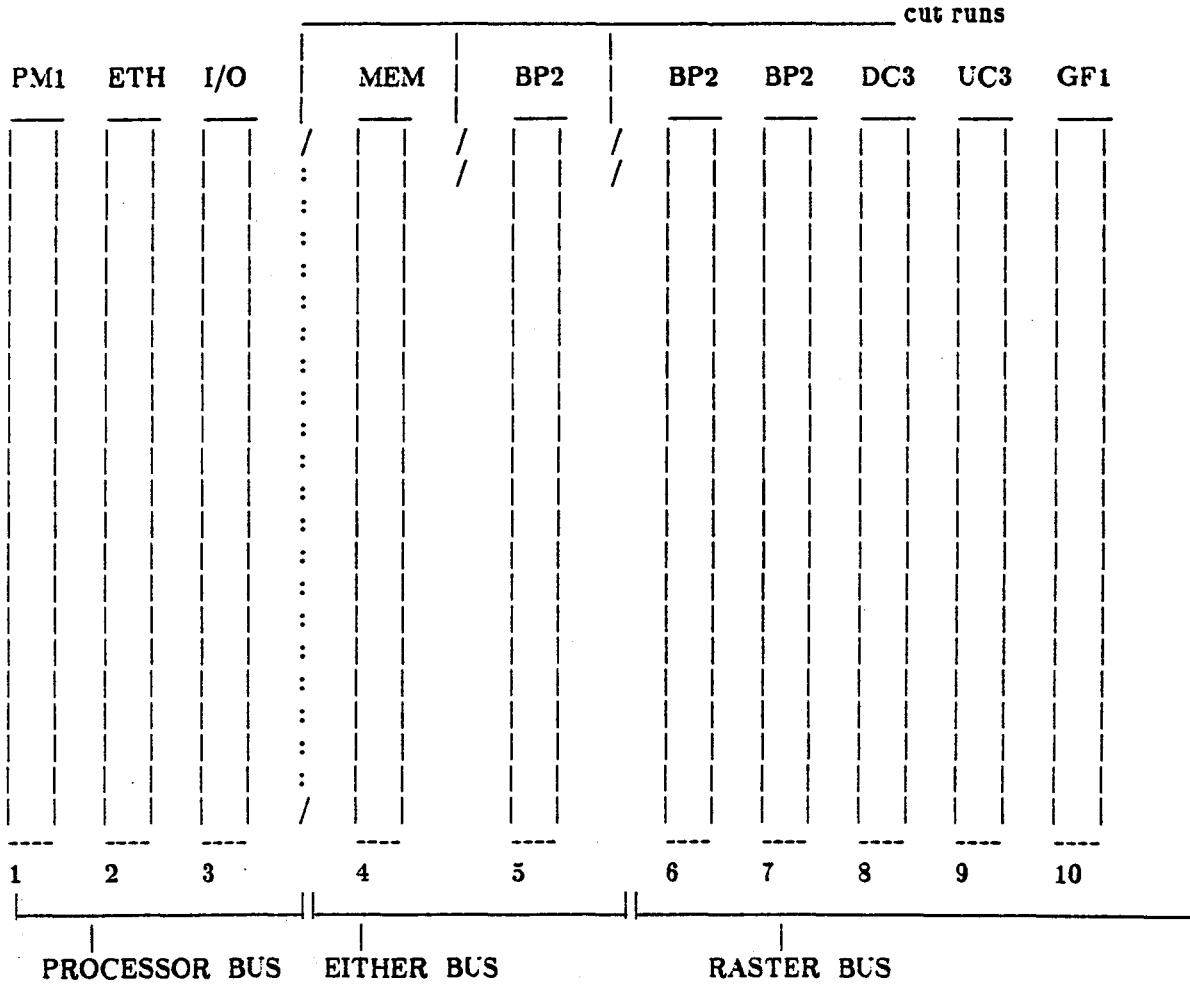
Use the following illustration and specifications to determine the correct backplane jumpering scheme for any given configuration:

### CARD CAGE INFORMATION

Each of the 10 Multibus card slots has two connections. The bottom connection is labeled P1, consisting of 86 pins and the top connection is labeled P2, consisting of 60 pins. The P2 bus is divided into a processor bus and a raster bus by cutting the backplane etch runs. All 60 runs are cut between slots 3 and 4. Four runs are cut between slots 4 and 5, and 5 and 6. These four runs become address lines when jumpered to the processor bus side, or control lines when jumpered to the raster bus side.

Number: GEN-004

TYPICAL CONFIGURATION



BACKPLANE JUMPER SPECIFICATIONS

1. Card slots 1 thru 3 are dedicated to the processor bus.
2. Card slots 6 thru 10 are dedicated to the raster bus.
3. Card slots 4 and 5 may be allocated to either bus according to the jumper configuration on the backplane.
4. If processor bus cards (ie. Ethernet, I/O controller, Memory) are to be in slots 4 and 5, then the jumpers on the backplane must connect slot 3 to 4, and 4 to 5.

Number: GEN-004

---

5. If raster bus cards (ie. Bitplanes) are to be in slots 4 and 5, then the jumpers on the backplane must connect slot 6 to 5, and 5 to 4.
6. If a processor card is to be in slot 4 and a raster bus card is to be in 5, then the jumpers on the backplane must connect slot 3 to 4 & 6 to 5.
7. The PM1 card must be in a slot to the left of any DMA controller (ie.: Ethernet, disk controller, etc.) because the PM1 card must have the highest bus priority.
8. The GF1 card must be in slot 10 for cooling considerations.



# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-005  
Model: IRIS 1000 & 1200  
Subject: 512K Multibus Memory Board Addressing  
Date: 1/20/84  
From: Roger Edgar

---

### Problem Definition:

The 512K Multibus Memory board (SGI P/N 93-0001) responds to an address range defined by a combination of board switch settings and a wirewrap jumper selection. Switch and jumper settings on replacement boards must match the switch and jumper settings of the board being replaced. A wrong switch or jumper setting will result in either duplicated memory addresses or a 'hole' in the contiguous address string with those addresses beyond the 'hole' unable to respond.

### Solution:

When replacing memory boards in the field or swapping address ranges between boards via the switch and jumper settings to isolate intermittent memory errors, insure that the address range selected for any given board is correct. Switches 'SW1 and SW2' select the upper and lower address limits for any given 512K boundary while the wirewrap jumpers 'E16 thru E20' select the megabyte boundary that the board will respond to. Use the following chart to insure correct and contiguous addressing:

Board	Address Range	SW1	SW2	Wirewrap
		8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	Required
1	000000 - 07FFFF	0 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	None
2	080000 - 0FFFFFFF	1 1 1 1 1 1 1 1	1 0 0 0 0 0 0 0	None
3	100000 - 17FFFFF	0 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	E16 to E20
4	180000 - 1FFFFFFF	1 1 1 1 1 1 1 1	1 0 0 0 0 0 0 0	E16 to E20

### Notes:

1. Open = 0, closed = 1
2. Jumper pins E16 & E20 are the two outer pins located between IC's Z13 & Z24.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-006  
Model: IRIS 1000 & 1200  
Subject: GF1 Board Cooling  
Date: 1/23/84  
From: Roger Edgar

---

#### Problem Definition:

The GF1 board typically generates more heat than other boards in the system and therefore requires special cooling considerations.

#### Solution:

The card cage design compensates for this consideration by allowing extra airflow through slot 10 in the Model 1000 and slot 20 for the Model 1200. Make sure the GF1 board always resides in the applicable card slot.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-007  
Model: IRIS 1000 & 1200  
Subject: GF1 Board Static Sensitivity  
Date: 1/25/84  
From: Roger Edgar

---

#### Problem Definition:

The GF1 board is sensitive to electrostatic discharge and can easily be damaged by improper handling and/or storage techniques.

#### Solution:

Always store spare boards in an anti-static bag, even when the boards are in the card caddy.

A *Portable Field Service Kit* from 3M is available through the Product Support Center that is designed for electrostatic protection of static sensitive components. The kit is comprised of a 24" x 24" conductive plastic sheet *Work Surface* and two grounding straps, one for *work surface to frame* and the other for *wrist to frame*. A one megohm resistor is designed into each of the grounding straps as a safety measure. Once grounded, the conductive plastic sheet cannot hold static charge. The high conductivity of the material insures it will shield the board from static laden carpeting or vinyl floor tile.

When handling GF1 boards in the field, adhere to the following procedure:

1. Set up the conductive work surface with the grounding strap connected to the system frame.
2. Attach the wrist grounding strap between your wrist and the system frame.
3. Remove the replacement GF1 board from its anti-static bag and place it on the grounded work surface.
4. Remove the defective GF1 board from the system and place it on the work surface.
5. Insert the replacement GF1 board into the system.
6. Place the defective GF1 board into an anti-static bag for return to the plant.

A good habit is to faithfully practice static control with all boards because it not only reduces call backs but also saves the customer and the company money. When static control is neglected, the result can lead to ex-

SILICON GRAPHICS FIELD ENGINEERING

---

TECHNICAL AIDS

(Cont'd)

Page 2

Number: GEN-007

---

tensive debugging periods lasting weeks and sometimes months.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-008  
Model: IRIS 1000 & 1200  
Subject: PM1-I/O Cable Abrasion  
Date: 1/26/84  
From: Roger Edgar

---

#### Problem Definition:

The PM1-I/O cable (p/n 5000-210) scrapes against the component leads of the board located in slot 1 as the board is moved in and out of the slot, potentially resulting in a damaged cable.

#### Solution:

The obvious and simplest solution is to physically hold the cable out of the way with your hand when working slot 1. In the Model 1200's, configurations may allow not using this slot, thus reducing the risk of damaging the cable.

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-009  
Model: IRIS 1000 & 1200  
Subject: PM1 Board Switch Settings  
Date: 2/7/84  
From: Roger Edgar

---

### Problem Definition:

The PM1 board switch is OR'd with the configuration switch on the IRIS I/O Panel and must be set correctly for the configuration switch to function correctly. The definitions of switch positions 1-10 on the PM1 board are not in the same order as the definitions of switch positions 1-9 on the configuration switch and an improper switch setting on the PM1 board will result in defeating the configuration switch settings.

### Solution:

Insure that the PM1 board switch settings are always set such that all positions are open except position 1 which should be closed. A bad I/O panel configuration switch position could be ascertained however by toggling the appropriate PM1 board switch position as a test. The following information may be helpful:

#### Switch Positions and Settings

PM1 Bd	I/O Panel	Function	Setting	Definition
10	1	Reserved	1	Leave in open position.
9,8	2,3	Serial line	00	300 baud
			01	19.200 baud
			10	1200 baud
			11	9600 baud
7	4	Silent	0	Report startup configuration data; values of configuration switches, physical size and virtual address for onboard and Multibus I/O. The Keyboard Port switch must be set to a "1".
			1	No startup configuration is displayed.
6	5	Checkout	0	No additional testing.
			1	Additional memory testing (time consuming).
5,4	6,7	Boot environment	00	Boot from a floppy disk.
			01	Prom monitor command interpreter.
			10	Boot over a network connection.
			11	Boot IRIS Terminal software over a serial line.
3	8	Keyboard port	0	Set for a up/down encoded keyboard.
			1	Set for a 9600 baud ASCII terminal.
2	9	Progress codes	0	No progress codes displayed.
			1	During the startup sequence a digit on the status indicator changes at various points of the startup procedure.

### Note:

1 means open, 0 means closed

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-010  
Model: IRIS 1000 & 1200  
Subject: Backplane Voltage Limiter for GF1 Board  
Date: 2/13/84  
From: Roger Edgar

---

#### Problem Definition:

Certain GF1 boards require an additional voltage source of 4.1vdc to 4.3vdc present on the backplane to accommodate Geometry Engines that run in this range as opposed to 5.0vdc. Either type of engine is acceptable but IRIS models shipped prior to February 1984 do not have the 4.1vdc to 4.3vdc present on the backplane and cannot be fitted with GF1 boards having this prerequisite until a voltage limiting diode assembly is installed on the backplane to provide the voltage source.

#### Solution:

Board engine types are easily identified by a physical inspection of the board at location H13J; four jumpers are present for 5 volt engines and the absence of the four jumpers means 4 volt engines. Board with 5 volt engines work with or without the presence of the voltage limiter diode assembly. Spare boards sent to the field will always be populated with 5 volt engines. An FCO (field change order) will follow to retrofit all IRIS units without the voltage limiter diode assy to alleviate compatibility considerations.

Theoretically, the only time this should be a consideration is when a 4 volt GF1 board ends up as a spare card as a result of someone taking a 4 volt card out of a system in effort to isolate a field problem. Once the FCO's are installed on all existing IRIS units it will no longer be of concern as units will be compatible to both type Geometry Engines.

SILICON GRAPHICS FIELD ENGINEERING

---

**TECHNICAL AIDS**

Number: GEN-011  
Model: 1000, 1200  
Subject: Excelan Jumper Configuration  
Date: Feb. 17, 1984  
From: John Carter

---

**Problem Definition:**

Potential operational failure caused by wrong jumpering of Excelan board.

**Solution:**

The correct jumper configuration of the Excelan board is Address 0, Priority Interrupt 2, 24 bit addressing mode, and Multibus interface connected to BPRO/.

Refer to page 136 of the EXOS/101 reference manual. The only jumpers that should be installed are:

JU22 select interrupt level 2  
JU29 enable 24-bit addressing  
JU30 connect BPRO/ to Multibus interface



# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-012  
Model: IRIS 1000 & 1200; R1b  
Subject: Keyboard lockup  
Date: 10/4/84  
From: John Carter

---

### Problem Definition:

Keyboard locks up, sounding beep, all keyboard LED's are on except 'online'.

### Solution:

This is a symptom of an ASCII message being sent to the keyboard. It is assumed that there would be an ASCII terminal connected to port 1 (along with the color monitor) that could display these messages.

The only way to see these messages, while still running graphics, and prevent the keyboard from locking up, is to use an ASCII terminal connected to port 1 along with the color monitor, using a special cable (described below).

With the ASCII terminal properly connected to port 1 reboot the system and watch for the error messages on that terminal and debug the problem from there.

To connect an ASCII terminal to port 1 along with the color monitor use a cable that has the following configuration:

P1	P2	P3
1	1	1
2	-	2
3	-	3
4	4	-
5	5	-
6	6	-
7	-	7
8	8	-
9	9	-

(pins 10-25: P1 to P2 is one to one; nothing to P3)

All connectors are male 25 pin DIN (RS232 variety).

Connect the cable with P1 to port 1 on the terminal, P2 to the connector on the color monitor, and P3 to the ASCII terminal.

When the special cable has been connected, unplug the keyboard from the color console, otherwise it will interfere with normal operations. All keyboard interaction will now be from the ASCII terminal.

When the system is rebooted the error messages will be displayed on the ASCII terminal (along with all other booting information) and you will be able to identify the type of error encountered during boot, and even during graphics operations (i.e., GE traps).

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-013  
Model: All  
Subject: Ethernet taps, changes  
Date: May 15, 1984  
From: John Carter

---

#### Problem Definition:

TCL, the manufacturer which is supplying our ethernet transceiver parts, announced a change in the design of these parts.

#### Solution:

The old transceiver has a rigid stinger. The new transceiver has a spring loaded stinger. The old type will be phased out by the manufacturer which is one good reason for the change. The manufacturer also believes the new stinger type will alleviate contact problems that have been reported to them in the past. The new stinger requires a third tapping tool to pierce the inner insulation for the stinger. This new tool (part number A0003-D2-0) is in stock at SGI. The coring tool must also be a new type (A0003-D0-1) which clears a bigger hole and is also in stock. To my knowledge, the older tool (A0003-D0-0) has never been in stock at SGI.

The new type transceiver comes with a new design tap block. The footprint on the cable is much longer and grips have been added inside the clamp to prevent the cable from moving around in the clamp. Additionally, there are four spikes instead of two to grab the cable. The shims that come with the kit are for use with the teflon cables only (.375" dia., orange), and should be cut to one inch length. These shims must be carefully centered in the tap block so as not to interfere with the protruding grips inside the clamp. It is also recommended that this shim be slit lengthwise in half and only one half of the shim used where the hole is to be bored. The reason for this is that if the whole shim is used it may overlap around the cable adding extra dimension to the cable diameter and cause the two halves of the tap block to be forced apart. The purpose of the shim is to provide a humidity shield for the smaller diameter cables.

When installing the tap block on the cable, be sure that the two halves meet each other around the cable. If they don't, the shield removing tool may not reach the shield and the piercing tool may not reach through the inner insulation far enough to allow the spring loaded stinger to finish piercing the insulation and make contact with the inner copper wire. When installing the tap block on the large diameter cable (.400" dia., yellow) this could be an exceptionally tedious problem as the cable grips on the tap must compress the cable considerably to allow the two halves of the tap block to meet all around. A long handle box wrench and a pair of vice grips might ultimately be needed to accomplish this task.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-014  
Model: R1400  
Subject: vi editor  
Date: June 7, 1984  
From: John Carter

---

#### Problem Definition:

Getting around in vi the first time.

#### Solution:

There are very few commands one needs to know to use vi effectively. These commands are:

i Inserts text at cursor position. Must exit with ESC key.  
I Inserts text at beginning of line. Must exit with ESC key.  
a Appends text after cursor position. Must exit with ESC key.  
A Appends text at end of line. Must exit with ESC key.  
o Creates new line below cursor position and goes into insert mode.  
Exit with ESC key.  
O Creates new line above cursor position and goes into insert mode.  
Exit with ESC key.  
dd Deletes line at cursor position.  
D Deletes characters from cursor position to end of line.  
x Deletes character under cursor.  
SPACE Moves cursor forward nondestructively to end of line.  
RETURN Moves cursor to beginning of next line.  
arrows Exits insert mode and moves cursor nondestructively in direction  
of arrow (if in insert mode last key stroke is lost).  
ZZ Writes to file and exits vi.

Any command preceded with a numeric value, e.g., 3dd, executes that command that many times.

I recommend that when you are entering text try to keep your screen looking clean. Do this by using the RETURN key when you get to the end of the screen instead of letting the text wrap around. This also allows you to edit a given sentence easier. Another benefit is that if you should print your file you won't lose text that is longer than the page is wide.

Number: GEN-014

---

Advanced commands:

**:wq** Same as ZZ.  
**:w file** Writes to named file, returns to vi. Without named file, updates current file.  
**:r file** Insert named file at cursor position.  
**:q!** Exits vi without writing to file.  
**:sh** Exits vi into a new shell. Use CTRL-d to return to vi. Text is not lost. Must use :w first.  
**/text** Finds occurrence of text in file. Terminate with RETURN.  
**n** Finds next occurrence of text defined with / command.  
**N** Finds previous occurrence of text defined with / command.  
**fc** Finds character c following cursor in current line.  
**Fc** Finds character c prior to cursor in current line.  
**rc** Replaces character under cursor with character c.  
**R** Replaces text starting at cursor. Exit with ESC key.  
**dw** Deletes balance of word from cursor position.  
**cw** Changes balance of word from cursor position. Must use ESC to exit.  
**u** Undo last command.  
**w** Moves cursor forward to beginning of next word.  
**b** Moves cursor backward to beginning of previous word.  
**CTRL-l**Repaints screen.  
**CTRL-d**Scrolls screen down.  
**CTRL-u**Scrolls screen up.  
**ESC** Aborts any keystroke sequence not yet a complete command.

This should get you started.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-015  
Model: All IRIS  
Subject: 175 ns jumper on GF1  
Date: 07/18/84  
From: J Carter

---

#### Problem Definition:

Configuring the GF1 board to run at 175 nanoseconds.

#### Solution:

On the GF1 board locate the jumper pin block at IC position H15F. Holding the board with the fingers pointing toward you, pin 1 is at the upper right and pin numbering will then be the same as for an IC. Each pair of pins (opposing sides) on this block represent a different access speed for the GE's.

Starting at pair 1-16 the speed is 400 ns and reduces in increments of 50 ns for each pair down to 50 ns. There will be a jumper between pins 4 and 13. This configuration is set for 250 ns.

To change the speed to 175 ns, first move the jumper to pins 5 and 12. Then, add a wire wrap between pins 1 and 9. Test the board using the flight and robot demos. If it works, you have been successful.

Note on the configuration sheet on the front panel of the workstation the new speed which the board is configured for.

Unfortunately, the modified board may not be as good as you think because you cannot test it at the voltage limits and you cannot temperature test it in the field. Therefore, this change should be regarded as an interim solution to replacing the board with a factory tested one.

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-016 R1  
Model: 1400  
Subject: Testing hard disk drives  
Date: 07/18/84 - Revised 10/1/84, JRC  
From: J Carter

---

### Problem Definition:

Testing hard disk drives in a workstation.

### Solution:

#### USING ONE DRIVE ONLY

(NOTE: the '#' in the following examples shows the UNIX single user prompt and is not a character that is to be typed in.)

To test a hard disk drive in a workstation, perform the following steps:

- 1) Bring the system up into single user mode.
- 2) Perform fsck to check the initial integrity of the drive.
- 3) Enter:  

```
# dd if=/dev/rmd0a of=/dev/null bs=119k conv=sync,noerror count=150 <return>
```
- 4) This test will take about 2 minutes.

(NOTE: *Using md0a instead of rmd0a takes much longer.*)

The errors you don't want to see are those that indicate a read or write error or other access problem, even once, before the process ends. You should also note that the input and output records are equal.

(typical for /dev/rmd0a: 75+0 in, 75+0 out)

- 5) Next, enter:  

```
# dd if=/dev/rmd0c of=/dev/null bs=119k conv=sync,noerror count=670 <return>
```
- 6) This test will take about 5 minutes. You should get similar results as above, only the amount of records read will be much larger.  
(typical for /dev/rmd0c: 355+0 in, 355+0 out)

Number: GEN-016 R1

### USING ONE DRIVE AND THE CARTRIDGE TAPE

You can get better results if there is a tape drive on the system and you change `/dev/null` to `/dev/rmt1`. If this is possible (two tapes will be needed, one for each part of the test), and if you have the time, read the tapes back in to the disk by reversing the 'if' and 'of' conditions (`if=/dev/rmt1, of=/dev/rmd0x`). Doing this will actually verify both read and write on the disk. Of course, if a serious problem should occur you may wipe out the disk (which is what we want to find out about anyway) so be sure you have the disk backed up properly before you do this.

If a problem should occur, don't forget to check out the DSD board and cables first (loose chips in the board, loose cables, etc.).

### USING TWO DISK DRIVES

If you have the luxury of two drives, use `/dev/rmd1a` and `/dev/rmd1c` instead of `/dev/null` for testing with `/dev/rmd0a` and `/dev/rmd0c` respectively. If the customer doesn't have two drives, or if you can't use their second drive for this test, use the test "USING ONE DRIVE" above.

### TESTING A DRIVE THAT CANNOT BE BOOTED

- 1) (NOTE: GEN-025 describes another method of restoring a drive)

Cable your spare drive as drive 1 and boot up to single user:

```
iris> d md(1,)unix1
```

- 2) Perform the 'dd' test on the customer's drive:

```
# dd if=/dev/rmd0a of=/dev/null bs=119k conv=sync,noerror count=150<return>
# dd if=/dev/rmd0c of=/dev/null bs=119k conv=sync,noerror count=670<return>
# sync
```

If there are errors, back up the disk and replace it.

- 3) Perform an fsck on drive 0:

```
# fsck /dev/md0a /dev/rmd0c
# sync
```

Repair the file system as necessary. Mostly what you might expect to happen is that some files will get destroyed and the customer will have lost some data. If the customer has backups of his disk, you will be able to recover any lost data.

### AN ALTERNATE TEST

You can also use the backup procedure given in the Workstation Installation Guide which uses `tar` or `cpio`. Either process will test the disk satisfactorily. This is not a complete test.

Number: GEN-016 R1

---

**RECOVERING A USER DRIVE (drive 0)**

This procedure assumes that the customer has only one drive installed in his system. It also assumes that the second drive to be used has `unix1` on it.

You should have with you a complete set of ribbon cables that will enable you to connect two drives to the DSD board.

- 1) At the DSD board, disconnect the system data cable (P/N: 5000-224-1) and connect your double data cable (P/N: 5000-224-2). Connect the second control cable (P/N: 5000-226-2) to the DSD board. Instead of routing these cables through the system simply bring them over the front of the system to the Mass Storage Compartment.
- 2) Connect the customer's drive to one of the connectors on the new data cable.
- 3) Place the backup drive on a piece of cardboard on top of the tape drive assembly. Connect the backup drive to the other connector on the new data cable and to the second control cable.
- 4) Configure the backup drive as drive 1.
- 5) Power up the system and boot `unix1` from drive 1:  

```
iris> d md(1,)unix1
```
- 6) In single user mode, perform `fsck` on drive 0:  

```
# fsck /dev/rmd0a /dev/rmd0c
```

Do this until the drive comes up clean. If any files are corrupted and need to be destroyed it will be hoped that the customer has a recent backup of his own files.
- 7) Backup customer's disk:  

```
# mkdir /mnt; mount /dev/md0a /mnt; cd /mnt; tar -cv .  
# mount /dev/md0c /mnt/usr; tar -cv usr  
# cd /; umount /dev/md0c; umount /dev/md0a
```
- 8) Perform `fsck` on drive 1:  

```
# fsck /dev/md1a /dev/rmd1c  
# sync
```
- 9) Perform a disk to disk transfer of everything on drive 1 to drive 0:  

```
# dd if=/dev/rmd1a of=/dev/rmd0a bs=119k conv=sync,noerror count=150  
# dd if=/dev/rmd1c of=/dev/rmd0c bs=119k conv=sync,noerror count=670  
# sync
```



Number: GEN-016 R1

---

- 10) Perform another fsck on drive 0.
- 11) Restore the customer's files to drive 0 from tape using the reverse of the same process he used to backup his files on tape.
- 12) Perform another fsck on drive 0.
- 13) Verify the backup was successful by rebooting from drive 0 defaultboot and running some of the demos.
- 14) Remove the backup drive and restore the customer's system to the original configuration.

#### FORMATTING DISKS IN THE FIELD

DON'T BOTHER. This process requires many hours of testing after formatting the disk; checking for bad blocks by performing repeated read and write passes. You especially do not want to use 'fex' to test the disk with unless you are certain of what you are doing and have been properly trained to use it. Some of the operations in this program will wipe the disk clean or write garbage (test data) on the disk and possibly require that the disk be reformatted.

When WS2.0 becomes available there will be adequate documentation to allow anyone to perform this risky operation.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-017  
Model: 1400  
Subject: Tape Drive and Release 1.7  
Date: 8/13/84  
From: J. Carter

---

#### Problem Definition:

Updating a workstation to release 1.7 can result in a disaster if the tape drive is at the wrong revision level.

#### Solution:

Before making the update to 1.7 turn on the power to the 1400 and look at the LED on the tape drive. If it is always on the tape drive must be replaced with the proper revision level. When you request a new tape drive you will get the right one automatically.

After installing the new drive you may proceed with the 1.7 installation.

SILICON GRAPHICS FIELD ENGINEERING

---

**TECHNICAL AIDS**

Number: GEN-019  
Model: 1400  
Subject: Auto reboot in progress..."  
Date: 8/21/84  
From: J. Carter

---

**Problem Definition:**

Some PM2 boards will perform an auto reboot on power up.

**Solution:**

Some versions of the PM2 board respond to unexpected interrupts from the mouse and force an automatic reboot. This condition is acceptable and the board should not be replaced.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-020  
Model: 1400  
Subject: High pitched noise from hard disk.  
Date: 8/21/84  
From: J Carter

---

#### Problem Definition:

Some Vertex disk drives develop a high pitched noise gradually, and this is sometimes followed by a loss in data.

#### Solution:

There is a copper grounding bus on the spindle of the disk drive (seen from the printed circuit board side of the drive) that should be reinforced with a piece of white foam (installed by the manufacturer). Some drives were delivered without this foam and this will allow the copper strap to move away from making contact with the spindle. The strap will then vibrate which causes the high pitched noise. This vibration can induce harmonics in the drive head which could result in loss of data.

To remedy the problem, place a piece of foam (available from the Service Center in Mountain View) and glue it in place. The copper strap may have to be adjusted for tension and to insure proper contact on the spindle.

Do not replace the disk drive for this problem, even if there is a loss of data on the disk. When a loss of data occurs, simply instruct the customer to recover the lost data from their backup. If the drive cannot be backed up (the formatting is destroyed) then it may be necessary to recreate the file system (using mkfs) before restoring the user file system (refer to techaid GEN-021).

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-021 R1  
Model: 1400  
Subject: How to create a disk file system (mkfs).  
Date: 8/21/84 - Revised 12/6/84  
From: J. Carter

---

### Problem Definition:

Sometimes it may be necessary to rebuild a file system because it is impossible to recover a corrupted file system with fsck.

### Solution:

The mkfs command will recreate a file system gracefully. This will also destroy any data that was in the old file system. Do not use this procedure unless the file system has been backed up first.

Mkfs is a command that creates file systems. A file system has to have a name, usually something like /dev/md0a. A file system needs to be allocated to some size (512 byte blocks), and for the root file system it is 17,850 blocks of 512 bytes, or about 8 Megabytes. The user file system, /dev/md0c, has a size of 79,730 blocks of 512 bytes, or about 40 Megabytes. The mkfs command needs two additional arguments that deal with the configuration of the disk drive itself.

After a file system has been created you will want to change the default names on the file system to something the customer prefers. This is done with the labelit command. The three arguments to this command are the name of the device (e.g., /dev/md0a), what you want to call it (root) and who owns it or what it is used for (e.g., SGI; maximum six characters).

### ROOT FILE SYSTEM

```
mkfs /dev/md0a 17850 10 119  
labelit /dev/md0a root name (six characters max)
```

### USR FILE SYSTEM

```
mkfs /dev/rmd0c 79730 10 119  
labelit /dev/rmd0c usr name (six characters max)
```

(Note the use of raw device mode for 'c' and not for 'a'.)

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-022  
Model: 1400  
Subject: Demo cclock hang  
Date: 9/26/84  
From: R. Ludwig

---

#### Problem Definition:

The demo 'cclock' does not work under Unix release 1.7 with over 16 bitplanes; if another demo has not previously been executed that initializes the upper bitplanes.

When 'cclock' is the first demo executed in a workstation with Unix release 1.7 with over 16 bitplanes, the screen is cleared and the program hangs. To exit, type control C, as usual.

#### Solution:

The demo has not been recompiled under WS 1.7. This will be done on the next release of software.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-024 R1  
Model: 1400  
Subject: Tape Cartridges  
Date: 11/7/84  
From: John Carter

---

#### Problem Definition:

Recent tape ejection problems have been encountered on the new Wangtek tape drives and the problem appears to be a combination of tape and drive shortcomings.

#### Solution:

The Wangtek tape drive was tooled for 3M cartridges, which are at one end of the ANSI spec, and not for the the DEI cartridges, which are at the opposite end. DEI and Wangtek have had communications and as a result, DEI has developed a tape more comparable to 3M specs and that should reduce ejection problems.

Presently, we purchase DEI and INMAC tapes (DEI manufactures INMAC tapes). In the future we should purchase only DEI tapes that have a -03 suffix on the part numbers. We should also purge stock, and any blank tapes that are readily available, and return them to the vendor for replacement.

You may encounter ejection problems with older tapes and this should not necessarily mean that the tape drive needs to be replaced. Test the tape drive with a -03 tape cartridge to confirm the problem. If the new tape does not stick, advise the customer to replace his older tapes with the newer ones to avoid the problem.

The other way to tell which tape is which: The old tape cartridges have four screws in the back. The new tape cartridges have three screws in the back.

In at least one case the sticky cartridge tape was attributed to the screws on the underside not being seated completely. You might want to check these screws for tightness before rejecting the cartridge altogether.

Where possible it may be wise to copy the older tapes to new ones as they are made available to further prevent hang-ups.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-022  
Model: 1400  
Subject: Demo cclock hang  
Date: 9/26/84  
From: R. Ludwig

---

#### Problem Definition:

The demo 'cclock' does not work under Unix release 1.7 with over 16 bitplanes; if another demo has not previously been executed that initializes the upper bitplanes.

When 'cclock' is the first demo executed in a workstation with Unix release 1.7 with over 16 bitplanes, the screen is cleared and the program hangs. To exit, type control C, as usual.

#### Solution:

The demo has not been recompiled under WS 1.7. This will be done on the next release of software.



## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-023  
Model: 1400  
Subject: Floating Point Installation  
Date: October 22, 1984  
From: John Carter

---

#### Problem Definition:

How to install the Floating Point Processor Board.

#### Solution:

The Sky Floating Point Board must be properly configured and modified.

The system must have UNIX 1.7 or later operating system.

All of the 1.7 release must be installed (some customers have not yet done this, or thought they didn't need to). In particular, the files /etc/fload and /etc/skyffp.fas must be on the system.

The file /etc/inittab must be modified to initialize the floating point processor. The new line should be the second line in the file. The first three lines of a correct version of inittab are:

```
is:s:initdefault:
fp::sysinit:/etc/fload >/dev/console 2>&1
b1::bootwait:/etc/bcheckrc </dev/console >/dev/console 2>&1 #bootlog
```

When the software is properly installed, one of the following messages will be displayed on the console at initial system boot:

```
fload: skyffp at mbio 40 - initialized and tested
```

(The above message means that a sky board is present and is functioning properly)

```
fload: skyffp not installed
```

(The above message means that the sky board is not present or is not configured or functioning correctly.)

Either of the following messages indicate that the board is not functioning correctly:

```
fload: board test failure
fload: board function test failure
```

The program fload should only be run by a superuser. It is available as /etc/fload.

Code for the Sky board must be compiled and loaded using the -Zf switch for cc, pc or f77.

The following test program will satisfactorily test the Sky board after installation:

Number: GEN-023

---

/\* this little program can be used to test the fp board's system presence. To use, compile this program using this line: "cc -Zg fptest.c" which compiles the program without the the board code. Run a.out and check the time it takes to run (appx 10 sec).

Now, compile the program with this line : "cc -Zf fptest.c". Run a.out again and check the time again. If the board is "there" according to the system, the second program will run 10 times faster (at least).

If you cannot run the program without crashing the system, your inittab file probably does not contain the fp line: [fp::sysinit:/etc/fload > /dev/console 2>&1]. \*/

```
#include "gl.h"
#include "stdio.h"
```

```
main()
{
float a,b;
int k;

    b=1.5;

    for(k=1; k <= 100000; k++)
        a = 50.4 * b;

    printf("a = %f0,a);

}
```

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-024 R1  
Model: 1400  
Subject: Tape Cartridges  
Date: 11/7/84  
From: John Carter

---

#### Problem Definition:

Recent tape ejection problems have been encountered on the new Wangtek tape drives and the problem appears to be a combination of tape and drive shortcomings.

#### Solution:

The Wangtek tape drive was tooled for 3M cartridges, which are at one end of the ANSI spec, and not for the DEI cartridges, which are at the opposite end. DEI and Wangtek have had communications and as a result, DEI has developed a tape more comparable to 3M specs and that should reduce ejection problems.

Presently, we purchase DEI and INMAC tapes (DEI manufactures INMAC tapes). In the future we should purchase only DEI tapes that have a -03 suffix on the part numbers. We should also purge stock, and return blank tapes that are readily available, and return them to the vendor for replacement.

You may encounter ejection problems with older tapes and this should not necessarily mean that the tape drive needs to be replaced. Test the tape drive with a -03 tape cartridge to confirm the problem. If the new tape does not stick, advise the customer to replace his older tapes with the newer ones to avoid the problem.

The other way to tell which tape is which: The old tape cartridges have four screws in the back. The new tape cartridges have three screws in the back.

In at least one case the sticky cartridge tape was attributed to the screws on the underside not being seated completely. You might want to check these screws for tightness before rejecting the cartridge altogether.

Where possible it may be wise to copy the older tapes to new ones as they are made available to further prevent hang-ups.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-025  
Model: 1400, WS1.7  
Subject: Bootable Tapes  
Date: December 5, 1984  
From: John Carter

---

#### Problem Definition:

Corrupted root file systems can only be restored from a bootable tape or from a second disk drive.

#### Solution:

A special tape is provided to allow restoring a root file system from tape, even when you can no longer access the disk.

This tape was built from a special operating system that you do not have access to yet. When WS2.0 is available you will be able to create your own bootable tapes. The tape is composed of three files.

File 1 contains fex4.4 and memtest. These can be booted directed from the tape (in the IRIS PROM monitor mode type: *tb name*). File 1 is written on the tape in cpio format.

File 2 contains a production level WS1.7 root file system. This file can only be accessed by fex4.4 and is only for the purposes of rebuilding the hard disk root file system. File 2 is written on the tape in dd format with a non-rewind version of the tape driver.

File 3 contains a production level usr file system. This file can only be accessed by a WS2.0 operating system or the special operating system that created this tape; it can be accessed by fex4.4 but it would only crash your usr file system to do so. Since you are not provided with the proper operating system it seems worthless to even put file three on this tape; nevertheless, here it is. It was expected that if the customer wasted his usr file system he should have a back up and not need this tape for that purpose. This tape is only intended to give you the capability to recover the root file system with production level software. There is no way to recover the root file system directly from a user's backup tape, unless he had backed up the root file system alone on one tape. More on that later. File 3 is written on the tape in cpio format with a non-rewind version of the tape drive.

To position the tape to files 2 or 3 the *smt* command is used. What prevents you from doing this of course is that the *smt* command doesn't work because the UNIX kernel doesn't support it properly. Fex4.4 will also position the tape to any file, but it only transfers files that have been written in the dd format.

To rebuild a disk from this tape you first need to boot fex4.4 from the tape. The steps are as follows (user inputs are shown as constant spacing typewriter type):

1. Boot fex4.4 from the tape -  
iris> tb fex4.4<return>

Number: GEN-025

- 
2. This fex has a new utility called 'Tape'. This utility will copy a file from the tape to any file system on the hard disk. The utility is executed by just typing a 't' (lower case, no return). The utility asks for the tape file to copy from (default=2), the disk drive unit to access (default=0), and the file system to copy to (default=a). The fex prompt is 'Fex 4.4>'.  
Type: t  
The response is:  
Fex 4.4> Tape to Disk copy  
Tape file (2)?
  3. Type: <return>  
The response is:  
Unit (0)?
  4. Type: <return>  
The response is:  
File System (a)?
  5. Type: <return>  
The response is:  
Copying 8925K in 119 blk chunks from tape file 2 to md0a  
Type 'go<return>' to begin...
  6. Type: go<return>  
The response is:  
Copy started...  
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150  
Tape to Disk Copy Complete  
Fex 4.4>
  7. Quit Fex: q  
The response is:  
Fex 4.4> Quit ..confirm quit with "y":  
The system will reset back to the IRIS PROM monitor.
  8. Boot the system to single user mode.  
iris> d<return>
  9. When the UNIX prompt appears, perform fsck:  
IRIS 1# fsck /dev/md0a<return>

Number: GEN-025

10. Insert the user's backup tape and restore it on the system. If the backup tape is both root and user file systems you will first have to mount the usr file system. It may be best to let the user restore his own system at this point.

If the usr file system is also damaged and the user cannot access it at all, it will be necessary to perform mkfs on it before attempting to restore it. The procedure for this is in another article, Techaid GEN-021. If mkfs fails, you will have to reformat the disk with fex4.4 and this is an overnight (24 hour) process because the exercise requires that much time to find all bad blocks.

The following files on the new root file system are not exactly the same as the production release, I have removed the gross errors from them:

```
/.login  
/.cshrc  
/etc/rc
```

If you had the proper level operating system and wanted to copy the third file from the tape to the usr file system on the disk you would first reboot the system to single user mode. Next you would execute the following commands:

```
mount /dev/md0c /usr  
cd /usr  
smt fsf 2  
cpio -iah2dum
```

If the customer has a root file system backup on a single tape and it is in the 'dd' format (not tar or cpio), then fex4.4 could restore that backup directly. After you have booted fex4.4 from the tape, insert the customer's tape. At step 3 where it asks to confirm the tape file (2), enter 1. This has not been tested, but like all other good theories it should work.

Remember, this tape loads the production release software, WS1.7, to the root file system using fex4.4. You will not be able to make your own tapes until you get WS2.0. Guard this tape carefully and don't take the safety off.

Good luck.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-026  
Model: 2400 GL2-W2.1  
Subject: New procedure for booting from floppies  
Date: 2/7/85  
From: Paul Mlyniec

---

#### Problem Definition:

New PROMs expect different command for booting from floppy

#### Solution:

Use the command: b mf0a:[filename]

The filename is optional.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-027  
Model: 2000/2200/2400  
Subject: GF2 Boards  
Date: 3/6/85  
From: R. Edgar

---

#### Problem Definition:

GF2 cards incorporate the GA chip. The current batch of chips have a voltage problem resulting in many field failures attributed to the setting of the power supply output. The failures can be circumvented by increasing the voltage output. The clue to look for is no display at the monitor. Manufacturing is setting the voltages to compensate for this, however, some number of upgrade and new units were ship before the problem was discovered.

#### Solution:

Measure across +S1 and -S1 of the supply with your DVM and adjust V1 to +5.17 vdc. Tag the supply to reflect this condition for future reference. Whenever replacing GF2 cards in future, inspect the PCB for a label or marking indicating Deviation Authorization #002 present. This will signify the card is populated with GA's requiring this change. Manufacturing will also mark the configuration label to reflect this deviation on all future units with these GA's.



## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-028  
Model: 1000 & 1200  
Subject: Multibus Memory Board Modification  
Date: 3-28-85  
From: Ron Ludwig

---

#### Problem Definition:

If performing an upgrade on a 1000 or 1200 terminal from GL1 to GL2, and the unit has a 1 Meg multibus memory board installed; then the memory board (5001-173-08) must be modified.

#### Solution:

The modification consists of resoldering two pins on the P2 bus connection of the board, pins P2-55 and P2-56.

This must be done because in a GL1 unit (1000, 1200) the two most significant memory addressing bits were not used and were actually disabled to eliminate noise; but in a GL2 unit (2000, 2200) these two address lines are now being used to address more memory and therefore must be reconnected.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-029  
Model: GL1-W2.1 and GL2-W2.1  
Subject: Tape drive errors  
Date: March 29, 1985  
From: John Carter

---

#### Problem Definition:

On both GL1 and GL2 systems, version 2.1 and up, using the tape drive can cause problems for casual users.

If the access LED on the tape drive stays lit after ANY attempt to use the tape (doesn't always happen), and the tape is accessed again after that, the system may hang with a tape error.

#### Solution:

The workaround is:

1. if the LED stays lit after a tape access, eject the tape.  
This turns out the LED.
2. type 'smt rew'.
3. an error message will be displayed and it may be ignored.
4. the next tape access should be okay.

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-030 R1  
Model: 1400 & 1500 & 2400 & 2500  
Subject: Disk Drive Restoration Procedure  
Date: October 10, 1985  
From: Ron Ludwig

---

### Problem Definition:

When fsck fails to correct file system errors and the disk errors continue to occur, then its time to reformat and restore the disk drive.

### Solution:

## DISK DRIVE RESTORATION PROCEDURE

### READ IN MDFEX FROM TAPE:

1. Insert bootable tape.
2. Enter 'b' for command list.
3. Enter proper command to boot from tape.  
Example: 'tb /stand/mdfex' or 'b mt:/stand/mdfex'

NOTE: Refer to Software Distribution Manual (Appendix B) for instruction on mdfex.  
Software releases below 3.0 use 'fex' instead of '/stand/mdfex'.

### SELECT DISK DRIVE ENVIRONMENT:

4. Type 's' to set miscellaneous variables.
5. Type 'u' to select drive.
6. Enter '0' or '1' to select drive to be restored.  
(The following step is required for 'fex' and not for 'mdfex':)
7. Type 'i' to initialize selected drive.
8. Type 's' to set miscellaneous variables (again).
9. Type 'w' to select writelock toggle function.
10. Type 'off <return>' to turn writelock off.

### ENTER BAD BLOCK LIST FROM MAP ERROR TABLE ON TOP OF DRIVE:

NOTE: This Map Error Table is a list of the already known bad blocks.

11. Type 'b' to enter bad block edit mode.
12. Type 'p' to print current bad block list and compare with  
list on disk drive label located on top of disk drive -- you must  
remove the access cover over the disk drive area to read that label.  
(IF the list in the file MATCHES the list on the label DO NOT do next 5 steps)
13. Type 'c' to clear bad block list.
14. Type 'y' to confirm.
15. Type 'a' to add bad blocks.
16. Now enter each entry, from the ERROR MAP on top of drive, in the

Number: GEN-030 R1

---

format of 'cylinder/head' (i.e. 31/0 or 921/6).

17. Press RETURN to end.

18. Type 'q' to quit bad block edit mode.

**FORMAT DISK:**

19. Type 'f' to format disk.

20. Enter 'go' to start.

NOTE: Formatting takes 4 minutes (8 for 'fex').

**EXERCISE DISK:**

21. Type 'e' to enter exercise mode.

22. Type 'c' for complete write/read test.

23. Press RETURN to repeat test forever.

24. Press RETURN for no alternate units.

NOTE: One test cycle takes 8 minutes.

This test should be run for at least 12 hours  
and preferably 24 hours for a thorough check  
of all blocks; but do what time allows.

(To terminate test:)

25. Press a to abort test (Press ESC when using 'fex').

26. Type 'e' for error display.

27. Type 'd' to display errors.

28. Now record list of bad blocks, if any, on paper.

29. Type 'q' to quit exercise mode.

**ADD NEW BAD BLOCKS TO BAD BLOCK LIST:**

30. Type 'b' to enter bad block edit mode.

31. Type 'a' to add bad blocks.

32. Now enter each entry from your list recorded on paper.

33. Press RETURN to end.

34. Type 'q' to quit bad block edit mode.

**REFORMAT DISK TO SET ALTERNATE TRACKS:**

35. Type 'f' to format disk.

36. Enter 'go' to start.

NOTE: Formatting takes 4 minutes.

Number: GEN-030 R1

---

**COPY ROOT FILE SYSTEM FROM TAPE TO DISK:**

37. Type 't' for tape to disk utility.
38. Press <RETURN> to copy from tape file (2). (root file system)
39. Press <RETURN> to copy to selected unit.  
(If using 'fex', next step is simply <RETURN> for file system (a) ).
40. Press a <RETURN> to copy to file system (a). (root file system)
41. Enter 'go' to begin copying.

NOTE: Copying takes 4 minutes.

42. Type 'q' to quit MDFEX.
43. Type 'y' to confirm.

**COPY /USR FILE SYSTEM FROM TAPE TO DISK:**

44. Boot UNIX from disk.  
Example: 'd' or 'b'

NOTE: Refer to Software Distribution Manual for instruction on how to load UNIX.

45. Enter 'newfs md0c' to make /usr file system on unit 0  
or enter 'newfs md1c' to make /usr file system on unit 1.

NOTE: The command 'newfs md0c' performs a 'mkfs' on the raw device for the /usr file system of unit 0.

46. Enter 'y' to confirm.

(NOTE: What follows is where most operator errors occur. Be certain that you mount the /usr file system before proceeding with the restore process.)

47. Enter 'mount /dev/md0c /usr' to mount /usr.
48. Enter 'cd /usr' to change directory to /usr.
49. Enter 'smt rew' to rewind tape.

(The following two steps can be combined into one: smt fsf 2)

50. Enter 'smt fsf' to position tape to file #2.
51. Enter 'smt fsf' to position tape to file #3.
52. Enter 'cpio -iahdv2' to copy /usr from tape to disk.

**RESTORE DISK DRIVE LABELS:**

53. Enter the 'sgilabel' command to label the disk drive.  
Usage: sgilabel [-n name] [-s serial#] dv#  
Example: sgilabel -n "Release GL1-W2.1" -s 14596 md0
54. Use the 'labelit' command to label each file system.  
Usage: labelit /dev/r???? [ fsname volume [-n] ]

NOTE: The question marks designate the file system of which to relabel:  
and the -n option skips the label check before relabeling.

- a. Label usr file system first.  
Example: labelit /dev/rmd0a usr W2.1
- b. Label root file system last.

Number: GEN-030 R1

---

Example: `labelit /dev/rmd0c root #2.1`

55. Type `'reboot -q'` to reboot the system.

NOTE: For software release below 3.0, press the reset button instead.

Do NOT enter `'reboot -q'` or else the root file system label will not be updated.

56. Boot up UNIX again up to multi-user level and ensure proper operation.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: gen-031 R1  
Model: All Workstations  
Subject: Hard Disk Bad Blocks  
Date: October 10, 1985  
From: John Carter

---

#### Problem Definition:

Every now and then one of the systems will develop a bad block. The error message seen will indicate a read or write hard error to the disk. The error message shows that there is a bad block and further gives the location as xxx/xx/x (cylinder/head/sector) near the end of the error message. The cylinder and head information is all that is required to map out the bad block.

#### Solution:

**NOTE:** This is a companion Tech Aid to GEN-030. Please refer to that Tech Aid for details of operations referred to here.

1. If the bad block reported is on track 0 or 1 and the reported errors do not appear to be random (all over the disk or of different types), forget the rest of this and swap out the disk drive. If the errors are on track 0 or 1 and you also have a recent history of random errors, look at paragraphs 6 and 7 concerning random errors.
2. When a bad block is encountered it is mapped out with 'mdfex' (used to be called 'fex' in software releases below 2.3). The utility used in mdfex is 'mapbad'. The process necessary to update the bad block list for the first time ever on a any system is to:
  - a. back up the system onto a bootable tape, if not already done,
  - b. update the bad block list as described in GEN-030,
  - c. and then restor the disk from the bootable tape.
3. Once the bad block list has been corrected and new bad blocks are found and need to be added to the list it is only necessary to use mapbad in mdfex.
4. Backing up the disk to tape is necessary because when we go through the process of updating the bad block list there is a high probability of losing data on the disk. Here's why: When the system disk was created it was first formatted with the correct bad block list. After that the software was loaded by copying the entire disk byte for byte from a master. This put the wrong bad block list on the disk and it is not updated by SGI manufacturing before sending it out. (NOTE: This may not be happening on the latest releases of systems.) As new data is stored on the disk a bad block error may occur and it just may be one of those real bad blocks already known on the label but not in the list. Also, blocks that are in the list which aren't real bad blocks will be routed to alternate tracks, which is a waste of space. In any event, if one were to simply update the bad blocks with mapbad they would have to enter all the bad blocks from the label and leave in the bad blocks from the master in order to maintain software integrity. One way to recover the space and configure the system is to back up the system, correct and update the bad block list, and then restore the system.
5. Backing up the system is covered quite well in the Workstation Installation Guide. A sure-fire command

Number: gen-031 R1

that will backup the entire system (as long as the total disk space is less than 40 Megabytes) is:

```
mkboot /usr
```

If this errors, try another tape.

6. After correcting and updating the bad block list, if you do get another error on this same block, or if you get many errors of bad blocks randomly on the disk, the easiest module to check out first is the DSD board and/or cables. If the errors persist after replacing the DSD and/or cables then it may be necessary to replace the disk, but before doing this I would recommend formatting the disk (sometimes this really helps, and should be done if the customer is out of warranty and has no maintenance contract.)

7. It has been found that some random errors from a disk drive can be related to dirty contacts anywhere. This is corrected by reseating all plug-in chips on the DSD board and reseating all cable and an PC boards. If the system is under warranty or maintenance contract this procedure is to be done only by an SGI Field Service Representative.

8. Before formatting a disk I suggest that the 'exercise' test be done for 12 hours (vendor recommended duration) to find any other bad blocks that may be lurking around. In mdxfex exercise, use the complete test ('c').

9. After exercising a disk it is necessary to take the new bad blocks found and add them to the bad block lists and then the disk must be formatted. When a disk is formatted it uses the bad block list in the label to allocate alternate blocks and flag bad ones.

10. So what causes bad blocks to suddenly appear? Mechanical problems, defects in the oxide surface of the disk, etc. . .

11. The mechanical hazards can be user induced. The system can only withstand a 2G shock while turned on and up to 20G's when power is off. This means that a bump of the knee with power on could cause a head slap which would result in a bad block. Other physical defects are caused by normal deterioration of the surface oxide or foreign particles entering the disk area and collecting on the disk surface.

12. What appears to be bad data on the disk may be caused by mishandling (no anti-static mat if disk is being removed for any reason) or by the absence of the anti-static ground strap on the spindle. The latter is one of much discussion lately as the vendor claims it is no longer necessary and is intending to remove same in future assemblies. It is even recommended to pull it away from contact with the spindle if this is the cause of a high frequency squeal (when the strap vibrates against the spindle). This strap was designed to prevent static build-up between the disk and head. A static discharge between head and disk could cause a pin hole in the oxide surface and result in a bad block. The least it would do is alter the data at that point and possibly result in a bad block.



# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: GEN-032 R2  
Model: GL2  
Subject: New IP1 board and PM2M1  
Date: Sept 16, 1985  
From: John Carter

---

### Problem Definition:

The new IP1 boards are now being shipped to replace the old PM2 boards. Unfortunately the old PM2M boards will not work with the IP1 boards. The new PM2M1 boards will work with both PM2 and IP1.

Obviously spares and repairs are going to be affected by this change so here is how to handle these situations.

### Solution:

#### PM2M1

All PM2M memory boards should be returned to the factory for update to a PM2M1 Rev D. The PM2M boards are Rev A, B and C. The part numbers on both boards are the same.

The PM2M1 board has a compatibility header on it at location U18K that allows this board to be used with either a PM2 or IP1. There are two types of headers, one for the IP1 and one for the PM2. The part number of this header is 5000-243-xx, where xx is as follows:

Type	Header Suffix	Pins Jumpered:
IP1	-01	1-10, 6-8-11, 9-13
PM2	-02	1-6, 8-13, 9-10-11

Attached is a diagram of the headers showing how to build one if you needed to. As of this writing it has been observed that there is no identification printed on the header, so you will need the accompanying diagram to be able to identify which header is on the PM2M1.

#### IP1

The IP1 board has 2 Megabyte of memory on it. If the customer has a 2.5 Meg system he will wind up with a 3 Meg system if an IP1 board is plugged in. This means actually removing the older PM2M1's and installing the newer PM2M1's to provide AT LEAST the amount of memory the customer originally ordered. Make sure that the configuration sheet is updated.

Some 2400 upgrades are being sent out with IP1's and you may discover that they did not send the PM2M1 memory boards with it. This means you will have to either use your PM2M1 spares or pull the IP1 and install your spare PM2 if you have one. If you can't configure the system to get it up with at least the amount of memory that the customer ordered, or you don't have the right mix of boards to do anything, let us know immediately. The right time to find this out is before you go out to do an upgrade. This may mean having the customer verify what we shipped (if we can't tell you what we shipped).

Number: GEN-032 R2

The configuration PROM on the IP1 for the primary/secondary monitor types is at location U136. Only part numbers are shown instead of text so the following will be useful in identifying what PROM you have in the board:

Primary/Secondary	IP1 PROM
NI/I	5000-402-06
NI/RS170	5000-402-07
I/RS170	5000-402-08
NI/EURO	5000-402-09
I/EURO	5000-402-10

A four position memory configuration switch is on the edge of the IP1 (like the PM2M1) and sets the on-board memory at one of four starting ranges. The values of the four switch positions are the same as for the PM2M1, meaning that they are ordered hexadecimal starting with the least significant bit at switch position 4. The following table shows the settings for the start address of each range.

switch	start @
None on	0 Meg
2 on	4 Meg
1 on	8 Meg
1 & 2 on	12 Meg

Like the 2 Meg PM2, no other memory can be configured to occupy any part of the same range as what the IP1 is configured for.

#### DC4

The DC4 board is a beastie all to itself. Here is the compatibility information:

	P/N	NI/RS170A	NI/I	I/RS170A	NI/EURO	I/EURO
Assy						
DC4	5000-090-xx	-02	-03	-04	-05	-06
PROMs						
U16D	5000-395-xx	-02	-01	-03	-04	-05
U17D	5000-396-xx	-02	-01	-03	-04	-05
U18D	5000-397-xx	-02	-01	-03	-04	-05
U14B	5000-398-xx	-02	-01	-03	-04	-05
U15E	5000-399-xx	-02	-03	-04	-05	-06
U14F	5000-400-xx	-02	-03	-04	-05	-06

The DC4 board I/NI option (5000-090-01) has been retired.

#### ADMINISTRATION

When handling repairs on systems, we try to make a real effort to keep the same kind of boards in the system when you put a spare in it. It's all right to leave a half meg more in the system than what was originally in it if you have to swap out the processor board, but for tracking purposes we need to know how you left the

SILICON GRAPHICS FIELD ENGINEERING

---

TECHNICAL AIDS

(Cont'd)

Page 3

Number: GEN-032 R2

---

system configured.

If all you have is an IP1 spare and the system has a PM2 with the older PM2M's then you seem to be forced to replace the PM2 with an IP1 and the PM2M's with PM2M1's if you have them. Naturally, if you don't have the spares to do this you have to let the system be until you have what you need. Knowing this in advance will help us plan not to let this happen.

A concerted effort is being made to prevent any more scambles like this. Your assistance in identifying the need for spares in your area will certainly help. Along this line, when you get this Techaid, please send Jackson Chun an inventory of all the spares you have. The inventory should show part number, Rev level, and location of board (in transit to us or temporarily at a customer site) and any other identifying aspects of the board (type of RAM chips, PROMS, etc) as comments:

QTY	P/N	Rev	location	remarks
-----	-----	-----	----------	---------

(If you put a spare into a customer system and sent his bad board to us then this is only one board, the one you sent to us. If you put a spare into a customer system as a loaner (not a replacement) and requested a new one and haven't gotten it yet then that is counted as one board, the one in the customer system.)

Thank you.

SILICON GRAPHICS FIELD ENGINEERING

---

**TECHNICAL AIDS**

Number: GEN-033  
Model: ALL  
Subject: Lines at bottom of Monitor  
Date: Dec. 16, 1985  
From: John Carter

---

**Problem Definition:**

Apparently, if the monitor is plugged into the switched AC, which it usually is, something happens with "convergence circuit" on power up, sometimes causing a line to be displayed across the bottom of the screen.

**Solution:**

Powering monitor down and up again solves problem, or move monitor power to wall outlet.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-034  
Model: ALL  
Subject: Tape/Floppy head cleaning  
Date: Dec. 18, 1985  
From: John Carter

---

#### Problem Definition:

Cipher would like to take this opportunity to reiterate the recommended cleaning procedures.

#### Solution:

The ONLY cleaning solvent considered safe for use on ANY magnetic tape (or floppy) drive is TRICHLOR-TRIFLOUROETHANE (also known as Freon T F).

It is noted that blends of freon and alcohol leave a residue on the roller guides and the take-up hub. The buildup of residue on the take-up hub can cause slippage and possible auto loading failure (which is now displayed as an error code "9" on the CacheTape).

The use of cotton swabs is also not recommended. The cotton swabs may leave cotton fibers in the read/write head gaps causing oxide buildup. The recommended swab is a plastic foam such as found in the Cipher Cleaning Kit (soon to be available as a SGI Cleaning Kit). Cipher strongly recommends you inspect your cleaning materials and eliminate those things which contain alcohol and/or chemicals and material not recommended. SGI seconds this.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-035  
Model: 2500  
Subject: Power Receptacles  
Date: January 14, 1986  
From: John Carter

---

#### Problem Definition:

Customers are asking what type of receptacle is needed to power their additional Eagle drives, and what are the current requirements for each.

#### Solution:

From the Hubbell catalog (#35R-1) the name for our 20A 125V plug is STRAIGHT BLADE. This term also applies to several other plugs of different physical characteristics, all having straight blades or angled blades, so long as they are not curved blades. Now the important thing to note is the NEMA reference for each type plug. For our plug it is 5-20P. The NEMA reference for the corresponding receptacle is 5-20R.

We are using a twist-lock receptacle in the 2500 to provide additional input power for a second Winchester drive and/or auxiliary equipment. The NEMA reference for this receptacle is L5-20R. When we provide a second Winchester disk drive we also provide a double male power cord. One end of the power cord has a twist-lock plug and the other has a straight blade plug. The twist-lock plug goes into the twist-lock receptacle on our power panel and the straight blade plug, another NEMA reference 5-20P, goes into the customers wall receptacle. The only thing not specifically mentioned anywhere is that when the second drive and/or auxiliary equipment is being used the customer must provide a separate outlet for each power cord, each outlet on its own 20A breaker. The second drive and/or auxiliary equipment gets plugged in to receptacles internal to our system on the one power panel.

If a customer uses a double receptacle for plugging everything into at his wall then each boss (tradename term for receptacle) should be on a separate breaker. This is where an isolated double receptacle may be required depending on the customers need for grounding, like in hospitals. The only thing isolated is the ground between the two receptacles.

NEMA stands for National Electrical Manufacturers Association. The NEMA 5-20 and NEMA L5-20 specify a 20A 125V A.C. or D.C. power system. This means it should not ever exceed these ratings.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: GEN-036  
Model: 2400/2400T  
Subject: Tape and Disk errors after upgrade  
Date: 5/9/86  
From: Dave Williams

---

#### Problem Definition:

Numerous problems with disk and tape after an upgrade. Tape doesn't read, disks get uncorrectable errors. Also blank CRT upon power up.

#### Solution:

If you experience the above problems check the V2 voltage at the power supply and at the drives. The 2400 Turbo upgrade document calls for V2 to be set to 12.10 volts at the power supply with the load board installed. The problem is with the +12 volts when using a fully (or nearly so) configured system. The 2 170 megabyte disks along with the tape drive put a load on the power supply that is close to the limit of the supply causing an excessive voltage drop and noise. A simple but not permanent fix is to raise the spec for the +12v to 12.14 volts with the load board as measured at the supply. A permanent fix is being investigated by Engineering.

NOTE: This may not be your problem, be aware that it can also be a bad tape, a bad tape drive, a bad or incorrectly formatted disk drive or a controller problem (and maybe others).

Thanks to Jerry Sood and Greg Eskridge for their invaluable assistance with this tip.

SILICON GRAPHICS PRODUCT SUPPORT ENGINEERING

---

**TECHNICAL AIDS**

Number: GEN-037  
Model: 2400 and 2500  
Subject: Configuring a Second Disk as One File System  
Date: 6/18/86  
To: Prod. Sup., Field Eng., System Eng., Tech. mktg, OEM, VAR  
From: Dave Williams  
Approved:

---

**Problem Definition:**

In version 2 of the IRIS 2000 Workstation guide on page 7-12 there is an error. The section concerns configuring a second disk as one file system. Under #3 the syntax of the newfs command is incorrect.

**Solution:**

Remove the references to /dev/ for the newfs commands, ie;

newfs /dev/md1d

should be

newfs md1d



TECHNICAL AIDS

INDEX - INTERFACE

Updated: 8/4/86

---

I/O-001	Configuring DSD 5215 Disk Controller
I/O-002	Serial Port (DTR)
I/O-003	DSD Disk Controller cable chart
I/O-004 R1	DSD board jumper locations
I/O-005	Serial Interface, host (R1400)
I/O-006	Vertex V100 Drive Configuration Dipswitch
I/O-007	Vertex V100 Noise Problem
I/O-008 R1	Modems, 1400's & cu
I/O-009 R4	Connecting a printer to a 1400.
I/O-010	Unable to access tape drive
I/O-011	Bad Tape Recovery
I/O-012 R1	ttyd1 gets assigned as syscon
I/O-013 R1	Connecting a printer to a 2400.
I/O-014	XNS between two workstations
I/O-015	NTSC problems
I/O-016	Interphase SMD Controller upgrade before install of 3.4
I/O-017	Euro Video Display Problem
I/O-018 R1	IEEE-488 programming tips
I/O-019	RS-232 devices are hanging the system

SILICON GRAPHICS FIELD ENGINEERING

---

**TECHNICAL AIDS**

Number: I/O-001 (updated 7/16/84 JRC)  
 Model: All  
 Subject: Configuring DSD 5215 Disk Controller  
 Date: 1/18/84  
 From: Roger Edgar

---

**Problem Definition:**

The Data Systems Design 5215 Multibus Disk Control card (SGI P/N 9400001) is designed to handle up to two 5 1/4 inch Winchester style disk drives, two Floppy disk drives, and one streaming style Tape drive. Due to the boards flexible nature in terms of drive types and bus characteristics it must be configured for proper operation via jumpers on the board.

**Solution:**

Use the following configuration chart to insure correct jumpering of options:

Jumper	Function	Pins	In	Out	Options	Remarks
W-2	DSD use only	1-2	*		None	Leave as set.
W-3	Data Transfer Size	1-2	*		Selects 16 bit data bus.	Out selects 8 bit data bus.
W-4	DSD use only	1-2	*		None	Leave as set.
		3-4	*		None	Leave as set.
W-5	Bus Arbitration Mode	1-2		*	In= yield to higher priority mode.	User option. Except for override mode, only one jumper at a time is 'In'.
		2-3		*	In= yield to any request mode.	
		2-4		*	In= single transfer mode	
		5-6		*	1-2 and 5-6 In = override mode.	
W-6	Drive class and diagnostic selection	0	*		Floppy drive class	0 & 1 in = class 0. (Tandon 101)
		1	*		Floppy drive class	" " " " " " " " "
		2	*		Winchester drive class.	2 in with 3 & 4 out = class 6. (ATASI)
		3		*	Winchester drive class.	" " " " " " " " "
		4		*	Winchester drive class.	" " " " " " " " "
		5	*	HyperDiagnostics.	Selected when out.	
W-8	Priority Continuity	1-2	*		Serial bus priority scheme selected.	Out = parallel priority scheme selected.

SILICON GRAPHICS FIELD ENGINEERING

TECHNICAL AIDS

(Cont'd)

Page 2

Number: I/O-001 (updated 7/16/84 JRC)

Jumper	Function	Pins	In	Out	Options	Remarks					
W-10	Interrupt priority level selection	C-0	*		Priority level 0 (highest)	User option. See Note 1. Wrap C-1 for workstation only.					
		C-1	*		Priority level 1						
		C-2	*		Priority level 2						
		C-3	*		Priority level 3	Remove wire wrap for all SGI models.					
		C-4	*		Priority level 4						
		C-5	*		Priority level 5 (DSD factory setting)						
		C-6	*		Priority level 6						
C-7	*		Priority level 7 (lowest)								
W-13	DSD use only	1-2	*		None	Leave as set					
W-14	8 or 16 bit I/O Address select	1-2		*	16 bit I/O addressing selected.	User option. 1-2 in with 2-3 out selects 8 bit I/O addressing. See Note 2.					
		2-3	*		Reverse selects 8 bit.						
W-15	DSD use only	A	*		None	Leave as set.					
		B		*							
W-16	DSD use only			*	5215-01 Version	Leave as set.					
				*	5215-02 Version.						
Jumper	Function	Pins	1	0	Hex	Jumper	Function	Pins	1	0	Hex
W-9	Least significant byte of I/O port address (See Note 3.)	0		*	LSB	W-7	Most significant byte of I/O port address	8		*	
		1		*	4			9		*	0
		2	*					10		*	
		3		*				11		*	
		4		*				12		*	
		5		*	0			13		*	0
		6		*				14		*	
7		*		15		*	MSB				

Notes:

1. The board is received from the manufacturer with C-5 wire wrapped. This wrap should be removed for all IRIS models. The point 'C' should be left open for IRIS Terminals and C-1 should be wire wrapped for IRIS Workstations.
2. The silkscreen numbers on the board for W-14 read 16, 8 and no marking for the three pins, while the jumper list calls for pins 1, 2 and 3. Hold the board such that the I/O fingers are toward you and the pins would be numbered 3, 2 and 1 reading left to right.
3. Jumpers W-9 and W-7 together make up the 16 bit I/O port address which SGI sets to a Hex '0004'.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: I/O-002  
Model: IRIS 1000/1200  
Subject: Serial Port (DTR)  
Date: 1/18/84  
From: Roger Edgar

---

#### Problem Definition:

RS232 port signal DTR (Data Terminal Ready) is not present at the IRIS backpanel Port 2 connector. Some host software may require DTR asserted for terminal operation.

#### Solution:

Make "DTR" available at the port connection by adding a jumper wire to the PM1 bd at location H2A pins 7 to 8. An ECR has been approved and future boards will have this jumper factory installed.

SILICON GRAPHICS FIELD ENGINEERING

---

**TECHNICAL AIDS**

Number: I/O-003  
Model: ALL IRIS  
Subject: DSD 5215 Disk Controller Cable Chart  
Date: 1/25/84  
From: Roger Edgar

---

**Problem Definition:**

Identifying the proper connector to attach the Disk and/or Tape data and control cables.

**Solution:**

Use the following information to determine the correct cabling configuration:

Device	Cable	# Pins	Bd. Connector
Tape	Data & control	50	J1
Winchester Disk	Control	34	J2A
Floppy Disk	Data & Control	34	J2B
Winchester Disk 1	Data	20	J3
Winchester Disk 2	Data	20	J4

**Note:**

J3 and J4 are interchangeable.

SILICON GRAPHICS FIELD ENGINEERING

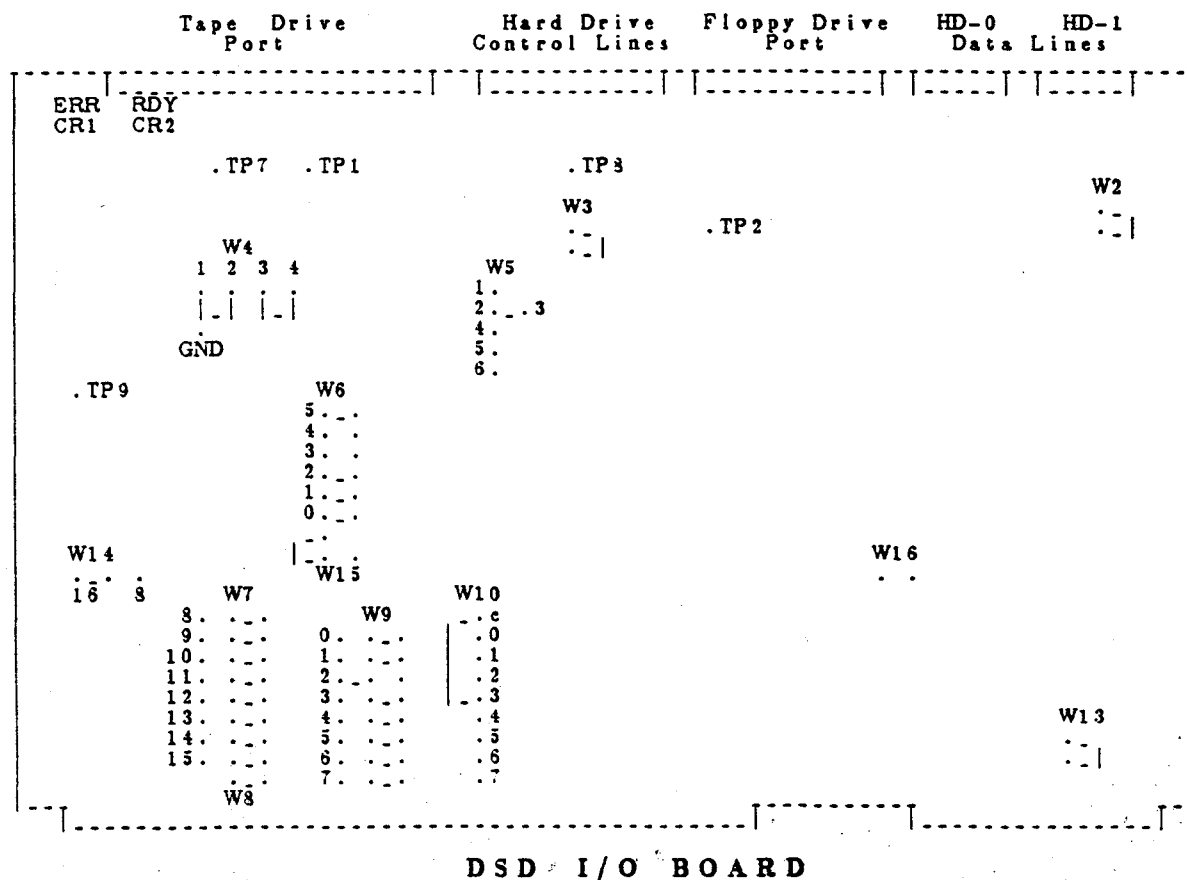
TECHNICAL AIDS

Number: I/O-004 R1  
 Model: All IRIS  
 Subject: DSD Board Jumper Locations  
 Date: 2/17/84 (Revised 10/4/84; RL)  
 From: Ron Ludwig

Problem Definition:

There is no pictorial description of jumper locations for the DSD board.

Solution:



NOTICE: There should a wire connecting W10-c to W10-3 for an Iris workstation (1400) and no wires on W10 for an Iris terminal (1000, 1200).

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: IO-005  
Model: 1400  
Subject: Serial interface, host  
Date: June 6, 1984  
From: John Carter

---

### Problem Definition:

- 1.) The workstation serial interface is supported by engineering for baud rates up to 1200. Above 1200 Baud the UNIX I/O driver has difficulty processing and receiving data fast enough to prevent losing some data. Of course the faster the data comes in, or the larger the chunks of data received at the high baud rates, the more data will be lost.
- 2.) Data transfers to and from a host computer using 'take' and 'put' commands will not work unless all files are properly set.

### Solution:

- 1.) The VAX UNIX system has a command that will cause each record (newline to newline) sent to pause after sending the record. This command is:

```
#stty nl cr2
```

where the argument cr2 is one selected to better support 9600 Baud and the argument nl specifies that only a carriage return character marks the end of a record or line. See the users manual for more details on stty.

- 2.) The files that must be set properly are:

```
/dev/ttydn (where n = 1, 2, or 3)  
/etc/inittab  
/usr/lib/uucp/L-devices
```

The number for /dev/ttydn (the n) should coincide with the port that the host is connected to. The port assignments are:

```
Port 1 = /dev/ttyd0 (IRIS keyboard)  
Port 2 = /dev/ttyd1 (usually a dumb terminal for system administration)  
Port 3 = /dev/ttyd2 (an ASCII terminal or host port)  
Port 4 = /dev/ttyd3 (a printer, ASCII terminal, or host port)
```

To make any of the changes mentioned here you must be logged in as a superuser.

Assuming you are using port 3 for host communications you should change the read/write mode of /dev/ttyd2 for read/write at all levels of access as follows:

```
#chmod 666 /dev/ttyd2
```

Number: IO-005

---

Next, the file `/etc/inittab` must be edited. There is one line that starts with `d2` which must look as follows:

```
d2:x:respawn:/etc/getty    ttyd2    co_9600
```

If you want to start communications at other than 9600 Baud then change the 9600 on that line to the desired baud rate.

Last, the file `/usr/lib/uucp/L-devices` must have an entry that will correspond to the baud rate that you are using in `/etc/inittab`, as follows:

```
DIR    ttyd2    ttyd2    9600
```

You may notice that this file has only two entries in it as delivered from SGI, both of which deal with `ttyd1` only.

If the port you use for host communication is changed to talk to an ASCII terminal then you must edit `/etc/inittab` to remove the 'x' after the first colon on the line that corresponds to that port.

Whenever a port is used with an ASCII terminal the corresponding `/dev/ttydn` file access mode is automatically set to write only. This means that each time you switch from an ASCII terminal mode to host mode on a port you must `chmod` that file for read/write access.

Each time a change is made in either `/etc/inittab` or `/usr/lib/uucp/L-devices` the system must be rebooted for the change to be recognized by UNIX.

You may have heard that the command 'telinit 2' will initialize the system without having to reboot, thereby setting any environment changes you may have made. I recommend that you do not use this command and it is not necessary to say anything to our customers one way or the other about it. If they should ask, advise them to use whatever they feel most comfortable with.



## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: IO-006  
Model: 1400  
Subject: Vertex V100 Drive Configuration Dipswitch  
Date: July 12, 1984  
From: Bob Lum

---

#### Problem Definition:

The Vertex V100 disk drive may have a dipswitch mounted in the drive configuration socket J6. It is possible that the dipswitch may be mounted in the socket backward and the configuration settings set as though the dipswitch were mounted correctly. This condition will cause the drive to do continuous recalibration seeks after the drive is powered- on. No boot will occur.

#### Solution:

When examining the configuration setting on the Vertex V100 disk drive, and a dipswitch is encountered, the dipswitch should be mounted in the socket J6 with pin 1 of the socket and pin 1 of the dipswitch connected. Switches 4 and 8 set to "ON" and all other switches set to "OFF" for drive 0; switches 4 and 7 set to "ON" and all other switches to "OFF" for drive 1.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: IO-007  
Model: 1400  
Subject: Vertex V100 Noise Problems  
Date: July 12, 1984  
From: Bob Lum

---

#### Problem Definition:

Noise problems may exist with the Vertex V100 disk drive. The problem has been determined by Vertex Corp. and corrected by installing a felt pad between the spindle grounding contacts.

#### Solution:

Examine the Vertex V100 disk drive on the logic board side of the assembly. The drive spindle grounding contacts are located above chip number U32 and protrude through the logic board. A felt pad should be installed between the contacts. If not, order the felt pad from H.Q. Logistics.

**TECHNICAL AIDS**

Number: IO-008 R1  
Model: 1400  
Subject: Modems, 1400's & cu  
Date: July 16, 1984 Revised 8/24/84  
From: John Carter

---

**Problem Definition:**

'cu' only seems to know about dialing on AT&T modems.

**Solution:**

When using cu with other modems, specify 'dir' instead of the phone number, then type the phone number to connect to.

**Example:**

```
% cu -lttyd1 -s1200 dir <return>  
<phone number>
```

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: IO-009 R4  
Model: 1400  
Subject: Connecting a printer to a 1400.  
Date: 7/30/84 (Revised 11/5/84)  
From: John Carter

---

### Problem Definition:

A simple null modem cable will not allow some printers to work.

Using a printer at 9600 baud only prints the first few lines. Asserting ixon with the stty command has no effect.

The lp command on Version WS1.7 does not work.

The lpr command on Version WS1.7 may require some file configuration.

### Solution:

There IS magic that has to be done to get a printer to work on the workstation. The first thing necessary is that the printer have an RS232-C interface capable of up to 9600 baud.

### CONNECTING A PRINTER TO A PORT

To make the printer work, the following steps are required:

Make sure that any printer used will be interfaced properly. A cable to make this happen will look like this:

printer	w/s	Signal
1	1	Frame GND
2	3	Transmit data
3	2	Receive data
4,5		Request to Send, Clear to Send (tied together on the printer side of the cable)
7	7	Signal GND
6,8,20	20	Data Set Ready, Carrier Detect, Data Terminal Ready (tied together on the printer side of the cable)

This is a special adaptation to what is known as a null modem. Notice that pins 4 and 5 on the printer side are connected together, as well as pins 6, 8 and 20. This is what makes this cable special. Not just any null modem cable will work because of this adaptation. Some printers require that pins 4, 5, 6 and 8 be active. The 1400 does not activate these signals, therefore it is necessary to make the printer think that something is there. This interface adapter will work for any device connected to the 1400.

Connect the printer cable to Port 4 (the recommended port; Port 2 or Port 3 could also be used).

Number: IO-009 R4

---

### ENABLING THE PRINTER PORT

Login as root or rootcsh.

Change /etc/inittab to disable the login on ttyd3 (ttyd3 is the same as Port 4). This is done by using vi to add an 'x' to follow the first colon (:) in the line that starts with 'd3'. You should now have a line in the file that looks like:

```
d3:x:respawn:/etc/getty ttyd3 dx_9600
```

Older versions of software may show the entry with `co_9600`. If this is the case you may continue to use this without problems. It is an indication that the upgrade to WS1.7 was not complete.

When the login is disabled the baud rate cannot be set by inittab. To change the baud rate for this port you must keep the port busy with the sleep command so that the stty command can change the baud rate. This will be shown later.

Change the access on /dev/ttyd3 to r/w as follows:

```
% chmod 666 /dev/ttyd3
```

Next, link this device to a printer device as follows:

```
% ln /dev/ttyd3 /dev/lp
```

Reboot the system (reboot -q) to initialize these changes. Come back up in multi user mode and login as root or rootcsh.

### CONFIGURING THE PRINTER PORT

Change the baud rate on /dev/lp by doing the following:

```
% sleep 1000000 < /dev/lp &  
% stty baud_rate ixon -tabs onlcr opost < /dev/lp
```

(Note: The last character of the sleep command is an ampersand. The sleep argument of 1000000 seconds is long enough for all day. The stty options shown here are required for a NEC 7720 printer.) If you know what you are doing you can add a new line to /etc/inittab that will do these two things for you. Check the UNIX Programmer's Manual Vol 1B for details on inittab file.

### THE lpr COMMAND

The command 'lpr' may not work if the file /usr/spool/lpd/lock exists. To make sure that lpr will work remove all files in /usr/spool/lpd before using the lpr command. Now you can output the file to the printer with the lpr command:

```
% lpr filename
```

Number: IO-009 R4

### A PRINT SHELL SCRIPT

Because the sleep and stty commands should be executed each time the printer is needed, you could have several sleep processes hanging around. To avoid this here is a shell script to print a file:

(Using vi to create /usr/local/bin/print)

```
#!/bin/sh
# Be sure you get the "!" in the above line.
# printer program

# calculate the sleep time required
stime=`wc -w < $*`

# test for a minimum time period (5 seconds)
if test $stime -lt 5
then
    stime=5
fi

# put the port to sleep
sleep $stime > /dev/lp &

# change the baud rate and other options as required
stty 1200 ixon -tabs onlcr opost < /dev/lp

# output the file
lpr $*

# end
```

The baud rate setting is not necessary if you are running at 9600 baud.

If your printer can do XON and XOFF for input buffer control you need the ixon option to allow the workstation to acknowledge XON/XOFF. Other features, such as DTR, CD, RTS/CTS or DSR\* control of data transfer is not available on the workstation as of this printing.  
(\* These are hardware lines on RS232.)

The -tabs\*\* option converts tab characters to spaces.

The onlcr\*\* option converts line feed to line feed and carriage return.

The opost option causes the previous two options to be processed. Check the UNIX programmer's manual on stty to see if there are other options you may want or need.

Number: IO-009 R4

---

(\*\* If your printer can do these things then you won't need these options.)

Now you need to change the mode so anyone can execute it:

```
% chmod 755 /usr/local/bin/print
```

Now that you have set up the printer to accept data you will want to send it something, like the contents of a file. All that is necessary is to use the new print command:

```
% print filename
```

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: IO-010  
Model: 1400  
Subject: Unable to access tape drive  
Date: 8/27/84  
From: J. Carter

---

### Problem Definition:

Accessing the tape drive may give you an error message that says UNIX cannot open the device.

### Solution:

Several things may cause this to happen. If you are working on the system and have been changing tape drives check the obvious: inspect the two power cables, the ribbon cable, the DSD, and last of all make sure a tape is inserted in the drive.

The less obvious problems are software related. One of the most common causes is that `/dev/rmt1` is missing. Look in the `/dev` directory and make sure the following three files are there:

```
mt1
rmt1
qic0
```

Do a long list of these files and see that each of them shows three links. These files should actually be linked to each other.

If you had to recreate all three files and link them together here is how it would be done:

```
rm -f /dev/mt1 /dev/rmt1 /dev/qic0
mknod /dev/mt1 c 13 0
mknod /dev/rmt1 c 13 0
mknod /dev/qic0 c 13 0
ln /dev/mt1 /dev/rmt1
ln /dev/mt1 /dev/qic0
```

If the files don't show the proper number of links, or if a file is missing (generally it will be `/dev/rmt1`) then use an appropriate admixture of the above commands to recreate the proper condition.

Be sure you report your findings to the Hotline.



**TECHNICAL AIDS**

Number: IO-011  
Model: All Workstations  
Subject: Bad Tape Recovery  
Date: April 4, 1985  
From: Bob Toxen

---

**Problem Definition:**

Sometimes you may find that a tape will develop a bad spot on it. The data is very critical to you so you want to recover as much as you can past the bad spot.

**Solution:**

First, clean the tape heads as per manufacturer's specifications, probably using alcohol and cotton or sponge swabs. Try tar. If this fails continue.

Next, decide whether the data is valuable enough to spend some time trying to extract it. If it is then continue.

Use "dd conv=noerror bs=512 </dev/rmt1 >tmp" to read the tape.

Then do "tar -xvbf 1 tmp" to extract the good stuff.

Note the file it chokes on. Then do GREP on tmp for that file name with the -b flag to tell what block in tmp that it is on. Then do a DD from tmp to skip to that block and copy the rest of the file to tmp2.

Then do "od -c tmp2 | more" to dump the file in a printable fashion searching for the "directory header" for the next file and note the offset given by OD.

You can use DD on tmp2 to extract from the block after the first dir header (on the file with the bad spot) to "file1". This is as much of the data for the file with a bad spot as you can get. It may be of some use.

Then do a DD on tmp2 skipping to the correct offset to reference the directory header for the second file (past the bad spot) copying to tmp3.

You can now do "tar -xvbf 1 tmp3" to extract the rest of the tape. If there are additional bad spots repeat the procedure, starting with tmp3.

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: IO-012 R1  
Model: GL1/GL2  
Subject: ttyd1 gets assigned as syscon  
Date: July 29, 1985  
From: John Carter

---

### Problem Definition:

There are conditions under which a workstation will boot up only on the ASCII terminal connected to port 2. One of them is normal and that is when the main console keyboard is disconnected. The other one is not normal and that is when, for some obscure reason, /dev/syscon gets linked to /dev/ttyd1 (or some other port for that matter).

This has been known to happen when a customer deliberately links syscon to some other port than console. It can also happen if the operator performs some operation on port 2 that causes that port to be linked (by the operating system) to the console. One such operation is to issue the command 'gclear' from port 2 and then press the reset button. When the gclear command is issued it hangs the main console (for some reason).

**Failure:** /dev/ttyd1 is linked to /dev/syscon

**How to know:** Type the following:

```
cd /dev
ls -li console syscon ttyd1
```

What you should see is that the inode numbers for syscon and ttyd1 are the same. Also note that the major and minor device numbers for console are 0,0 and for ttyd1 and syscon they are 3,1.

An example of a correct environment is as follows:

```
154 crw--w--w- 2 root  sys    0, 0 Jul 29 18:32 /dev/console
154 crw--w--w- 2 root  sys    0, 0 Jul 29 18:32 /dev/syscon
385 crw--w--w- 1 root  sys    3, 1 Jul 29 17:58 /dev/ttyd1
```

### Solution:

The preferred way to get out of a main console hang is to try to kill the offending process from another terminal (or over the net if that is working).

To resolve the problem of linked drivers simply stay in single-user mode when the system is rebooted or while in multi-user mode login as root. Now type the following commands:

Number: IO-012 R1

---

```
cd /dev
rm syscon (you must remove the linked file first)
ln console syscon (this command recreates syscon as an identical file to console)
ls -li console syscon ttyd1
reboot -q
```

and everything should be all right.

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: IO-013  
Model: 2400 W2.3  
Subject: Connecting a printer to a 2400.  
Date: August 6, 1985  
From: John Carter

---

### Problem Definition:

The IRIS Workstation Guide (5001-096-001-0) seems to leave a little to the imagination, but not much, to make a printer work, so here's the rest of the story.

### Solution:

#### Software Data Flow Control

Some printers use XON (DC3, CTRL-Q) and XOFF (DC1, CTRL-S) for software control of data flow. Data flow control is necessary when the data being sent to the printer occurs at a faster rate than the printer can use it. When the printer's input buffer becomes full what could happen is that garbage will begin to print out (because of data overrun in the input buffer) unless something is done to stop the input data flow. For software control XON/XOFF is used. The workstation is configured to acknowledge this control by setting ixon at the port (`stty ixon < /dev/lp`). A similar example is shown on page 63 of the manual. A mistake in the manual is found on page 63 on the line that reads:

```
sleep 10000000 > /dev/ttyd3 &
```

it should read:

```
nohup sleep 10000000 < /dev/ttyd3 2> /dev/null &
```

There are three changes. The first is the inclusion of the nohup. This command causes the sleep to continue to be executed even if for some reason the parent process dies.

The next change is the redirection symbol used. The '>' should be changed to '<' or the sleep command will not show up on a process status.

The last change is the inclusion of the '2> /dev/null'. This is used to redirect the standard error caused by the nohup at the beginning of the line and prevents it from being seen on the monitor when going into multi-user mode. The message not seen simply states, "Sending output to nohup.out". Another line (after the stty) could be added to remove nohup.out but it is rather pointless.

An older manual has confused some customers in finding the right place in /etc/rc by instructing the programmer to place the new three lines above the last line in /etc/rc which reads 'fi', and in fact there is no such last line. A change was made in /etc/rc and now the instructions are to put the three lines after 'echo "."'. This should occur near the bottom of the file after all the Daemons have been started.

#### Hardware Data Flow Control

In Appendix F are several tables for connecting devices to the workstation. These are good but do not tell

Number: IO-013

the whole story for a clear understanding.

Some printers use pins 4 and 5 for hardware control of data flow (pin 5 at the printer is the signal from the printer).

One serial to parallel interface adapter (Tigertronix) uses only pins 2 and 20 for hardware control of data flow (pin 20 at the printer is the signal from the printer).

To accomodate these devices it will be necessary to install the printer using the modem control port, port 3 (/dev/tty2). Here is the information needed to configure port 3 in software:

In /etc/inittab, make sure the following entry exists:

```
m2:x:respawn:/etc/getty tty2 dx_9600
```

and also make sure that the d2 entry has an x in its second field.

Make sure lp is not already linked to tty2. This can be done just by executing 'rm /dev/lp'.

Now link tty2 to lp:

```
ln /dev/tty2 /dev/lp
```

It will not be necessary to change the access mode of tty2 (lp) because it is already able to write and does not need to read. If the baud rate is other than 9600 then it will be necessary to include the three lines in /etc/rc just like for ixon except only the speed option will be used in the stty command.

The printer presumably will signal the workstation via the Clear To Send signal (pin 5, CTS, on the workstation). The cable needed to connect the printer should look like the following:

WS	PRINTER	SIGNAL
1	1	shield
2	3	Transmit data (TD)
3	2	Receive data (RD)
4	5	Request to Send (RTS)
5	4	Clear to Send (CTS)
6	6	Data Set Ready (DSR) (typically not used)
7	7	Signal Ground (SG)
8	20	Carrier Detect (CD)
20	8	Data Terminal Ready (DTR)

If the device is a Tigertronix serial to parallel converter then the following simplified cable is required:

WS	PRINTER	SIGNAL
2	3	TD
7	7	SG
5	20	CTS
6,8,20		DSR, CD, DTR -- Tied together at WS side.

Number: IO-013

---

But until the modem control feature is fixed it won't do anyone any good. The bug in the modem control is that it does not care whether or not the connected device attempts to stop data flow by signalling Clear To Send (pin 5, CTS).

The design intention was to have Carrier Detect (pin 8, CD) on the workstation serve for flow control. Data Terminal Ready (pin 20, DTR) on the printer or modem was to provide the hardware data flow control. This also does not work because anytime pin 8 on the workstation is toggled for data flow it causes an interrupt that cancels the output altogether, as found out by manual experimentation.

An SCR has been submitted recently and you will be notified if this status changes at all.

# SILICON GRAPHICS PRODUCT SUPPORT ENGINEERING

---

## TECHNICAL AIDS

Number: IO-013 R1  
Model: 2400 W2.3  
Subject: Connecting a printer to a 2400.  
Date: August 6, 1985  
To: Prod. Sup., Field Eng., System Eng., Tech. mktg, OEM, VAR  
From: John Carter  
Approved:

---

### Problem Definition:

The IRIS Workstation Guide (5001-096-001-0) seems to leave a little to the imagination, but not much, to make a printer work, so here's the rest of the story.

### Solution:

**Software Data Flow Control** (NOTE: The update to this Tech Aid is complex. Please read it all. One change concerns the use of `tty2`. Don't. It is broken at all release levels up to and including 3.4/2.4. Another change talks about testing serial ports. This is very useful information.)

Some printers use XON (DC3, CTRL-Q) and XOFF (DC1, CTRL-S) for software control of data flow. Data flow control is necessary when the data being sent to the printer occurs at a faster rate than the printer can use it. When the printer's input buffer becomes full what could happen is that garbage will begin to print out (because of data overrun in the input buffer) unless something is done to stop the input data flow. For software control XON/XOFF is used. The workstation is configured to acknowledge this control by setting `ixon` at the port (`stty ixon < /dev/lp`). A similar example is shown on page 63 of the manual. A mistake in the manual is found on page 63 on the line that reads:

```
sleep 1000000 > /dev/ttyd3 &
```

It should read:

```
nohup sleep 1000000 < /dev/ttyd3 2> /dev/null &
```

There are three changes. The first is the inclusion of the `nohup`. This command causes the sleep to continue to be executed even if for some reason the parent process dies.

The next change is the redirection symbol used. The `>` should be changed to `<` or the sleep command will not show up on a process status.

The last change is the inclusion of the `2> /dev/null`. This is used to redirect the standard error caused by the `nohup` at the beginning of the line and prevents it from being seen on the monitor when going into multi-user mode. The message not seen simply states, "Sending output to nohup.out". Another line (after the `stty`) could be added to remove `nohup.out` but it is rather pointless.

An older manual has confused some customers in finding the right place in `/etc/rc` by instructing the programmer to place the new three lines above the last line in `/etc/rc` which reads `'fi'`, and in fact there is no such last line. A change was made in `/etc/rc` and now the instructions are to put the three lines after `'echo .'`. This should occur near the bottom of the file after all the Daemons have been started.

Number: IO-013 R1

### Hardware Data Flow Control

(NOTE: Repeat - don't use ttym2 as it is broken. See note above.)

In Appendix F are several tables for connecting devices to the workstation. These are good but do not tell the whole story for a clear understanding.

Some printers use pins 4 and 5 for hardware control of data flow (pin 5 at the printer is the signal from the printer).

One serial to parallel interface adapter (Tigertronix) uses only pins 2 for input data and 20 for hardware data flow control (pin 20 at the printer is an output from the printer).

To accommodate these devices it will be necessary to install the printer using the modem control port, port 3 (/dev/ttym2). Here is the information needed to configure port 3 in software:

In /etc/inittab, make sure the following entry exists:

```
m2:x:respawn:/etc/getty ttym2 dx_9600
```

and also make sure that the d2 entry has an x in its second field.

Make sure lp is not already linked to ttyd2. This can be done just by executing 'rm /dev/lp'.

Now link ttym2 to lp:

```
ln /dev/ttym2 /dev/lp
```

It will not be necessary to change the access mode of ttym2 (lp) because it is already able to write and does not need to read. If the baud rate is other than 9600 then it will be necessary to include the three lines in /etc/rc just like for ixon except only the speed option will be used in the stty command.

The printer presumably will signal the workstation via the Clear To Send signal (pin 5, CTS, on the workstation). The cable needed to connect the printer should look like the following:

IRIS	PRINTER	PRINTER SIGNAL
1	1	shield
2	3	Receive data (RD)
3	2	Transmit data (TD)
	5	Clear to Send (CTS) (NOTE: do not connect to IRIS)
	4	Request to Send (RTS) (NOTE: do not connect to IRIS)
6	6	Data Carrier Detect (DCD) (intended for hardware data flow control. DO NOT USE)
7	7	Signal Ground (SG)
20	8	Data Set Ready (DSR). This most often is not needed.

NOTE: Pin 20 on the IRIS does not become active until either cu is executed or the port is configured for a login. The active state never goes away until the system is rebooted. This means that if the printer or modem requires an active pin 20 and no login (getty daemon) is to be used, then a cu should be executed and



Number: IO-013 R1

then exited to bring up pin 20. The sequence is as follows:

configure /usr/lib/uucp/L-devices for the port. Example entry:

```
DIR ttyd3 ttyd3 9600
```

Then do cu:

```
cu -lttyd3 -s9600
```

When it says "Connected", exit cu by typing ~. (tilde dot). This must be done after a RETURN has been pressed or it will not be recognized properly.

DTR is now active. The other way to activate it is with ioctl calls in a program. This is left to the user to discover.

If the device is a Tigertronix serial to parallel converter then the following simplified cable is required:

IRIS	PRINTER	PRINTER SIGNAL
2	3 RD	
7	7 SG	
	4,5 RTS,CTS	
	6,8,20 DCD,DSR,DTR	

Also, make sure that the baud rate is set low enough that the printer input buffer will not overrun.

Until the modem control feature is fixed it won't do anyone any good.

The design intention was to have Carrier Detect (IRIS pin 6, CD) serve for flow control. Whatever pin on the printer or modem that was to provide the hardware data flow control was to be connected to the IRIS at pin 6. This also does not work because any time pin 6 on the workstation is toggled for data flow it causes an interrupt that cancels the output altogether, as found out by manual experimentation. Release 3.4/2.4 causes an even more catastrophic failure.

An SCR has been submitted recently and you will be notified if this status changes at all.

#### TESTING THE SERIAL PORT:

Use a paperclip bent like a hair pin and inserted into pins 2 and 3 of the port.

Execute cu as shown above and when connected, type anything on the console keyboard and it should show up on the monitor exactly as typed. If it does, you have verified that most of the hardware and system software works, and enough of it works to be able to use the printer.

Remove the paperclip and connect the printer. Type a line and when you press RETURN (for mo= printers) the line you entered will print. If it doesn't then the problem is with the printer or cable for sure.

Number: IO-013 R1

---

Exit cu by typing ~. (tilde dot). This must be done after a RETURN has been pressed or it will not be recognized properly.

Send a test file to the printer (try /etc/rc). If it looks good you're home free, otherwise the user has some figuring out to do to set up his printer and the IRIS port stty options to make things work they way they want it to.

**TECHNICAL AIDS**

Number: IO-014  
Model: Workstations  
Subject: XNS between two workstations  
Date: August 20, 1985  
From: John Carter

---

**Problem Definition:**

Customer just got two 2400's and wants to play dogfight on XNS ethernet. When they start the programs they don't seem to work right. Flight is okay.

**Solution:**

Edit the /etc/sys\_id files and make sure each system has a different name. Then reboot both systems. Test the ethernet by trying to log in from one system to another. This is done by first logging in to one system as root or guest or whatever, and then type 'xlogin hostname' where hostname is the name of the other system. If this doesn't work, either you have a hardware problem or you didn't do something right.

**Possible failure symptoms:**

unknown host: (hostname)  
ethernet timeout: (reason)

**Possible software problems:**

- a. The first approach is to assume you didn't do this sequence right, so do it again.
- b. The /defaultboot program may not be a good copy of /kernels/2400 (or /kernels/1400 or whatever).
- c. /etc/xnsd is bad.

**Possible hardware problems:**

- a. Make sure that there are no systems connected to the ethernet coax that are not powered up. This may not be the problem but we should have a controlled environment when troubleshooting a problem like this.
- b. Make sure that both ends of the coax have a proper terminator. Somehow convince yourself that the transceivers are making good connection (stinger to inner conductor). You can't do this without taking the cover off the transceiver and testing for continuity between the stinger and the shield (should be ~25 ohms).
- c. Check for all the possible bad connections at the cables and PC boards and the IC's on the PC boards.
- d. Replace anything you can with a known good spare.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: IO-015  
Model: GL1/GL2  
Subject: NTSC problems  
Date: Dec. 16, 1985  
From: David Koontz

---

#### Problem Definition:

NTSC output is unstable, will not sync.

#### Solution:

The determination as to where the problem is can be difficult.

The easiest thing to start with is to make sure the customer is using either "black burst" sync (similar to composite video, only no video - all black,  $+0.7v$  to  $-0.7v$ ), or composite sync (0 to  $-4v$ ). It is common that the customer may be using a composite video signal that is not truly a black burst signal.

Otherwise, the problem either lies in the CG1 or the DC4. If the RGB RS-170 image will show up on a monitor with the sync from the IRIS, the problem is probably on the DC4. If the image won't lock (Horizontal and Vertical frequencies drastically wrong) the problem is more than likely on the CG1.

Exception: A known trace open due to a spot on the negatives occurs on the DC4 (early boards). This was located on the foil side of the board at location U14A. Pins 1 and 19 should be at ground potential and are sometimes floating.

Common problems causing the IRIS to free run (with a viewable image when using the sync from the IRIS, in a genlock environment):

INT/EXT Clock jumper not in EXT, DC4 location H22D.

Jumper E23 on CG1 installed (Should be removed).

Refer to the CG1 functional Description for indicator LEDs. LED3 (the fourth one from the top lights when the CG1 is genlocked. LED0 (top) lights when external sync is input. Refer to the CG1 Test Procedure for other information the LEDs convey.

Common Causes of a free running IRIS output that cannot be viewed with the IRIS sync in a genlock environment:

Above DC4 problem with etching short

Cable between the DC4 and the CG1

Cable CG1 to Aux I/O panel (connection to pot), LED0 will indicate difficulty with external sync reaching the CG1.

Number: IO-015

---

Missing Jumpers on the CG1. E11, E13 and E15 do not have any significance in the present environment. Refer to the Functional Description and Test Procedure for jumpering information, or the Assembly Drawing.

A BUM CG1.

**Other Symptoms:**

This may be caused by a faulty connection to the external sync input, poor quality signal on that input, or a misaligned CG1. Hardware faults on the CG1 may be suspect.

**System Hangs Up:**

When Prom monitor boot environment specified, and Keyboard hooked up, periodic cycling of the leds on the Keyboard may be seen. The IRIS monitor may appear to flicker in time with this action (Note interval between cycles approximately 8 seconds).

This is caused by no clock being supplied to the DC4 from the CG1. Suspect cables or a defective CG1. If the keyboard is disconnected, port 2 on the rear panel may be used as console to check operation of the system as a computer only.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: IO-016  
Model: 2500/2500T  
Subject: Interphase SMD Controller upgrade before install of 3.4  
Date: 5/7/86  
From: David Williams

---

#### Problem Definition:

All 2500/2500Ts shipped prior to 3/5/86 (including 2500 Turbo upgrade kits) must have their 2190 controllers swapped out with modified 2190s before the new 3.4 software is installed. This is because 3.4 uses the Extent File System.

#### Solution:

Check to see that your 2190 SMD controller has the following DA#92 installed:

If rev. d: check that U61-13 has been lifted and jumpered to U34-1

If rev. f: check that U61-13 is jumpered to U34-1

All rev's: check that U44 prom has P/N 39044430

This upgrade is too complex to be done in the field. If your controller does not have the upgrade, order a replacement from Logistics.

If it is necessary to order a replacement this does not mean that the system is to be left unuseable. The root file system can be loaded onto the Extent File System as usual (not much choice, it gets created that way during Tape to Disk load using fex). When ready to load the usr files (partitions c and d), create the partitions using mkfs.bell and not mkfs. Here's the command lines to use:

```
mkfs.bell /dev/rip0c 49280 2 440
```

```
mkfs.bell /dev/rip0d 616000 2 440
```

Load the usr files as usual and inform the customer that his usr partition is temporarily not up to date. When the replacement SMD arrives backup the usr files and recreate the usr partitions in the correct manner:

```
mkfs /dev/rip0c
```

```
mkfs /dev/rip0d
```

Reload the usr files as before.

The only affect of using the Bell File System is that disk access will be slower (the same as before).

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: IO-017  
Model: 2x00 30x0  
Subject: Euro Video Display Problem  
Date: xx  
From: David Williams

---

### Problem Definition:

Two versions of DC-4 boards (5000-090-05 and 5000-090-06) were sent out with proms sets that fail when running in EURO-VIDEO format. The symptom is that the last 12 pixels of data display in each scan line are not displayed.

### Solution:

If customer is running in EURO-VIDEO format check that DC-4 has DA#104 installed on revision 05 and 06 DC-4's. The change involves 2 proms at U17D and U18D. The correct prom rev # vs board rev # follows:

500-090-05	U17D	should be	5000-396-07
"	U18D	"	" 5000-397-07
500-090-06	U17D	"	" 5000-396-08
"	U18D	"	" 5000-397-08

If the board has the correct proms but doesn't have DA#102 marked on the board please mark it.

If customer is running in EURO-VIDEO and has either DC4 rev 5 or 6 without DA#104 installed please arrange to swap it out at the customers convenience.

**TECHNICAL AIDS**

Number: IO-018  
Model: ALL  
Subject: IEEE-488 programming tips  
Date: May 9, 1988  
From: John Carter

---

**Problem Definition:**

Subject: IEEE access

Everything the customers need to know to make it really work.

(This article is mostly the effort of Scott Bovinizer and Donavon Fong)

**Solution:**

Pointers on using GPIB:

Init:

- file: /etc/ibtab
- first entry is self
- rest of entries are everyone else
- contents have to be set up according to situation
- determine by options, flags, nodes, etc. in manual

Bringing up system:

Multi user

- lib called by rc.488
- r option which has to do restart.
- takes info in ibtab file and stores it in low
- core that is part of driver in kernel

Usage:

Use standard C as if device was file

- open
- close
- read
- write

For really complicated stuff

- Most programs should be able to access gpib without
- the need for using ioctl.

raise protocols etc.

Involves calls to ioctl

- use same file descriptor from open

- give set of commands

- ie: IBIOGETNODE

- to use device on more direct
- basis

- /usr/include/sys/ib\_ioctl.h

- has list of commands



Number: IO-018

As part of ioctl call the last argument has to be a pointer to data structure of type sgnode defined in .h file

eg: struct sgnode ibbuf;  
ioctl(fd,cmd,&ibbuf);

BEFORE DOING IOCTL CALL you must set SLOTNO to a value from 0 to 7 (for access to core data structure) which corresponds to ib0-7; it's an individual data structure for the individual device.

There are subroutines to fool around with ibtab file.

/usr/include/ibtab.h

For gut level (really low level) access

/usr/include/gpib/\*.h

(some of these files may not be in the final release.)

A simplified method for doing a straight read is as follows:

set up the ibtab file,  
init system,  
type: cat < /dev/ib? (where ? is a number 1-7)

if the other side is sending something it will be received.

#### NOTES:

1. IEEE 488 has 1k read buffer on board.
2. For buffer to begin to be passed to a software read command, ie cat, the sender must activate the end of transmission line (EOI) with the last byte sent.
3. Our board does support extended addressing.
4. Never use /dev/ib0 for i/o. It is supposed to be used for Configuration only.

Some things to watch out for in 2400 to 2400T upgrades:

- 1) some old versions of the IEEE 488 release evidently had lib and ibtab files in a directory "/etc/gpib". in the turbo release, they occur in "/etc". check /etc/rc.488.
- 2) The site-dependent ibtab file can get lost during the upgrade process.

# SILICON GRAPHICS PRODUCT SUPPORT ENGINEERING

---

## TECHNICAL AIDS

Number: IO-018 R1  
Model: ALL  
Subject: IEEE-488 programming tips  
Date: May 9, 1986 Revised May 23, 1986  
To: Prod. Sup., Field Eng., System Eng., Tech. mktg, OEM, VAR  
From: John Carter  
Approved:

---

### Problem Definition:

Subject: IEEE access

Everything the customers need to know to make it really work.

(This article is mostly the effort of Scott Bovinizer and Donavon Fong)

### Solution:

Subject: IEEE access

### Pointers on using GPIB:

#### Init:

File: /etc/ibtab

First entry is self.

Rest of entries are everyone else.

Contents have to be set up according to situation.

determine by options, flags, nodes, etc., in manual.

To configure the local system as other than a controller

modify the first entry from:

```
/dev/ib0:/dev/ib0:0:VALID|SWAB|SRQ|PPC|SC :5:010:board 0 self
```

to:

```
/dev/ib0:/dev/ib0:0:VALID|SWAB|SRQ|PPC :5:010:board 0 self
```

### Bringing up system:

#### Multi user

iib called by rc.488

-r option which has to do restart.

takes info in ibtab file and stores it in low

core that is part of driver in kernel.

### Usage:

Use standard C as if device was file:

open  
close  
read  
write

Number: IO-018 R1

For really complicated stuff:

Most programs should be able to access gpib without the need for using ioctl.

Raise protocols etc.

Involves calls to ioctl -

Use same file descriptor from open.

Give set of commands

ie: IBIOGETNODE

(to use device on more direct basis)

/usr/include/sys/ib\_ioctl.h

(has list of commands)

As part of ioctl call the last argument has to be a pointer to data structure of type sgnode defined in .h file.

eg: struct sgnode ibbuf;

ioctl(fd,cmd,&ibbuf);

BEFORE DOING IOCTL CALL you must set SLOTNO to a value from 0 to 7 (for access to core data structure) which corresponds to ib0-7; it's an individual data structure for the individual device.

There are subroutines to fool around with ibtab file.

/usr/include/ibtab.h

For gut level (really low level) access

/usr/include/gpib/\*.h

(some of these files may not be in the final release.)

A simplified method for doing a straight read is as follows:

set up the ibtab file,

init system,

type: cat < /dev/ib? (where ? is a number 1-7)

If the other side is sending something it will be received.

#### NOTES:

1. IEEE 488 has 1k read buffer on board.
2. For buffer to begin to be passed to a software read command, ie cat, the sender must activate the end of transmission line (EOI) with the last byte sent.
3. Our board supports extended addressing.
4. Never use /dev/ib0 for i/o. It is supposed to be used for configuration only.

Some things to watch out for in 2400 to 2400T upgrades:

- 1) Some old versions of the ieee 488 release evidently had ib and ibtab files in a directory "/etc/gpib". In the turbo release, they occur in "/etc".

**TECHNICAL AIDS**

(Cont'd)

Page 3

Number: IO-018 R1

---

Check /etc/rc.488 .

- 2) The site-dependent ibtab file can get lost during the upgrade process.

**TECHNICAL AIDS**

Number: IO-019  
Model: ALL  
Subject: RS-232 devices are hanging the system during boot  
Date: 5/14/86  
To: Prod. Sup., Field Eng., System Eng., Tech. mktg, OEM, VAR  
From: Dave Williams  
Approved:

---

**Problem Definition:**

Customers are connecting RS-232 devices that work until the system is rebooted. It seems that our system enables the line RTS (pin 4) during the boot but the system is not able to deal with this line changing until the Getty for that port has been started.

**Solution:**

**Workarounds include:**

- 1) Connect and power up the device before booting the IRIS.
- 2) Use a 3-wire null modem cable as explained in the IRIS workstation guide Appendix H for 1000 series, Appendix F for 2000 series or Chapter 7.10 for 3000 series. This is the preferred method.

The failure is in software at the kernel level of the UNIX operating system.

**NOTE:**

The device must be able to support XON/XOFF for data flow control or be set at a baud rate low enough not to need data flow control (multiply the print cps rate by 10, set available baud rate at or below that).

**TECHNICAL AIDS**

**INDEX - DIAGNOSTICS**

Updated: 2/27/86

---

**DIAG-001**

**Sky Board Diagnostics**

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number:       DIAG-001  
Model:        1400  
Subject:      Sky Board Diagnostics  
Date:         8/27/84 (Revised 10/4/84; JRC)  
From:         J. Carter

---

### Problem Definition:

A problem has been found in the Sky Board diagnostic that went out with Release 1.7. When a board is installed in a system and the system is booted, the diagnostic may give the error message:

fload: board test failure

even though the board is really functioning correctly.

### Solution:

The message should simply be ignored, as the board has been initialized correctly and has been checked by manufacturing (with a new version of the program).

This will be fixed in the next release of software (1.8? due in Nov.?).

Any other messages from fload should be correct.

TECHNICAL AIDS

INDEX - SOFTWARE

Updated: 8/4/86

---

SOFT-001	Software Release R1a Overview
SOFT-002	XNSD start up (disable xns)
SOFT-003	TPE (Terminal Programming Environment).
SOFT-004	Fex - new release for 2.4/3.4
SOFT-005	GL2-W2.3 Release Notes



# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: SOFT-001 (updated 9/28/84; JRC)  
Model: 1000 R1b, 1200 R1b  
Subject: IRIS Distribution Tapes  
Date: Jan. 25, 1984  
From: John Carter

---

### Problem Definition:

Introduction of software release R1a requires some orientation of the tapes available, their names, part numbers, contents, and purpose.

### Solution:

The following is a complete description of the tapes available that comprise the software release known as R1a as of January 20, 1984.

Tapes by Name:	Part Number:
Programming Environment	5001-100-001-0
IRIS Source	5001-101-001-0
Serial Distribution	5001-102-001-0
Net Distribution	5001-103-001-0
Net Distr/Driver Source	5001-104-001-0

The 'standard' distribution tape is the Serial tape (102), assuming that the customer has not ordered the option for the host ethernet.

The IRIS source tape (101) is issued to those customers that have a Vkernel source license from Stanford and gave ordered source from us. This tape contains source code for the terminal Vkernel as well as the IRIS source code.

The Net Distribution tape (103) is used in place of the Serial Distribution tape for those customers that have ordered the host ethernet option.

The Net Distr/Driver Source tape (104) is only used in place of the Net Distribution tape for those customers that have a UNIX 4.2 source license from Berkeley. This tape does not contain source code for the terminal Vkernel. This tape is necessary if the customer has more than one Interlan board on his host.

The Programming Environment tape (100) allows the customer to create programs that run in stand-alone mode on the terminal. This tape is available only to those customers that have signed a software licensing agreement with SGI.

The maximum number of tapes that could go to a customer is three:

1. One of the following three:
  - Net Distribution
  - Net Distr/Driver Source
  - Serial Distribution

**Number:      SOFT-001 (updated 9/28/84; JRC)**

---

2. IRIS Source
3. Programming Environment

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: SOFT-002  
Model: 1400  
Subject: XNSD start-up  
Date: 12/13/84  
From: Vincent Machnig

---

### Problem Definition:

System continually logs xnsd failures in multi user mode when E-net board is removed or not recognized by system.

### Solution:

Disable all processes that attempt to access the ethernet.

- 1 Before removing E-net board, bring system up in single user mode.
- 2 Change directory to /etc.
- 3 Using 'vi' edit the 'rc' file and comment out the line for xnsd startup. Also make sure the lines containing 'sgboot' and 'sgbounce' are commented out. See example 1 below.
- 4 Again using 'vi', go into 'inittab' file and insert an 'x' in the second field for both the 'n1' and 'n2' entries (last two lines). See example 2 below.

(Example 1: /etc/rc)

```
/etc/update;          echo "Daemons:"
/etc/cron;            echo " update"
# /etc/xnsd;          echo " xnsd"
(NOTE: the last line above has the '#' in the first column)
# uncomment these two lines if you are booting iris terminals over xns:
#
# /etc/sgboot '/bin/hostname' /usr/local/boot; echo " sgboot"
# /etc/sgbounce '/bin/hostname' /usr/local/boot; echo " sgbounce"
(NOTE: the last two lines above have the '#' in the first column)
echo "Daemons started."
```

(Example 2: /etc/inittab)

```
n1:x:otherstuff...
n2:x:otherstuff...
```

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: SOFT-003  
Model: ALL  
Subject: TPE (Terminal Programming Environment).  
Date: August 20, 1985  
From: John Carter

---

#### Problem Definition:

Customer has a workstation and a terminal and wants to download Iris from the workstation to the terminal. The purpose here is to have the workstation serve as the remote cpu to the terminal.

#### Solution:

What follows is a guideline. It is assumed that whoever is doing the install knows what they are doing and the information given here is sufficient to cover the needs specific to this configuration.

This requires a software package on a 1/4 inch cartridge tape. This software is the RGL (Remote Graphics Library) for the terminal. The tape should be identified as being configured for one of the following:

Workstation	Terminal
1400/1500	1000/1200
1400/1500	2000/2200
2400/2500	1000/1200
2400/2500	2000/2200

There should be sufficient documentation to walk the customer through installing this tape and making everything work, however, I suspect that one detail may be overlooked. Be sure that the /etc/rc.xns file has been edited to remove the pound signs from the sgboot and sgbounce lines. If this has not been done the workstation will not serve the terminals (no bootserver responding).

Essentially the RGL that will work is the very same software that the customer would get to put on his VAX except it will now be installed on the workstation. The documentation should mention that two system variables are to be set after loading the software and before performing the install. These two variables are SYSTEM and MACHINE. SYSTEM is set to IRIS1400 or IRIS2400, as appropriate, and MACHINE is set to MC68000. The rest of the install procedure will be the same as for a VAX and the terminal software installation guide will do well for that.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: SOFT-004  
Model: ALL Workstations  
Subject: Fex - new release for 2.4/3.4  
Date: May 13, 1986  
To: Prod. Sup., Field Eng., Sales Eng., Tech. Mktg, OEM, VAR  
From: John Carter

---

#### Problem Definition:

Customers have been using mdex, ipex, or sifex to operate on their disk drives and have then called in with more problems than they started with. Also, customers have always had to call in to get information on how to use these tools to correct problems with their disk. In all cases, the number of hours spent training customers on the proper use of fex, and sometimes even trying to figure out what damage they did when they willy-nilly did their own thing with it, has been too great a headache to continue with. There is a better way to deal with this kind of problem.

#### Solution:

When the next release of software comes out (2.4/3.4) the fex programs (mdex, ipex, sifex) will all be changed.

There will be fewer commands available to the customer in the new program.

What the customer loses:

- format
- mapbad
- exercise
- badblock

What the customer gets to keep:

- tape copy
- set label (for changing file partition size)

The reason for this is the original fex program was never created to be anything other than a diagnostic tool for Field Engineering and Manufacturing and Engineering. It was originally written by Chase Bailey for his own debugging use. It inadvertently got into one of the releases and was never handled properly. It cannot be documented because it has too many unknown and known problems. No Engineering effort has been given to it to correct its problems.

Now, we have an Engineering priority to replace much that fex does with supported tools that will be fully documented for customer use. Since there is no time to provide this code by the next release, a decision was made to modify the various versions of fex.

This means that fex is going to be distributed as it was, except most of the commands will be hidden from view. A special procedure must be used to access these commands. When fex is booted type 'h' for the help

Number: SOFT-004

---

list and you will notice that most of the commands are gone. Type 'Z' (capitalized) and you get a prompt for a password. Here is where the fun begins, but first a little background.

We expect that a new release of software with the properly supported utilities will be available before many customers have a need to use fex for anything other than doing a tape copy. What we want to do is protect our customers from themselves and prevent their using fex to exercise, format, or to map bad blocks (which I've found not to work many times) without us knowing about it (because it isn't documented, and two customers and some SE's have wiped out disks in their anxiousness to do the right thing). In this light, the new fex will ask for a password each time you enter 'Z'. To prevent anyone from learning the password and then charging ahead on their own, the program has been armed with a revolving password, eight of them. This means that each time you enter 'Z' you will need a new password, for the first eight times. We figure that will give us enough time to release the properly documented tools.

ONLY the FE's, OEM's, and Product Support will have access to these passwords. No password shall be provided in print, even here. To get the list of passwords, call the Hotline. ONLY FE's will be given the password list over the phone. OEM's will get the list by special mailing. All others will get only the password needed at the time of use.

Each time you enter 'Z' the prompt "Enter Password" will appear. Notice that the first time you do this the 'P' will be capitalized. The second time you do this the 'A' will be capitalized, and so on for the eight letters of PASSWORD (e.g., passWord). This is the key as to which password to use each time you enter 'Z'. After eight times the list will start over again. By this time the new tools will be out. It is unlikely that any customer will have the smarts enough to write down each of the passwords, especially if we tell them it changes internally at each use and that only we have the key. Don't bother explaining anything to them either.

There will be a new procedure for assisting customers who call in with disks that are corrupted and may need formatting or bad blocks mapped. Customers with FULL maintenance support or are in warranty and have purchased installation (on-site service) will have an FE dispatch on this type of call. The FE will perform the required service, with the aid of Product Support if necessary. The FE may choose to aid the customer over the phone in lieu of going on site, but in any event it will be an FE service and not a Hotline service for this type of call. The FE may elect to replace the disk drive instead of trying to recover it.

For those customers who have BASIC maintenance or are in warranty and did not purchase installation (parts only service), and for those customers who neither have on-site service or parts only service, the Hotline will process the call completely.

If there are any comments or questions you have about this memo, please feel free to call on Beat Poltera.

**TECHNICAL AIDS**

Number: SOFT-005  
Model: All GL2-W2.4 systems at 2.3  
Subject: GL2-W2.4 Crash Recovery & Updates  
Date: 25 July 1986  
To: Prod. Sup., Field Eng., System Eng., Tech. mktg, OEM, VAR  
From: John Carter  
Approved:

---

**Problem Definition:**

There are two problems to note here. One concerns Crash Recovery and the other concerns updating a system to 2.4.

**CRASH RECOVERY:** The IRIS Workstation Guide Series 2000 Ver. 2.0, Section 4.5.3, (Recovery by Booting from Tape) is incomplete for this procedure.

**UPDATE:** The GL2-W2.4 Workstation Release Notes Version 1.0 (007-3101-010) beginning on page 7 of the document are slightly inaccurate and incomplete.

**Solution:**

**For CRASH RECOVERY:**

1. Steps 6, 7, and 8 should be replaced with the instructions given in the man page for mkboot in the new man pages for UNIX Programmer's Manual, Vol 1A Ver. 2.0.
2. Step 9 and 10 on page 4-30 is fine for 60 Hz monitors, but if the customer has a 30 Hz monitor then he must do the following sequence:

b md0g:mon (The system will load a new monitor kernel and return to the IRIS prompt. This must be done each time the system is rebooted.)

b (This is the final boot)

3. Step 13 on page 4-31 should read:

```
mt rewind
mt fsf 3
cpio -ivhmud2
```

Number: SOFT-005

---

**For UPDATE:**

1. On page 7, step 6, the last line shows the **mdfex** prompt as:

**mdfex7.1>**

and should be changed to read:

**mdfex2.4>**

2. Step 7 is to be replaced with the instructions given in the man page for **mkboot** in the new man pages for **UNIX Programmer's Manual, Vol 1A Ver. 2.0**.

3. Step 9 on page 9 is fine for 60 Hz monitors, but if the customer has a 30 Hz monitor then he must do the following sequence:

b md0g:mon (The system will load a new monitor kernel and return to the IRIS prompt. This must be done each time the system is rebooted.)

b (This is the final boot)



SILICON GRAPHICS FIELD ENGINEERING

---

TECHNICAL AIDS

INDEX - HOST

Updated: 2/28/86

---

HOST-001  
HOST-002

Null Modem  
DELNI Multiplexer

# SILICON GRAPHICS FIELD ENGINEERING

---

## TECHNICAL AIDS

Number: HOST-001  
Model: All  
Subject: Null Modem  
Date: Jan. 31, 1984  
From: John Carter

---

### Problem Definition:

Host to terminal communication is incomplete due to mismatch in RS232 connectors.

Normally a host is configured to look like a data terminal and will connect to a modem. The same is true for most terminals. This is the case for the SGI terminals and workstations. Data sets (modems) transmit on pin 2 (Tx) and receive on pin 3 (Rx). This means that data terminals (including most computers) transmit on pin 3 and receive on pin 2. Other signals that are affected in this way are RTS/CTS and DTR/DCD.

### Solution:

Create (or buy) a null modem. A null modem is nothing more than an RS232 cable with some wires cross-connected as follows:

TERMINAL SIDE	TO	HOST SIDE	SIGNAL
1		1	Fr. GND
2		3	Tx
3		2	Rx
4		5	RTS
5		4	CTS
6		NC	DSR
7		7	Sg. GND
8		20	DCD
20		8	DTR

On the host side connect pin 6 to pin 8. On the terminal side connect pin 6 to pin 8.

All of this can be accomplished with the aid of a breakout box to first make sure that such a cable will actually work.

## SILICON GRAPHICS FIELD ENGINEERING

---

### TECHNICAL AIDS

Number: HOST-002  
Model: All  
Subject: DELNI Multiplexer  
Date: Dec. 03, 1984  
From: Greg Eskridge, Bob Lum

---

#### Problem Definition:

Digital Equipment Corporation manufactures a multiplexer device, which is used as an alternative to the Ethernet cable for centralized equipment. The nomenclature used for this device is DELNI, which represents "D" for (D.E.C.)/Ethernet Local Network Interconnect. Ethernet devices are connected directly to the DELNI box with the Ethernet drop cable. No Ethernet transceiver box is required on the DELNI.

It has been identified by Excelan that this configuration with the DELNI multiplexer and their board did not work and they have revised their boards to support both the Ethernet cable configuration and the DELNI multiplexer configuration.

#### Solution:

When installing SGI products to a DELNI multiplexer remove the Ethernet board (P/N 5000-173-2) from the machine and examine the board. It must be Revision-H and have a four pin oscillator crystal (Conner Windfield 20MHZ) on the board. With the components facing up and the edge connector facing to the right, the oscillator crystal is located at the top of the board, one chip location to the left of the 8088 microprocessor chip (top right-hand location).

If a two pin oscillator is encountered, the Ethernet board must be replaced. It will not work with the DELNI configuration.

# FIELD ALERT NOTICE

SiliconGraphics  
Computer Systems

FAN # soft/10

SHEET 1

## TITLE

Dial-in Modems

## MODELS AFFECTED

All workstations using a dial-in modem interface.

## ASSEMBLIES / DOCUMENTS AFFECTED

2000 Series Workstation Guide and 3000 Series Owner's Guide

## DESCRIPTION

The steps below show the necessary actions for a dial-in modem to work correctly:

[1] Use port ttym2. (Other ttym ports have been proven to have problems and ttyd ports should not be used with dial-in modems.)

[2] Use a straight-through 25 pin cable.

[3] /etc/inittab should look as follows if using ttym2 and a 2400 baud modem:

```
d2:23:respawn:/etc/getty ttym2 du_2400 none LDISC0
```

[4] /usr/lib/uucp/L-devices should look as follows for ttym2 and a 2400 baud modem:

```
DIR ttyd2 ttyd2 2400
DIR ttym2 ttym2 2400
```

[5] Ensure that /etc/gettydefs has an entry for the baud rate you are using.

[6] If using a Hayes modem or Hayes clone which uses the same command set, do the following procedure:

a) Disable the getty on the port you are calling into.

b) Use the 'fix' script attached to this FAN in the following manner:

```
% cat /dev/ttyd2 &
% fix -i 2
% kill <process id of cat command>
```

Note: The -i is the dial-in modem flag and the 2 is the designated port you are calling into.

(Cont'd)

ORIGINATED BY

Pamela Gordon

RELEASED BY

Lynne McDonald *LM*

RELEASE DATE

5/21/87

## DISTRIBUTION

PRODUCT SUPPORT

TECHNICAL SUPPORT

FIELD ENGINEERING

APPLICATION SUPPORT

THIRD PARTY

SYSTEM ENGINEERS

2011 Stierlin Road  
Mountain View, California 94043  
Telephone (415) 960-1980  
Geometry Hotline  
(800) 345-0222 (North America)

# FIELD ALERT NOTICE

**SiliconGraphics**  
Computer Systems

**FAN # soft/10**

**SHEET 2**

- c) Turn the modem off.
- d) Set the smart/dumb jumper to dumb.
- e) Turn the modem on.
- f) Enable the getty again on the port.

## **ACTION REQUIRED**

Inform customers with modem problems of these instructions.

## **DOCUMENTS ATTACHED**

Shell script to 'fix' the Hayes Smartmodem.

## **REFERENCE MATERIAL**

None

## **EFFECTIVE DATE**

Immediately

2011 Stierlin Road  
Mountain View, California 94043  
Telephone (415) 960-1980  
Geometry Hotline  
(800) 345-0222 (North America)

```
#!/bin/sh
```

```
# 'Fix' a Hayes Smartmodem
```

```
usage='usage: -i|-o portno'
```

```
if test $# != 2 ; then  
    echo "$usage"  
    exit 1  
fi
```

```
case $1 in
```

```
    #-i)cmd='&D2 S0=1 Q1 E0';;
```

```
    -i)cmd='&D2 S0=1';;
```

```
    -o)cmd='&D3';;
```

```
    *)echo "$usage"; exit 1;;
```

```
esac
```

```
# the device to fix
```

```
dev=/dev/ttyd$2
```

```
stty 2400 -echo -icrnl -ixon -isig < $dev  
set -x
```

```
echo -n '\rAT&FQ\r' > $dev
```

```
sleep 1
```

```
echo -n 'AT B1 &C1 &R1 S2=128 S18=5 S25=0 "$cmd" &W\r' > $dev
```

```
sleep 1
```

# FIELD ALERT NOTICE

**SiliconGraphics**  
Computer Systems

**FAN # io/3**

**SHEET 1**

**TITLE**

Mkboot problem on Hitachi 511-8 in GL2-W2.4 and 2.5

**MODELS AFFECTED**

Iris 2000 series

**ASSEMBLIES / DOCUMENTS AFFECTED**

none

**DESCRIPTION**

"mkboot" on the 2.4 and 2.5 releases doesn't make the image of partition 'g' correctly. The symptom is usually a "bad magic number" error when trying to boot the system after restoring from tape.

**ACTION REQUIRED**

The only thing affected is the partition "g" image. Since this is the same on all 2.4 and 2.5 systems, you only need to get a new image of partition 'g' installed and then it will work ok.

For a proper long term solution, the following two lines in the /etc/mkboot script should be edited:

```
gsize='expr $2 '*' 512'  
gcount='expr $1 / $2'
```

They should be changed to read:

```
# gsize='expr $2 '*' 512'  
gsize=60928  
# gcount='expr $1 / $2'  
gcount=4
```

There are several places in the script that this change should be made. Make sure you get all of them.

**SCR FILED**

SCR number 2464 has been accepted for this bug.

**DOCUMENTS ATTACHED**

none

**REFERENCE MATERIAL**

none

**EFFECTIVE DATE**

Immediately

**ORIGINATED BY**

D. Christopher Dunlap

**RELEASED BY**

Beat C. Poltera

**RELEASE DATE**

3/9/87

**DISTRIBUTION**

PRODUCT SUPPORT

TECHNICAL SUPPORT

FIELD ENGINEERING

APPLICATION SUPPORT

THIRD PARTY

SYSTEM ENGINEERS

2011 Sterlin Road  
Mountain View, California 94043  
Telephone (415) 960-1980  
Geometry Hotline  
(800) 345-0222 (North America)

# FIELD ALERT NOTICE

**SiliconGraphics**  
Computer Systems

<b>FAN # soft/8</b>
SHEET 1

<b>TITLE</b> Missing Files in 3.5 systems
<b>MODELS AFFECTED</b> 3xxx
<b>ASSEMBLIES / DOCUMENTS AFFECTED</b> none
<b>DESCRIPTION</b> IRIS 3000 series systems with 3.5 operating system are missing /usr/bir/nroff and other files associated with man and document processing.
<b>ACTION REQUIRED</b> Transfer files from 3.5r2 update tape to affected systems. See attached list for files to replace.
<b>DOCUMENTS ATTACHED</b> List of files missing from 3.5 systems
<b>REFERENCE MATERIAL</b> none
<b>EFFECTIVE DATE</b> immediately

<b>ORIGINATED BY</b> Archer Sully	<b>RELEASED BY</b> Lynne McDonald <i>LMCD</i>	<b>RELEASE DATE</b> March 30, 1987
--------------------------------------	--	---------------------------------------

DISTRIBUTION		
<input checked="" type="checkbox"/> PRODUCT SUPPORT	<input type="checkbox"/> TECHNICAL SUPPORT	<input checked="" type="checkbox"/> FIELD ENGINEERING
<input checked="" type="checkbox"/> APPLICATION SUPPORT	<input type="checkbox"/> THIRD PARTY	<input checked="" type="checkbox"/> SYSTEM ENGINEERS

2011 Stierlin Road  
Mountain View, California 94043  
Telephone (415) 960-1980  
Geometry Hotline  
(800) 345-0222 (North America)

ISO



usr/bin/checkmm  
usr/bin/diffmk  
usr/bin/eqn  
usr/bin/graph  
usr/bin/hyphen  
usr/bin/mm  
usr/bin/mmt  
usr/bin/mmt  
usr/bin/mvt  
usr/bin/neqn  
usr/bin/nroff  
usr/bin/ptx  
usr/bin/tbl  
usr/bin/tc  
usr/bin/troff  
usr/lib/eign  
usr/lib/macros/an  
usr/lib/macros/an6  
usr/lib/macros/mmn  
usr/lib/macros/mmt  
usr/lib/macros/vmca  
usr/lib/tmac/tmac.an  
usr/lib/tmac/tmac.an6  
usr/lib/tmac/tmac.m  
usr/lib/tmac/tmac.ptx  
usr/lib/tmac/tmac.v  
usr/lib/tmac/tmac.vcat  
usr/man/u\_man/man1/diffmk.1  
usr/man/u\_man/man1/eqn.1  
usr/man/u\_man/man1/hyphen.1  
usr/man/u\_man/man1/man.1  
usr/man/u\_man/man1/mm.1  
usr/man/u\_man/man1/mmt.1  
usr/man/u\_man/man1/mvt.1  
usr/man/u\_man/man1/neqn.1  
usr/man/u\_man/man1/nroff.1  
usr/man/u\_man/man1/ptx.1  
usr/man/u\_man/man1/tbl.1  
usr/man/u\_man/man1/tc.1  
usr/man/u\_man/man1/troff.1  
usr/pub/eqnchar  
usr/spool/lp/etc/lib/netprinterinterface  
usr/spool/lp/etc/util/addclient

# FIELD ALERT NOTICE

SiliconGraphics  
Computer Systems

FAN # gen/23

SHEET 1

## TITLE

SMD Board Configuration

## MODELS AFFECTED

2500 & 2500T Models

## ASSEMBLIES / DOCUMENTS AFFECTED

SMD Controller for the 474meg Eagle disk drive

## DESCRIPTION

Submitted by Gus Gessert and Greg Eskridge:

The SMD Boards, Rev 140A, are being shipped to the field configured for the manufacturer, not for SGI. Several changes in jumpers are required to make the board work in the Iris.

In some cases this would not be an issue because the FE only needs to match the settings with the board that is being replaced.

However, in our case, rev's lower than 140 require the SMD Manual in order to figure out the correct settings.

The current manual does not illustrate jumper W19 on appendix A-68.

## ACTION REQUIRED

Configure SMD boards to the correct configuration for the Iris.

## DOCUMENTS ATTACHED

SMD Controller Jumpers (Engineering Update)

Appendix A-68 Figure 10 of the SMD Manual

## REFERENCE MATERIAL

SMD Manual

## EFFECTIVE DATE

Immediately.

ORIGINATED BY

Ron Ludwig

RELEASED BY

Beat Poltera *B.P. 230*

RELEASE DATE

2/23/87

## DISTRIBUTION

PRODUCT SUPPORT

TECHNICAL SUPPORT

FIELD ENGINEERING

LOGISTICS SUPPORT

THIRD PARTY

SYSTEM ENGINEERS

2011 Sterlin Road  
Mountain View, California 94043  
Telephone (415) 960-1980  
Geometry Hotline  
(800) 345-0222 (North America)

## Interphase 2190 SMD Controller Jumpers

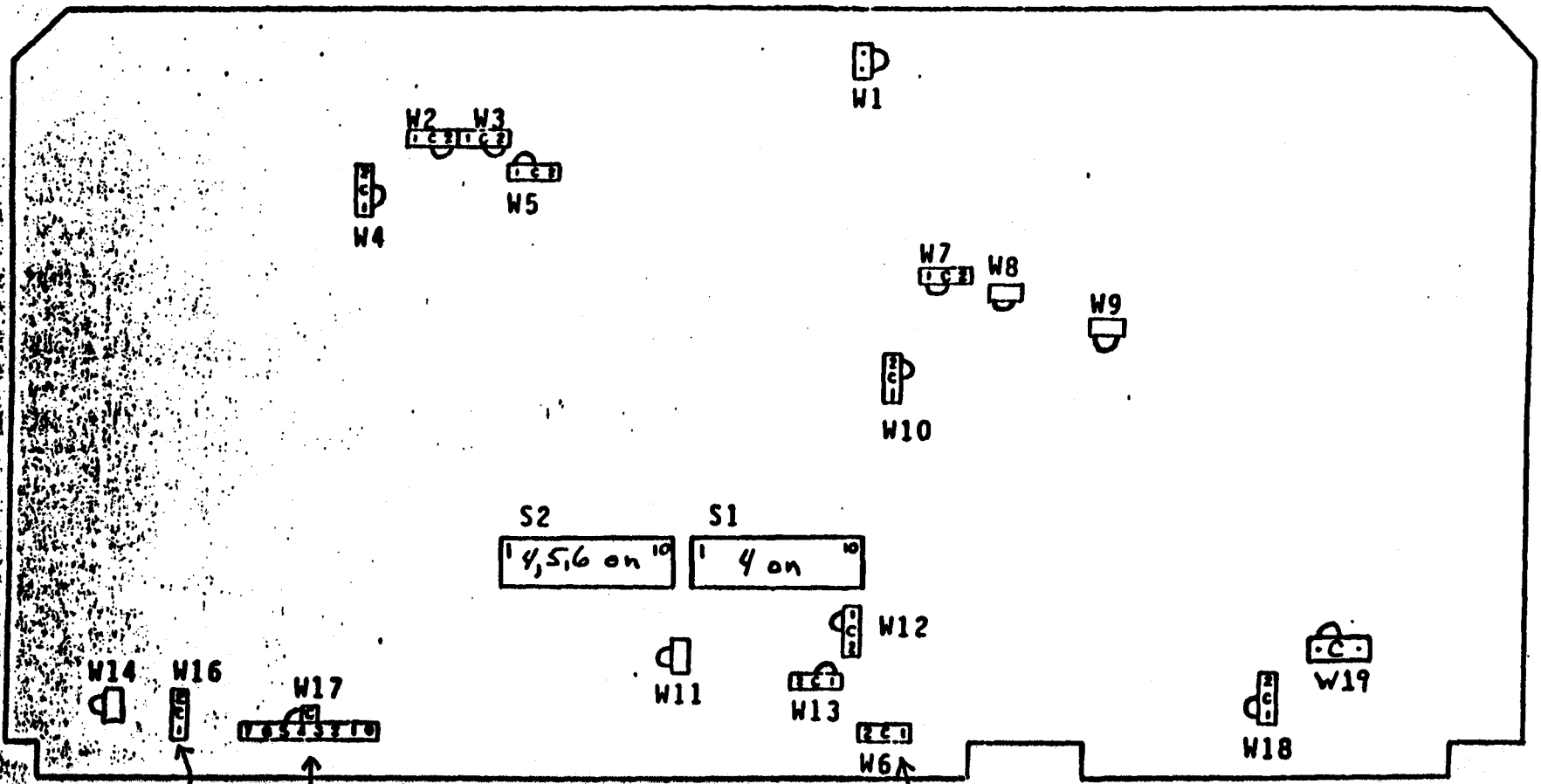
Chase Bailey  
Silicon Graphics, Inc.  
July 20, 1984  
Update: September 4, 1984

Changes have been made to the SMD Disk controller to change the address from 0x1020 to 0x7010. Switches S1 and S2 change.

Jumper	Function	Pins	IN	OUT	Strap	Description
W1	CBRQ	1-2	X			Monitors CBRQ
W2	Unit Selection	1-C-2			C-2	Units 0 through 3
W3	Unit Selection	1-C-2			C-2	Units 0 through 7
W4	Sector Flag	1-C-2			C-1	Sector Detect Flag
W5	Sector Flag	1-C-2			C-1	Sector Detect Flag
W6	-5 Volt Reg.	1-C-2		X		-5 Volts From the Bus
W7	Prom Size	1-C-2			C-1	2732 Proms
W8	Ram Size	1-2	X		1-2	2K Ram installed
W9	Buffer Size	1-2	X		1-2	4K or larger Buffer
W10	Buffer Size	1-C-2			C-2	4K or larger Buffer
W11	8/16 Bit I/O	1-2	X		1-2	16 I/O Address
W12	I/O Decode	1-C-2			C-1	Normal Mode
W13	I/O Decode	1-C-2			C-1	Normal Mode
W14	Priority	1-C-2			C-1	Serial Priority
W15	BCLK	1-2		X		Not Supplied to Bus
W16	Priority	1-C-2			C-1	BPRO Not Supplied
W17	Interrupt	C-0..7			C-5	Interrupt Level 5
W18	-5 Volt Source	1-C-2			C-1	-5 Volts From the Bus.

Switch	Function	ON	OFF	Description
S1	I/O Address	4	1-3,5-10	
S2	I/O Address	4-6	1-3,7-10	Address = 0x7010

FIGURE 10. JUMPER AND SWITCH POSITIONS  
APPENDIX A-68



wire jumper  
Com → 5

w6  
no jumper

W16  
3000 C→1  
2500T C→2

# FIELD ALERT NOTICE

**SiliconGraphics**  
Computer Systems

**FAN # gen/24**

SHEET 1

**TITLE**

Compatibility in Ethernet Installations

**MODELS AFFECTED**

All

**ASSEMBLIES / DOCUMENTS AFFECTED**

None

**DESCRIPTION**

There are differences between Ethernet 1.0, Ethernet 2.0 and IEEE 802.3 that require correct jumpering on the controller board and transceiver. The difference is in the SQE (Heartbeat) signal. Heartbeat is a test to ensure that the collision presence circuitry is working.

Ethernet Version 2.0 and 802.3 use Heartbeat, but Ethernet 1.0 does not. Additionally, IEEE 802.3 specifications state that 802.3 compliant repeaters must not be connected to transceivers that generate Heartbeat.

The Cabletron Transceiver and the Excelan controller both have jumpers for enabling and disabling the Heartbeat function. When doing an installation it's important to make sure that these jumper settings match.

If the Heartbeat settings of the controller and transceiver don't match, the system may get intermittent problems on the ethernet. Error messages of "Excessive Collisions" are common.

**ACTION REQUIRED**

The Cabletron transceiver's "LANview" LEDs indicate the state of the transceiver. A jumper option on the transceiver controls the Heartbeat function. See the attachment for a diagram of the jumper location on the Cabletron transceiver.

(Cont'd)

ORIGINATED BY

D. Christopher Dunlap

RELEASED BY

Lynne McDonald

RELEASE DATE

April 17, 1987

**DISTRIBUTION**

PRODUCT SUPPORT

TECHNICAL SUPPORT

FIELD ENGINEERING

APPLICATION SUPPORT

THIRD PARTY

SYSTEM ENGINEERS

2011 Sterlin Road  
Mountain View, California 94043  
Telephone (415) 960-1980  
Geometry Hotline  
(800) 345-0222 (North America)

# FIELD ALERT NOTICE

**SiliconGraphics**  
Computer Systems

**FAN # gen/24**

SHEET 2

On the Excelan 201 board, jumper "J11" controls the Heartbeat feature. If the jumper is installed, Heartbeat is disabled. If it's not installed, Heartbeat is enabled. Excelan 201 boards are shipped with the jumper J11 installed. See the attachment for a diagram of the jumper location on the Excelan 201 board.

#### ADDITIONAL NOTE

There may be a compatibility issue when IEEE 802.3 devices and Ethernet 1.0 devices are on the same network. If an installation is having unusual problems it may be worthwhile to pursue this as a possible cause. We do not have any better information at this time, but are working to define the situation.

#### DOCUMENTS ATTACHED

Diagram of Cabletron transceiver jumper  
Diagram of Excelan 201 controller board jumper

#### REFERENCE MATERIAL

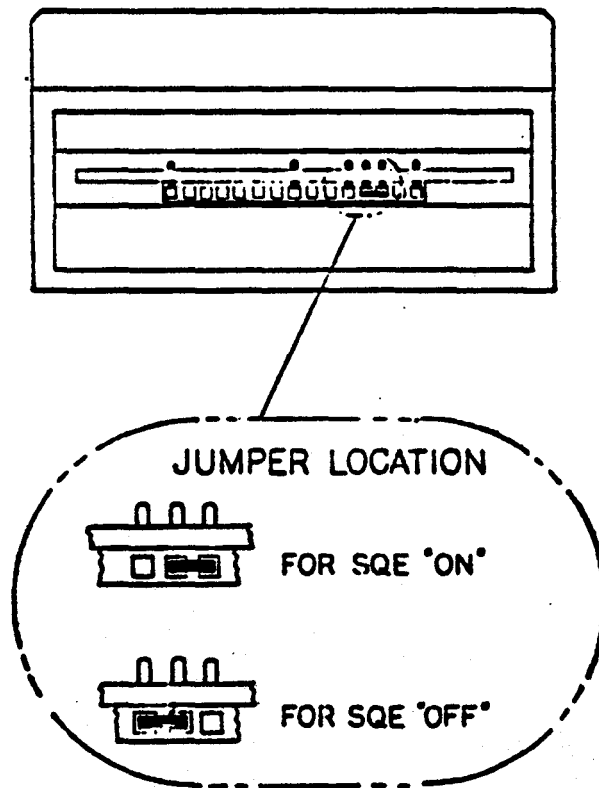
Excelan EXOS 201 Reference Manual  
Cabletron ST-500 Transceiver Manual  
Field Alert Notice Gen-20

#### EFFECTIVE DATE

Immediately

## CONFIGURING THE HEARTBEAT (SQE) TEST

- Remove the tap from the ST-500
- Look into the rectangle indent where the tap once was resident.
- Inside the rectangle the user will notice a 15 position receptical. Six of the positions contain metal contacts. Turn the ST-500 so that the bulk of the six contacts fall on the right hand side of the unit.
- To configure heartbeat (SQE) test on, move the jumper to the two contacts located to the right hand side of the three contacts which are located adjacent to each other.
- To configure heartbeat (SQE) test off, move the jumper to the two contacts located on the left hand side of the three contacts located adjacent to each other.



EXOS 201 Serial No., Assembly No.,  
and Revision Level stickers

LEDs: DS1  
DS2  
DS3

Ethernet Connector  
(16-pin IDH Type)

80186 CPU

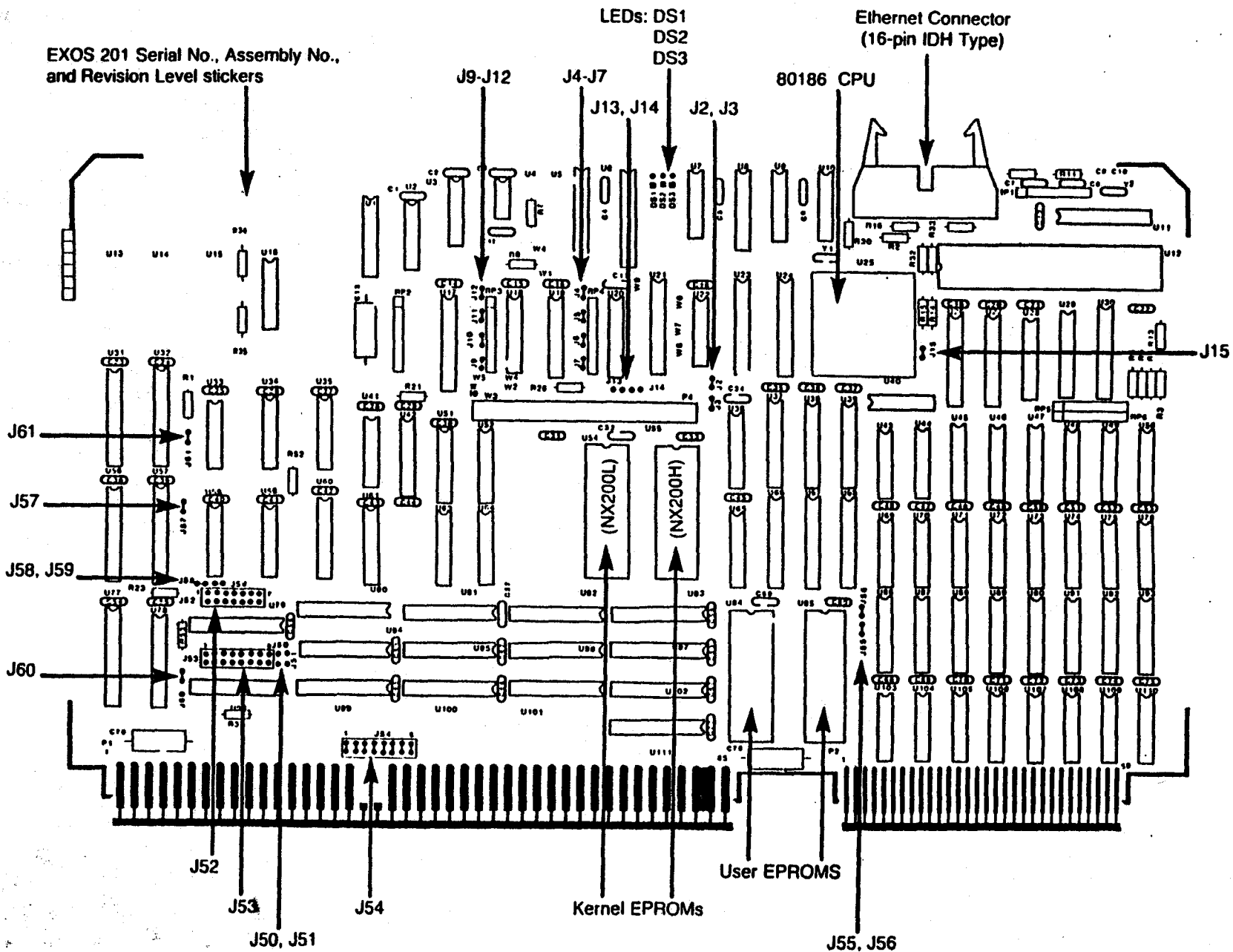


Figure A-1: EXOS 201 Component Layout



# FIELD ALERT NOTICE

**SiliconGraphics**  
Computer Systems

**FAN # gen/25**

SHEET 1

**TITLE**

ESD Clip

**MODELS AFFECTED**

All workstations with quarter inch Cipher tape drives.

**ASSEMBLIES / DOCUMENTS AFFECTED**

Tape Drive (Cipher 540-CT)

**DESCRIPTION**

Electro-Static Discharge is causing intermittently failures on some systems with Cipher 1/4" tape drives.

The symptom of the failure is that the system will hang when accessing the tape.

Installing a ESD Clip to the tape drive will eliminate this problem.

**ACTION REQUIRED**

Install the ESD Clip according to the attached procedure.

Refer to the two figures on the attached procedure for the proper placement of the ESD Clip.

**DOCUMENTS ATTACHED**

Procedure To Install The ESD Clip On The Cipher 540-CT

**REFERENCE MATERIAL**

None

**EFFECTIVE DATE**

Immediately

ORIGINATED BY

Ron Ludwig

RELEASED BY

*Greg Anders*  
Greg Anders

RELEASE DATE

June 3, 1987

**DISTRIBUTION**

PRODUCT SUPPORT

TECHNICAL SUPPORT

FIELD ENGINEERING

APPLICATION SUPPORT

THIRD PARTY

SYSTEM ENGINEERS

2011 Stierlin Road  
Mountain View, California 94043  
Telephone (415) 960-1980  
Geometry Hotline  
(800) 345-0222 (North America)

## PROCEDURE TO INSTALL THE ESD CLIP ON THE CIPHER 540-CT

1. Place the 540-CT on a flat surface and open the tape cartridge tray as in figure 1.
2. Remove the adhesive cover from the ESD clip.
3. Place the ESD clip on the tape cartridge tray as shown in figure 2.
  - A. The ESD clip should be attached to the tray at a point that is approximately  $3/4$ " behind the front of the cartridge tray.
4. Rub the ESD clip with a blunt tool. A tool similar in fashion to a popsicle stick.

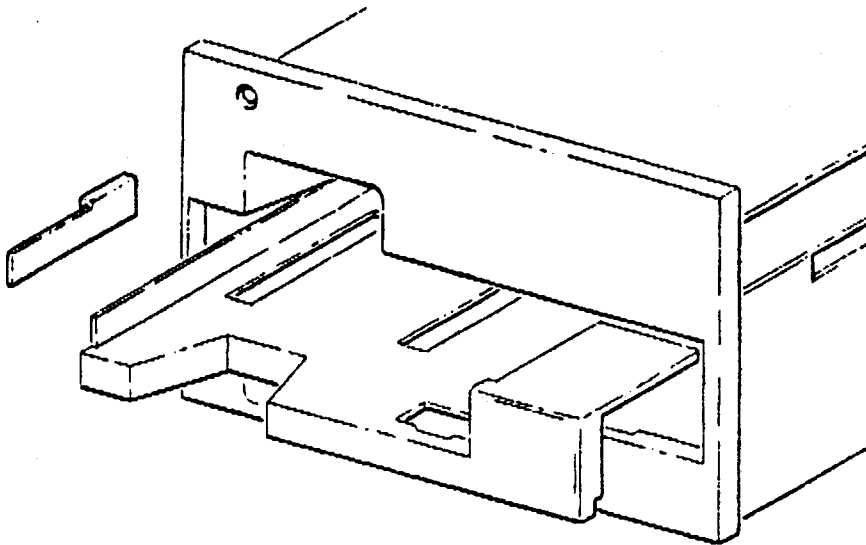


FIGURE 1

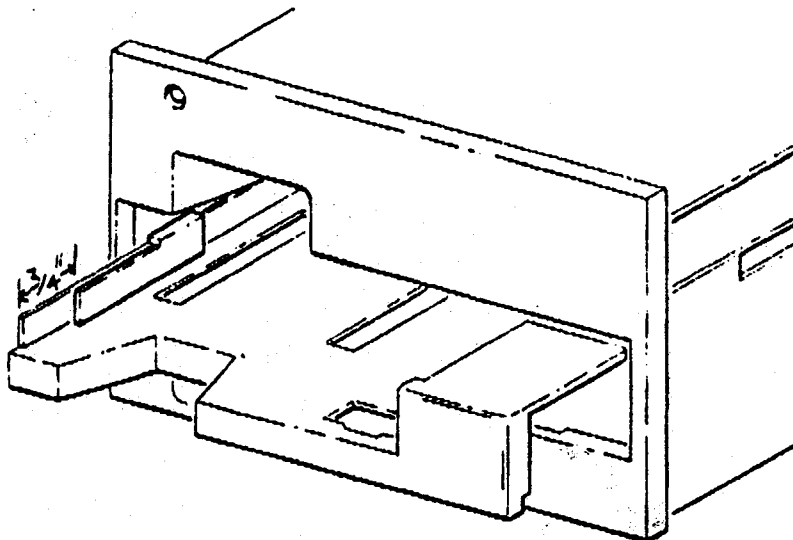


FIGURE 2

# FIELD ALERT NOTICE

SiliconGraphics  
Computer Systems

FAN # gen/26  
SHEET 1

TITLE  
CG1 Sync

MODELS AFFECTED  
All workstations with the Genlock RS-170A (D-GENR) Option

ASSEMBLIES / DOCUMENTS AFFECTED  
CG1 Board (5000-543-01)


DESCRIPTION  
The CG1 board currently supports one type of external synchronization signal. This is a TTL-level composite sync signal. Some customers may try to sync to an alternate form of external sync known as blackburst. The Lenco external sync generator supplies both comp sync and blackburst outputs. The CG1 board currently supports only the composite sync signal. Engineering is currently looking into the matter and will decide if modifications will be made to the CG1 board so that both types of signals will be supported.

ACTION REQUIRED  
Be aware of this potential customer problem.

DOCUMENTS ATTACHED  
None

REFERENCE MATERIAL  
None

EFFECTIVE DATE  
Immediately

ORIGINATED BY Herb Ebner	RELEASED BY  Greg Anders	RELEASE DATE July 1, 1987
<b>DISTRIBUTION</b>		
<input checked="" type="checkbox"/> PRODUCT SUPPORT	<input checked="" type="checkbox"/> TECHNICAL SUPPORT	<input checked="" type="checkbox"/> FIELD ENGINEERING
<input checked="" type="checkbox"/> LOGISTICS SUPPORT	<input type="checkbox"/> THIRD PARTY	<input checked="" type="checkbox"/> SYSTEM ENGINEERS

2011 Sterlin Road  
Mountain View, California 94043  
Telephone (415) 960-1980  
Geometry Hotline  
(800) 345-0222 (North America)

P Fields

1

## Review Exam 1: Lessons 2 thru 7

Revision A 10/23/86

96

### INSTRUCTIONS:

1. You have two hours to complete this exam.
2. **DO NOT** spend too much time on any one question.
3. **DO NOT ASSUME** any facts or conditions about any question. If you do not understand the question, ask your instructor for help before you attempt to answer the question.
4. Some of the questions may have more than one answer. Select the answer that is most correct.
5. The objective of this exam is to review the lesson information. Some of the questions you should be able to answer from recall, while others require that you refer to your class notes and/or supporting documentation.

While most of the questions will be on covered material, there will be some that require you to search your documents for the correct answer. For these questions, you will have to use the documents that you have been given, with your ability to extract the correct answer from them.

6. You may use your workstation and UNIX to help you determine the answer to any question.
7. When you complete this exam, you may leave the room.

EXAM 1

NAME:

1. The PROM monitor performs a rudimentary check of the memory system when a power-on reset occurs.

- A. True IM1 configuration - 4mb → f's?  
B. False Done By auto configuration

2. When is the process INIT started? (when you boot)

at Power on Reset to Multi user. + The last step of kernel boot process is the process program INIT 8/9.

3. Why does a new IRIS workstation stop in Single-User mode when the user executes the boot command?

because the way the initTab is set up (RState)  
+ IRIS Default level is Single user P12

4. A generic UNIX system offers several run levels. IRIS only uses three run levels. List the three run levels IRIS uses and give a brief explanation of each level.

1. (5) Single user mode
2. (2) - Run level when user is using XMS
3. (3) Run level when user is using TCP/IP

- 5. A generic UNIX system offers two device consoles: *systty* and *syscon*. These two consoles are linked together on an IRIS system. How do they become linked?

This is done by link command in file  
/etc/rc.s0.

P14

- 6. What configuration file holds the instructions (commands) for mounting the filesystems, starting the daemons and any other processes that the user may want run when the system is booted?

- A. /etc/rc
- B. /etc/inittab
- C. /etc/motd
- D. /etc/mnttab

P 17

- 7. How do you instruct UNIX to start *getty* processes against ASCII terminals NOT on ports 2-4 for XNS users?

XNSd daemon does this

P 18 L7

- 8. What is the minimum system that will boot UNIX? *No Graphics*

IP2, 4 meg of memory, disk controller, ASCII terminal

P 25

- 9. Show the command that you would use to boot the workstation from the second disk, assuming you have a second disk with a skeleton UNIX in partition a.

B, S11a: VMUNIX

P 27

10. Show the command that you would use to manually write the pattern a55aa55a into all memory locations in the third meg. (Use the long format)

FML 20000 a55aa55a 0 40000

11. Show the commands that you would use to read the first and last locations in the third meg. (use the long format)

Dm 20000 1

Dm 2FFFFFF 1

↑ long word must end with D, C, etc.

12. You have a 2400 Turbo workstation. You want to test the GF2 board. What diagnostic will you use?

A. g2

B. g21

C. g3

D. gf2

p 2 7.7  
we use 93

13. Show the command that you would use to load the bpcd diagnostic on a 2400 Turbo workstation.

Bp SIOF: /TMP / bpcd2

14. You cannot boot your system because your primary monitor's screen is dead because of some system failure. What simple trick could you use so that you could boot the workstation?

Pull off the Keyboard cable and Boot  
The first Terminal ASCII

15. When using the *bpcd* diagnostic, what command must you execute between individual tests?

INIT

16. If you have a failing 3030 system and you are not sure where the problem is located, what diagnostics should you run and show the suggested order of testing.

1. IP2 Diag
2. Bpcd 2
3. G3

17. Before you load the diagnostic *g21*, what must you do?

Disconnect the Keyboard & Hit Reset or Power on so you can  
use the ASCII Terminal

31

18. If the terminal you login at is NOT set for 9600 baud, and the system attempts communication with your terminal using 9600 baud, garbage will appear when the login message is received. What must you do to instruct the *getty* process to try another baud rate?

HIT Break Key

19. What board will hold the pixel data for address bit<sub>9</sub> when in Color- Map mode, single buffer?

- A. BP3 slot 12
- B. BP3 slot 13
- C. BP3 slot 14
- D. BP3 slot 15

P 19 L.C



20. What board will hold the pixel data for address bit 9 when in Color-Map mode, double buffer?

A. BP3 slot 12

B. BP3 slot 13

C. BP3 slot 14

D. BP3 slot 15

P22 46

21. While running *fsck*, you are informed that a file is defective. All *fsck* supplies you with is the *i-node* number. Also, the filesystem that was being checked was */usr*.

Show the UNIX command that you would use to find the file and report its pathname to you. The *i-node* number was 3216.

Find  $\wedge$  *INUM* 3216 -PRINT *for /usr*

*or*  
Find  $\wedge$  -user  $\wedge$  /usr  $\wedge$  -inum  $\wedge$  3216 -print

22. A file called *abc* has the following permissions set for it: 666. You want to change the permissions to allow the following:

Owner = Read and Write permission. 6  
Group = Read only. 4  
Other = Read only. 4

Show the complete command that you would use to set these permissions.

Chmod  $\wedge$  644  $\wedge$  ABC

23. Show the command and its arguments that you would use to copy a file from a host named *julie* to your workstation. The file pathname is *~/mark/folder/records/xyz*. (Show where any spaces in the command syntax would go using the up arrow character)

XCP  $\wedge$  Julie  $\wedge$  ~mark /folder /records /xyz  $\wedge$  XYZ

24. Show the command for restoring a single file called *lesson2* from the *usr* backup tape. The file lives in the directory */usr/people/rich/course*, and this is where it was when the *usr* backup tape was created.

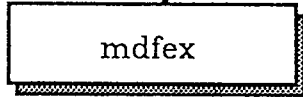
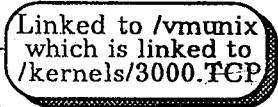
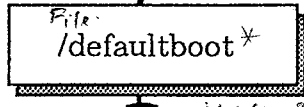
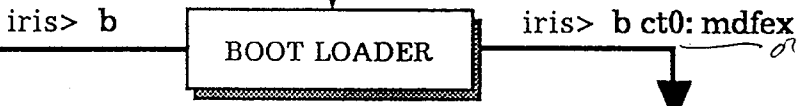
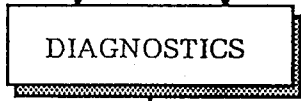
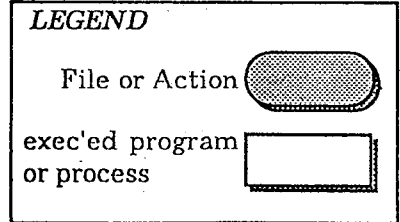
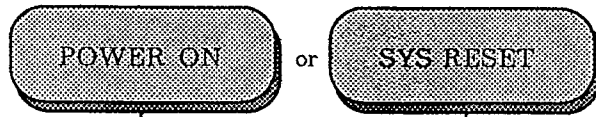
`cpio -i -vhdum1_n /usr/people/Rich/course/lesson2`

25. If you cannot boot from disk, what steps should you take before you start testing the system or replacing any boards in the system?

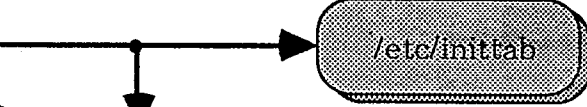
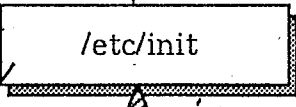
- p27 27
1. `LS_n B10a:` - List Disk 0
  2. `LS_n S10a:` List Disk 1 if you have it
  3. `b_n S11a: VMUNIX` Boot from Disk 1 if it configured
  4. `b_n CTO: S10X` Boot from Tape
  5. `N_n Name : VMUNIX` - Boot over net work

Great job!  
 (even if you  
 don't like  
 tests...)

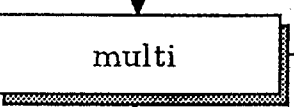
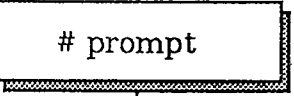
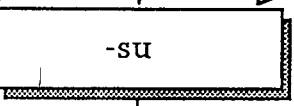
# UNIX STATE DIAGRAM



*Looks to see what hardware is there.*



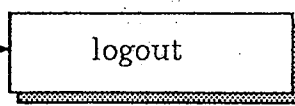
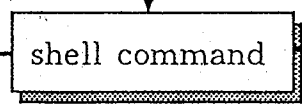
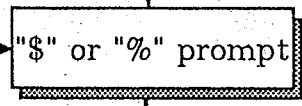
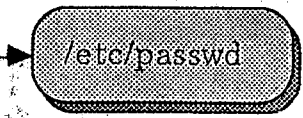
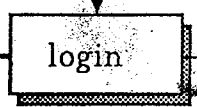
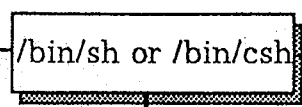
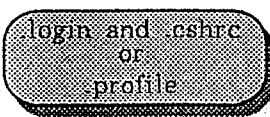
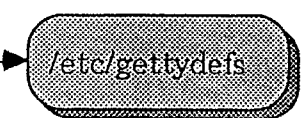
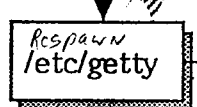
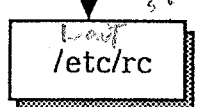
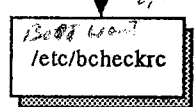
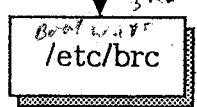
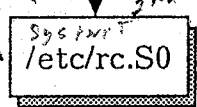
*Reads /etc/inittab Log you in as ROOT in single user mode.*



*1st with ab*

*don't run until multi*

*TTY type*



TERMINATE signal sent to init process

## UNIX STATE DIAGRAM: RC FILES

/etc/rc\* : 1. mounts Filesystems listed in /etc/fstab - credits /etc/crontab  
↳ 7. 17  
2. Starts up daemons - process that runs continuously

/etc/rc.s0\* : 1. Set up console LINK sysTTY to sysCON  
↳ 7. 14  
terminal Physical

↳  
/etc/brc\* : 1. check /etc/model for model #  
↳ 7. 15

/etc/bcheckrc\* : 1. FSCK  
↳ 7. 16  
2. Date

/etc/getty\* : 1. refers to /etc/getty defs to initialize Terminal (login prompt)

## UNIX STATE DIAGRAM: FILE DESCRIPTIONS

*function*  
/defaultboot\* :

1. contain the initial UNIX
2. linked to /usr + /kernel: 13000.XNS files
3. Spawns the INIT process

/etc/init\* :

1. auto configuration
2. Reads /etc/inittab
3. Single user mode - logs you in as root

/etc/inittab : 1. stores PC files L7, 14

2. activated when you change Run State (S, 2, 3)

/etc/gettydefs :

1. contains Band Rate + login prompt for 16 ports
2. starting point (label) from <sup>/etc/</sup>inittab File

/etc/passwd :

1. contain all system account information

# Contents

<b>LESSON 7: TROUBLESHOOTING LAB</b> .....	1
Boot Process/Exercise: General Information .....	2
Boot Process/Exercise: Initialization Process .....	3
Boot Process/Exercise: Micro Code Configuration .....	4
Boot Process/Exercise: Configuration Switches .....	5
Boot Process/Exercise: Kernel Boot Process .....	6
Boot Process/Exercise: Auto Configuration - page 1 .....	7
Boot Process/Exercise: Auto Configuration - page 2 .....	8
Boot Process/Exercise: Generic Init Process .....	9
Boot Process/Exercise: IRIS Init - Single User Mode .....	10
Boot Process/Exercise: IRIS Run Levels .....	11
Boot Process/Exercise: Multi Command .....	12
Boot Process/Exercise: Going to Multi User .....	13
Boot Process/Exercise: File rc.s0 .....	14
Boot Process/Exercise: Remove Mount Table .....	15
Boot Process/Exercise: Run Fsck .....	16
Boot Process/Exercise: Mount Filesystems .....	17
Boot Process/Exercise: Start Getty Processes .....	18
Boot Process/Exercise: Auto Boot to Multi Mode .....	19
Boot Process Summary - page 1 .....	20
Boot Process Summary - page 2 .....	22
TROUBLESHOOTING HINTS: System will not Boot .....	23
TROUBLESHOOTING HINTS: ASCII Terminal Port 2 .....	24
TROUBLESHOOTING HINTS: Minimum System to Run .....	25
TROUBLESHOOTING HINTS: Minimum System to Boot .....	26
TROUBLESHOOTING HINTS: Cannot Boot from Disk .....	27
TROUBLESHOOTING HINTS: PROM Monitor - page 1 .....	28
TROUBLESHOOTING HINTS: PROM Monitor - page 2 .....	29
TROUBLESHOOTING HINTS: Bit Plane Troubleshooting .....	30
TROUBLESHOOTING HINTS: Bit Plane Troubleshooting Map .....	32
Lab Bug Report .....	33
Lab Bug Report .....	34
Lab Bug Report .....	35
Lab Bug Report .....	36
Lab Bug Report .....	37



## **LESSON 7: TROUBLESHOOTING LAB**

The objective of this lesson is to allow each student to apply the knowledge learned and reinforce that learning by diagnosing induced failures on the training workstations.

- Each student must demonstrate that he/she can locate a majority of the lab bugs using any available documentation and/or tools.
- Each student must be able to locate, unaided by other students and within one hour, specific failures known as performance problems. He/she may ask general questions to collect necessary information (concerning the problem) from the lab instructor.
- Each student must demonstrate that specific diagnostics can be located, loaded and correctly run.



**Boot Process/Exercise: General Information**

This portion of the lesson attempts to document, explain and illustrate the IRIS workstation *Boot Process*.

The boot process occurs in a sequential manner which this document follows. The student will be required to perform several functions at his/her workstation during the lecture to reinforce several of the points covered.

It is **imperative** that each student remain in sync with the instructor and not wander off on his/her own during the exercise.

Several notes must be made:

- The demonstration was produced on a 3030 or 3130 workstation using Revision 3.5 software.
- The discussion is generic and meant to help the student understand what occurs during the boot process and look at the process as a *first line* diagnostic to be used to help isolate system failures.

The boot process supplies diagnostic information to the person who understands what occurs during the process.

- It is assumed the student has a basic understanding of UNIX.

**Boot Process/Exercise: Initialization Process**

From power-on reset to multi-user mode, several *software* processes initialize the workstation for use:

- Microcode - *from monit*
- Kernel Boot Process and Auto-configuration. *default boot - PCY*
- Init process *in task*

## Boot Process/Exercise: Micro Code Configuration

- **Power-on / Reset / Reboot**

These three functions cause the Prom Monitor (micro code) to perform a rudimentary system configuration.

- **The student should now execute the *Reboot* command**

IRIS (IP2 - Revision B) Monitor Version 3.0.7 December 20, 1985  
 Memory Size 4mb (Physical Map (1mb/bit) 0x0000000f) - 1mb  
 Configuration Switch: 0x0100  
 Multibus Window (2mb) at Megabytes 0 and 1.  
 Multibus accessible memory (1mb) begins  
 at Physical memory page 300,  
 at Virtual address 2000000.

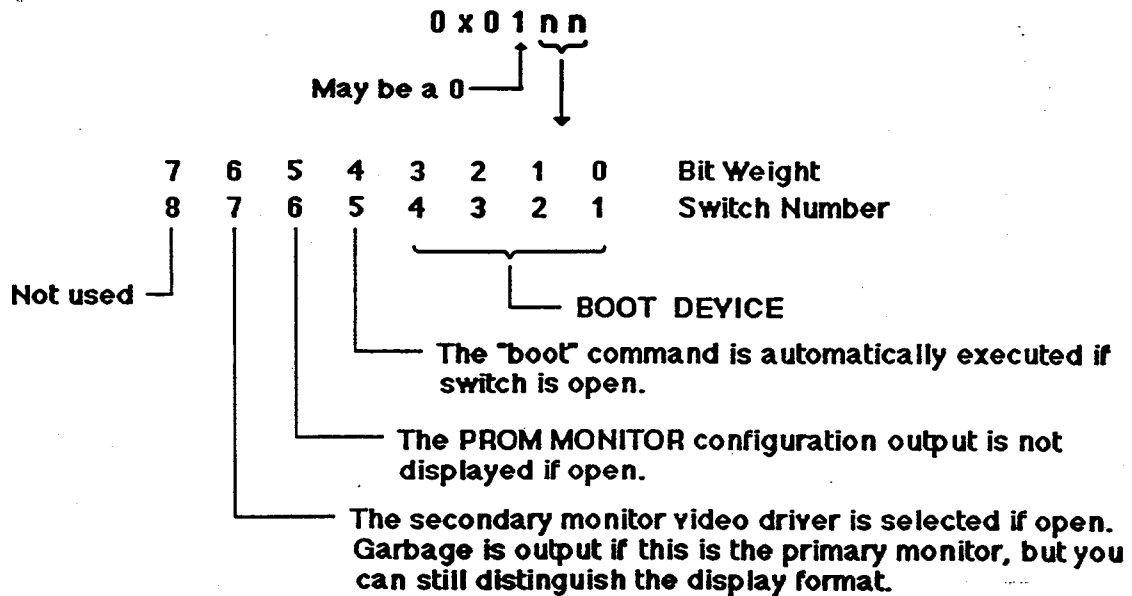
1 BIT = 1 mb  
 1 x 2 = 4  
 4 mb = 0x00000F  
 8 mb = 00FF

- **Micro Code *Diagnostics***

- The micro code reads the revision register of the IP2 board.
- The micro code reports the amount of memory that it can access. The mask (0x0000000f) is bit significant per meg.
- The micro code reports the settings of the rear panel switches.
- The remainder of the information is determined by jumpers installed on IP2 at manufacture time.
- If the PROM monitor cannot communicate with the primary console, it looks for an **ASCII terminal on port 2.**

### Boot Process/Exercise: Configuration Switches

- See page 3-9 of the IRIS 3000 Owner's Guide.
- The micro code reports configuration switch settings per the following table. (Closed switches are reported as zeros)



- You should set switches 1 and 2 to the open position and then depress *reset* or enter *exit*, followed by *b*.

What happens?

- You should now return switches 1 and 2 to the closed position, open switch 7 and execute *reboot*.

What happens?

Is screen output readable?

- You should now return all switches to the closed position and depress *reset*.

**Boot Process/Exercise: Kernel Boot Process**

- When the PROM monitor boot command is executed, the micro code loads the Kernel into memory and then transfers program control to the Kernel by jumping to a specified location from which Kernel program execution begins.

SGI Extent Filesystem  
Loading: si:0:defaultboot  
Text: 038318 bytes  
Data: 0113d8 bytes  
Bss: 024d7c bytes  
Jumping to load program ~ 20000400

- After the Kernel is loaded, the *Auto-configuration* process determines what devices are available and on-line.

**This process should be viewed as a diagnostic tool.**

- **You should now execute the boot command *b*.**

Configuration output is shown on the next page.

**Boot Process/Exercise: Auto Configuration - page 1**

SYSTEM 5 UNIX #0 [Wed May 7 04:49:59 PDT 1986]

(C) Copyright 1986 - Silicon Graphics Inc.

real = 4194304

kmem = 561152

user = 3633152

bufs = 819200 (max=16k)

dsd0 not installed

qic0 not installed

sii0 at mbio 0x07200 ipl 5

si0 (Hitachi 512-17 name: Hitachi 512-17) slave 0

si1 not installed

sf0 floppy (80/2/8) slave 2

siq0 at mbio 0x73fc ipl 5

sq0 (qic02 cartridge tape) slave 0

iph0 not installed

tmt0 not installed

ik0 not installed

nx0 (FW 2.5 HW 4.0) (0800.1400.3948) at mbio 0x7ffc ipl 2 *-FW 2.5*

fpa installed

lpen not installed

kernel debugger disabled.

root on si0a *where /etc is mounted*

swap on si0b. swplo=0 nswap=64000

INIT: SINGLE USER MODE *can't find /etc*

Welcome to the world of the  
IRIS personal workstation.

The SGI education department  
hopes that you have a very  
pleasant and informative stay  
with us.

**Boot Process/Exercise: Auto Configuration - page 2**

- The auto-configuration process performs rudimentary diagnostics on all devices that it can communicate with.

- If specific hardware or software errors occur, the *Kernel Debugger* is entered.

This is a tool that allows engineers to look into the software to determine what caused certain error conditions.

If your system is placed into the Kernel Debugger, enter `reboot <return>` to reboot the system.

- If communication with the primary terminal is lost during the process, error messages may be output to the ASCII terminal on port 2.
- 
- As a last step of the Kernel Boot process, a process (program) called *Init* is invoked.

**Boot Process/Exercise: Generic Init Process**

- UNIX systems always run in one state or another. The actions that cause the various states to exist are under control of the *init* process, which is the *first general process* created by the system at boot time. It reads the file */etc/inittab*, which defines exactly which processes exist for which run level.

**NOTE:**

Run level does not specify a process priority as the name may imply.

- See *INIT(1M)* for generic description of *init*.
- See *INITTAB(4)* for generic description of *inittab*.

The following page describes how *init* is used on the IRIS.



**Boot Process/Exercise: IRIS Init - Single User Mode**

- Init scans the file */etc/inittab* for the statement *initdefault* when it is initially invoked.  
If *initdefault* exists, the specified run level is set as the initial run level.
- If the specified run level is *single user* (which is the default for IRIS systems shipped), further scanning of *inittab* is not done.
- The user is informed that the system is in single user mode. (See page 7.7)

**INITTAB**

```
is:s:initdefault:#default state
s0::sysinit:/etc/rc.s0 1>/dev/console 2>&1#fix console
bc::bootwait:/etc/brc </dev/console >/dev/console 2>&1#bootrun cmds
bl::bootwait:/etc/bcheckrc </dev/console >/dev/console 2>&1#bootlog
rc::bootwait:/etc/rc 1>/dev/console 2>&1#run com
pf::powerfail:/etc/powerfail 1>/dev/console 2>&1#power fail
co::respawn:/etc/getty console co_9600 none LDISC0#console
d1::respawn:/etc/getty ttyd1 dx_9600 none LDISC0#serial 1
d2:x:respawn:/etc/getty ttyd2 dx_9600 none LDISC0#serial 2
d3:x:respawn:/etc/getty ttyd3 dx_9600 none LDISC0#serial 3
```

**Boot Process/Exercise: IRIS Run Levels**

- The second field of each entry in *inittab* (*rstate*) defines the run level.
  
- IRIS uses only three run levels:
  - **s** = Single user mode.
  - **2** = Run level when user is using *XNS*.
  - **3** = Run level when user is using *TCP/IP*.
  
- The run level is changed using the *telinit* command.
  - *Telinit* is linked to *init*. *To init 2 or 3*
  - When *telinit* is executed, *init* again accesses *inittab* and executes all processes that match the specified run level (i.e. *telinit 2*).

Processes with *rstate* fields that are empty, are always run.

**Boot Process/Exercise: Multi Command**

- IRIS default run level is *Single User*.
- The command *multi* will place the system into *multi user*.
- **The student should now execute *more /etc/multi*.**

*/etc/multi*

```

#! /bin/sh
# bring system up from single-user mode to multi-user mode

#
# Use run-level 3 for tcp systems, and run-level 2 for
other systems.
#
if test -x /etc/havetcp && /etc/havetcp; then

    telinit 3
else
    telinit 2
fi

```

- *Multi* contains two commands: *telinit 2* and *telinit 3*.
  - *Telinit 3* is executed if you are using *TCP/IP* Kernel. (The file */dev/EXOS/xmem* will not be empty)
  - *Telinit 2* is executed if you are using the *XNS* Kernel.
- Executing *telinit 2* would be the same as executing *multi*, the student should now execute *telinit 2*.

## Boot Process/Exercise: Going to Multi User

- When *telinit* is executed, *init* re-reads */etc/inittab* and reads the files *rc.s0*, *brc*, *bcheckrc* and *rc*.

These files all have *rstate* fields that are empty.

### INITTAB

```
is:s:initdefault:# ~(#)$Header: /ws/rel/src/etc/RCS/inittab,v 1.10
87/03/09 17:25:06 fong Exp $
s0::sysinit:/etc/rc.s0 1>/dev/console 2>&1
b0::bootwait:/etc/brc </dev/console >/dev/console 2>&1
b1::bootwait:/etc/bcheckrc </dev/console >/dev/console 2>&1
rc::wait:/etc/rc 1>/dev/console 2>&1
pf::powerfail:/etc/powerfail 1>/dev/console 2>&1
co::respawn:/etc/getty console co_9600 none LDISC0
d1::respawn:/etc/getty ttyd1 dx_9600 none LDISC0
d2::respawn:/etc/getty ttyd2 dx_9600 none LDISC0
d3:x:respawn:/etc/getty ttyd3 dx_9600 none LDISC0
t0:3:respawn:/etc/getty ttyT0 dx_9600 none LDISC0
t1:3:respawn:/etc/getty ttyT1 dx_9600 none LDISC0
t2:3:respawn:/etc/getty ttyT2 dx_9600 none LDISC0
t3:3:respawn:/etc/getty ttyT3 dx_9600 none LDISC0
t4:3:respawn:/etc/getty ttyT4 dx_9600 none LDISC0
t5:3:respawn:/etc/getty ttyT5 dx_9600 none LDISC0
t6:3:respawn:/etc/getty ttyT6 dx_9600 none LDISC0
t7:3:respawn:/etc/getty ttyT7 dx_9600 none LDISC0
```

**Boot Process/Exercise: File rc.s0**

To allow you to use console

- The student should now login as *root* and change directories to */etc*.
- The student should now execute *more rc.s0*.

Shell script  
rc.s0

**rc.s0**

#!/bin/sh

# /etc/rc.s0

# This script is run only during the read of inittab at boot time.

# This script resets the system console to the actual physical

# console as defined by /dev/systty

rm -f /dev/syscon

ln /dev/systty /dev/syscon

Physical console

linked

- *rc.s0* links */dev/systty* and */dev/syscon*. The *rstate* field *sysinit* (in inittab) insures that this occurs first.

- */dev/systty*: UNIX virtual console.
- */dev/syscon*: UNIX physical console.
- */dev/console*: IRIS physical console.

Enables you to have control at console

- On IRIS, all three console types are linked together.
- The student should execute the following commands:
  - `cat /etc/motd > /dev/console`
  - `cat /etc/motd > /dev/syscon`
  - `cat /etc/motd > /dev/systty`

## Boot Process/Exercise: Remove Mount Table

- *Init* reads the file */etc/brc*, which does the following:
  - Removes the file */etc/mtab* (mount table).
  - Checks for the system model number (*/etc/model*) and if it does not exist, prompts the user for the model number.
- Since the *rstate* field in *inittab* for this command specifies *bootwait*, this process will be started and the system will wait for it to terminate. When it dies, it will not be restarted. *only die if time*
- The student should now execute *more brc*.

**Boot Process/Exercise: Run Fscck**

- File checker*
- **Init** reads the file `/etc/bcheckrc`, which does the following:

- Checks the date and prompts the user if it is incorrect.
- Runs **Fscck**.

*etc/FSTab*  
**Fscck** will access the file `/etc/checklist` for the filesystems it must check.

- The student should now execute the following commands:

- more `bcheckrc`
- more `checklist`

*FSTab*

Boot Process/Exercise: Mount Filesystems

*mount -fstab!*

- **Init** reads the file **/etc/rc**, which does the following:
  - Creates a new mount table file **/etc/mtab** using the entries in the file **/etc/fstab**.
  - Mounts the filesystems per the file **/etc/mtab**.
  - Starts the daemons: update, cron, xnsd, lpd and lpsched.
  - Starts any other processes that are specified in **/etc/rc** (i.e. initialize a printer port).

● The student should now execute the following commands:

- more rc
- more mtab
- more fstab

*RC.local is used for local changes*

*/etc/rc.local. is there just make changes here*

*/etc/mtab  
What file system  
Have been mounted*

*/etc/fstab*

*What file system is mounted*

*containing*

*1 dev1 s10A 1  
1 dev2 s10F 1 USA*

*Files should look like*

*1 is single user mode it is  
Not mounted until fsck is done*



**Boot Process/Exercise: Start Getty Processes**

- The last step *init* performs, is the starting of the *Getty* processes against any attached ASCII terminals on ports 2-4.
- Any ASCII terminals not on ports 2-4 are handled by the *xnsd daemon*. That is why there are no entries for these ports in the *inittab* file.
- *TCP/IP* uses a different type of driver, therefore *inittab* has entries for the *TCP/IP* ports, 1-8.

**Boot Process/Exercise: Auto Boot to Multi Mode**

- As a last exercise, you will modify your workstation to boot into *multi user* from a *power-on/reset* condition.
  - Edit */etc/inittab*, and replace the **s** with a **2** in the second field of the first line.
  - Set rear panel configuration switch **5** to the **open** position.
  - Execute **reboot** and watch what happens. *don't have to*
- Restore configuration switch **5** to the **closed** position and edit */etc/inittab*, replacing the **2** with the **s**. *↙*

## Boot Process Summary - page 1

### Sequence of events:

- **Reboot or reset:** The micro code reads IP2 revision level, sizes memory and reads rear panel configuration switches.

If the micro code cannot communicate with the primary console, control is switched to the ASCII terminal on port 2.

- **If** configuration switch **5** is open, the micro code executes the boot command.

**Else**, the PROM monitor is entered, from which the boot command must be executed manually.

- The micro code looks for the file *defaultboot* (or another boot file that was specified with the boot command) on the boot device and loads the boot program (Kernel) into main memory.

- The Kernel then performs the *auto-configuration* process.

If specific types of errors are detected during the configuration process, or communication is lost with the primary console, error information is sent to the ASCII terminal on port 2 and the process is terminated.

- After the auto-configuration process completes and no errors have been detected, the Kernel invokes the *Init* process.

- *Init* scans the file */etc/inittab* for the statement *initdefault*.

**If** *initdefault* is present, *init* places the system into the specified run level. (IRIS default is single user)

**Else**, the system prompts the user to enter a run level.

**If** the default run level is **2** or **3**, *init* reads all configuration files in */etc/inittab* that match the run level and executes all specified processes.

**Else**, the boot process halts with the system in *single user* mode. The user must then

execute **multi** to continue the bring up process.

## Boot Process Summary - page 2

Assuming *initdefault* specified run level 2 or 3, or the user executed *multi*, the process continues:

- Upon issuance of a run level change, *init* re-reads the file */etc/inittab* and reads the files: *rc.s0*, *brc*, *bcheckrc*, *rc* and any other files that match the run level.

Commands contained in these configuration files are executed in an order specified by the *action* field of each entry in *inittab*. (i.e. *sysinit*, *bootwait*, *wait*)

- *rc.s0* links */dev/systty* and */dev/syscon*.
- *brc* removes the old */etc/mnttab* file and checks for the system model number.

If the model number is present in the file */etc/model*, the process continues.

**Else**, the user is asked to enter the model number.

- *bcheckrc* checks for the date and prompts user to answer if the date is correct and if *FSCK* is to be run.

If *FSCK* is to be run, the file */etc/fstab* is read to determine what filesystems should be checked.

- *rc* creates a new mount table file */etc/mstab*, mounts the specified filesystems, removes locks and logs, starts daemons and anything else the user specified in the *rc* file.
- *Init* starts a *getty* process for each ASCII terminal specified in the file */etc/inittab*.
  - The *login* prompt is displayed on all on-line terminals and the system is in *multi-user* mode.

**TROUBLESHOOTING HINTS: System will not Boot**

If your system does not boot and/or enter multi-user mode in a normal way, or your customer is complaining various terminals were not configured, or filesystems were not mounted, etc., before you start system troubleshooting you should always ask your customer if he/she changed any configuration file just prior to the failures.

i.e.

.login *... mail + account (csh)*

.cshrc

.profile *- sh Bourne*

bcheckrc

brc

checklist (In Rev. 3.4 and earlier)

crontab *- Set Daily Message*

fstab (Replaces rc.fs in Rev. 3.5) *- Files System should be mounted*

gettydefs *- Band Rate - Parity*

group *-*

inittab

mnttab (In Rev. 3.4 and earlier)

mstab (Replaces mnttab in Rev. 3.5) *- File mount*

model *of det*

multi *- xNA - Tepp D*

passwd *- account*

rc

rc.fs (In Rev. 3.4 and earlier)

rc.s0

rc.tcp

rc.xns

sys\_id *- system name*

termcap *- supported Terminal*

ttytype *- what type of terminal*

**Remember, the key question is: Did you change any configuration files?**

## TROUBLESHOOTING HINTS: ASCII Terminal Port 2

- Remember, if the micro code or Kernel **cannot** communicate with the primary console, they look for an ASCII terminal on port 2 to output normal and error information.
- Know what the normal boot output should look like for each of the systems you maintain. A good idea is to make a copy of the auto-configuration output and keep it on site. (Be prepared)
- Know what the system acronyms mean.

**i.e.**

GF2 - Geometry Engine Board

IP2 - Processor Board

FBC - Frame Buffer Controller

**TROUBLESHOOTING HINTS: Minimum System to Run**

- If your system is experiencing Bus Errors or the system is dead:
  - The minimum system that will still run Flight is:
    - IP2
    - 4 meg memory
    - Disk Controller
    - DC4
    - UC4
    - GF2
    - 2 bit planes (slots 16 & 17)
  - If Flight or system is still failing after establishing minimum system, try two other BP3's in slots 16 & 17.
- You can move BP3's and memory electronically by changing the board configuration switches. The following is a quick approach to possible bit plane problems if the boot process or Flight are failing:
  - Back out all bit planes except slots 16 and 17. If the system still fails, back out 16 and 17 and install 14 and 15 (with power off). Change the edge switches to reflect the first two boards.

If the system still fails, the above procedure should eliminate the bit planes as the probable cause, except for possible cable problems.



**TROUBLESHOOTING HINTS: Minimum System to Boot**

- If you have a system with an FP1 and optional memory boards, the FP1 and all memory boards except the first 4 meg can be backed out and the system will still boot up. (You do not have to recable the boards)
  
- You can boot the system with GF2, UC4 and DC4 backed out if you have an ASCII terminal on port 2.
  
- You **cannot** boot if you have a hole on the multibus between IP2 and the Disk Controller.  
  
    i.e. ENET board backed out.
  
- You could back out the Disk Controller and try booting over the network, but it will fail when the disk drive is accessed, but at least it will boot to a point.

**TROUBLESHOOTING HINTS: Cannot Boot from Disk**

If the system will not boot from disk, the following may help you isolate the problem.

- Can you list the contents of disk 0?

**ls hd0a:**

- If you have a second disk, can you:

- List the contents of disk 1?

**ls hd1a:**

- Boot from the second disk?

**b hd1a:vmunix1**

The above assumes the second disk has been configured per section 6.2.3 of the 3000 Owner's Guide (3 partitions). Partition **a** is loaded with the *skeleton* root system per section 4.4.3 of the 3000 Owner's Guide.

- Can you boot from tape?

**b ct0:sifex**

- Can you boot over the network?

**n .julie1:vmunix**

The above command boots the education systems. Your account will have a different host name and may also have a different boot program name.

### TROUBLESHOOTING HINTS: PROM Monitor - page 1

If you are experiencing system problems and want to verify if memory can be written into and read from, the PROM Monitor offers several commands that allow basic testing of the memory system.

The commands available are listed below with descriptions as to their functions:

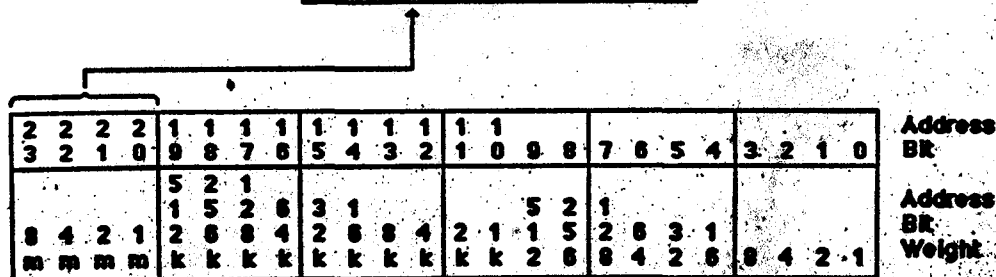
*6 16 32 bits*  
**fm{b|w|l} ADDR, VALUE, [INCR] [CNT]**  
 fill memory as byte, word or long stating at ADDR, with initial VALUE, incrementing VALUE by INCR for CNT times. (INCR defaults to 0; CNT defaults to 1)

**dm{b|w|l} ADDR [CNT]** display memory as byte, word or long. (CNT defaults to 1)

**em{b|w|l} ADDR** edit memory interactively as byte, word or long.

Memory is addressed with a 24 bit address. The following table shows the address ranges for each meg and the card that contains the locations.

MEG	ADDRESS	CARD
1	000000-0FFFFF	1
2	100000-1FFFFF	1
3	200000-2FFFFF	1
4	300000-3FFFFF	1
5	400000-4FFFFF	2
6	500000-5FFFFF	2
7	600000-6FFFFF	2
8	700000-7FFFFF	2
9	800000-8FFFFF	3
10	900000-9FFFFF	3
11	A00000-AFFFFF	3
12	B00000-BFFFFF	3
13	C00000-CFFFFF	4
14	D00000-DFFFFF	4
15	E00000-EFFFFF	4
16	F00000-FFFFFF	4



## TROUBLESHOOTING HINTS: PROM Monitor - page 2

Two notes about PROM monitor memory commands:

- If you give argument values that the PROM monitor does not like, a bus error may occur or the PROM monitor goes into a halt (see "J" box halt light) or loop. If this occurs, just depress reset.
- When you display memory ( dm ), if the bytes of data are valid ASCII characters, the characters are displayed next to the memory data.
- Addressing can be performed using three formats:
  - Byte(8) - last digit can be 0-F.
  - Word(16) - last digit must be 0, 2, 4, 6, 8, a, c, e.
  - Long(32) - last digit must be 0, 4, 8, c

The following illustrates several examples of memory reads and writes:

*had*  
 32 BIT - sequential  
 16 BITS -  
 8 BITS -

Command	Description
<i>Full memory</i> fml 0 0 0 <del>fff</del> <sup>40,000</sup>	Clear first meg.
dml 0 100	Read <sup>100</sup> longwords starting at location 0.  <Return> to keep reading, q to terminate read.
fml 0 0 4 <del>fff</del> <sup>40000</sup>	Write address in address for longwords in first meg.
eml 0	Edit (change) first location and continue sequentially until you depress q.
	A bug in the PROM monitor will not let you write an all F's value (FFFFFFFF) into a single location; but the command fml can write an all F's value.

## TROUBLESHOOTING HINTS: Bit Plane Troubleshooting

1. Login as **guest**.
2. `cd /usr/people/mexdemos/bin.`
3. Execute **mex**.
4. Execute **showmap** (draw the largest possible window).
5. **Push** showmap to the background.
6. Execute **cedit** (sweep a small window in upper right corner).
7. `cd /usr/people/mexdemos.`
8. Execute **interp** (sweep a small window below *cedit* window).
9. Execute **demomakemap**.
10. **Push** the console window to the background.
11. Using *cedit*, make the third square (bottom line left) black.  
(Place the cursor over the square, and press the left mouse button.)
12. Using *cedit*, make the last square (top row right) black.  
(Place the cursor over the square, and press the left mouse button.)
13. Using *interp*, mark the third and last squares (left mouse button).  
(Place the cursor over the square and press the left mouse button.)
14. Using *interp*, interpolate the color range between the third and last square. (Press the middle mouse button).

NOTE: If any bit plane is broken (hot bits) you will see spots.

15. You can repeat steps 11-14 using white to check for cold bits.

**NOTE 1:**

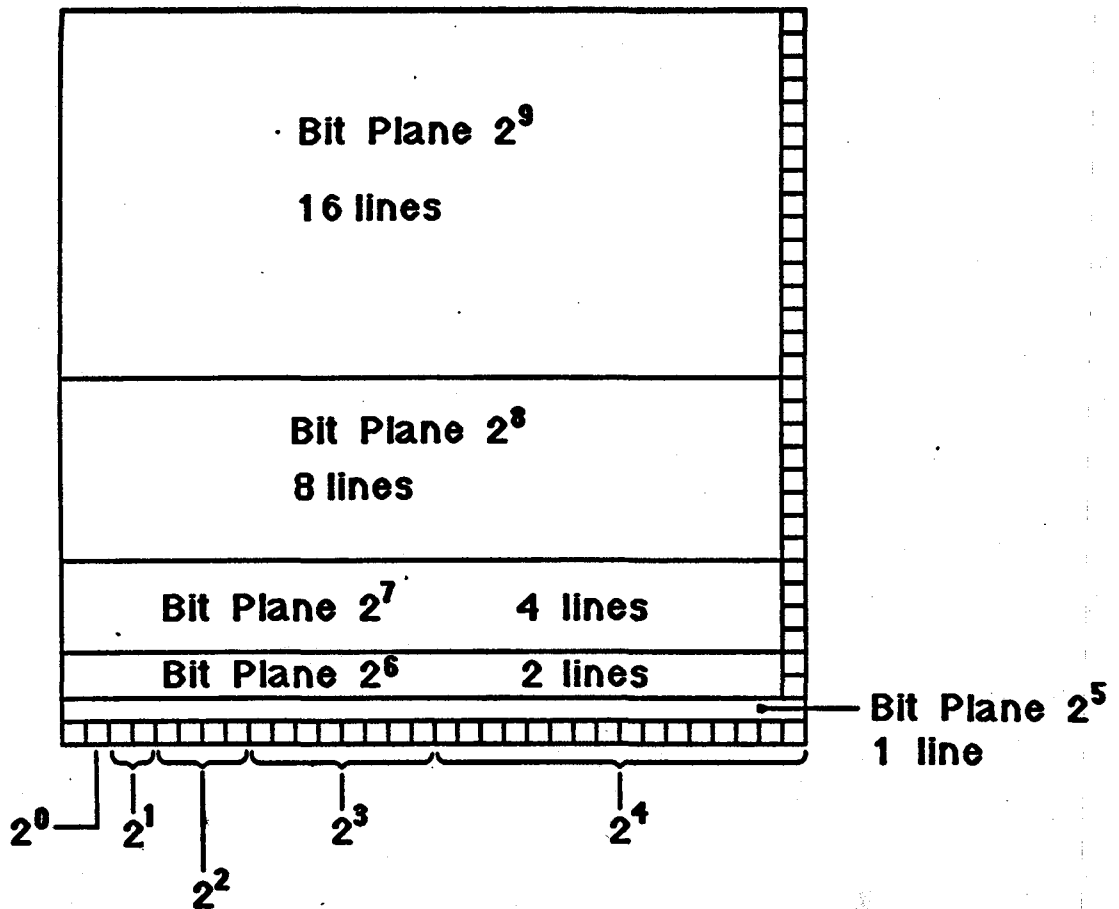
Showmap runs in *single buffer mode*. If you open a window that uses *double buffer mode* (i.e. jet), then Showmap will also run in double buffer mode changing the way that the bit planes are addressed.

**NOTE 2:** Additional information regarding **cedit**, **interp**, and **showmap** may be found in Vol. 1A of the UNIX Programmer's Manual.

**NOTE 2:** Use the shell command **gclear** to restore the graphics screen display after completing each procedure. See next page for a map of the bit planes if errors can be seen.

**TROUBLESHOOTING HINTS: Bit Plane Troubleshooting Map**

If you can detect areas of the screen that are not correct using the procedure on the previous page, then use this chart to determine what bit plane is defective.



**Pop up menus are bit planes 2<sup>10</sup> and 2<sup>11</sup>**

## LESSON 7: TROUBLESHOOTING LAB

## Lab Bug Report

Student / Team name: *Miles/Parks*Bug Number: *1*

## What were the symptoms?

*System won't boot.  
Main Monitor won't work.  
Default to Terminal*

## What component would you replace?

*1M1 jumper off*

## List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

*1 do 1M1 Have to Have Terminator*

*2 559A mem starts writing*

*Prom monitor says no memory installed*

*Wrote memory - will not write 0 - 255 59A*

*Diags will not load*

*Switches OK*



**Lab Bug Report**Student / Team name: *mike / Parks*Bug Number: *2*What were the symptoms? *can't execute commands in student account.***What component would you replace?***would change .login file to include a path for commands ex path = ( /bin, /usr/bin, etc, /usr/etc )**\* → Tell user not to use vi editor or any editor***List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.***login as student - could not do LS, PWD, PS, who, whoami, where - su, ReBoot.**- But could cd + logout / exit*

**Lab Bug Report**Student / Team name: *Parkes/miles*Bug Number: *3*      *5/25/88*What were the symptoms? *can login to student account.**Fstab Does NOT Have Dev*

What component would you replace?

*add device to Fstab,**/dev/sioF<sub>n</sub> /usr, rw=rw=  
/dev/lb10F<sub>10</sub>X0*

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

*Tried to access student / then any usr. Demos, guest**Fsck did NOT check /usr.**check RC Files*

## Lab Bug Report

24-5-88.

Student / Team name: MILES/PARKS

Bug Number: 4

What were the symptoms? BOOT PROCESS HALTS.

System Boots. Part way Then move to DX1 Terminal  
 System Fails Bpctest2 - (GFTest) R/W - all FFFF'S  
 Expect a single B.  
 11 wont load G3

What component would you replace?

GF2 Bd.

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

Swapped console

wrote mem OK

Ran Diag Bpctest2 - Failed GFTest  
 " " G3 - wouldn't load.

**Lab Bug Report**Student / Team name: *miles/parks*Bug Number: *5*What were the symptoms? *Graphics Display Problem**Screen looks jumpy*

What component would you replace?

*Cable - Top one Now*

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

*Ran Demos - arch, cube, Flight**Ran Diags - Bpcd 2**Stripe Test Failed**C2(01) Bad BIT Plane**Slot 12**0000**Swapped switches**Error stayed with C2(01) - Bottom cable**Swapped cable**It moved to (23) Top cable*

## Lab Bug Report

Student / Team name: Miles Parks

Bug Number: mx 6 / 5 / 25 / 88 09100

What were the symptoms?

System Booted - ok  
 Flight Looked ok  
 user still complained.

What component would you replace?

DC 4

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

- 1 Booted - ok
- 2 Ran Flight - ok
- 3 Ran IP2 Diag - ok
- 4 Ran Bpcd2 - Failed - Stripe Test
- 5 Ran g3 - ok
- 6 Restricted Boards + Cables.
- 7 BpcTest Fails - Stripe Test - 24 Errors / 12 Errors - 3Buf  
 DcRTST - 2 Errors.
- 8 Stripe Test - Bud Planes = F0FFFF - Both cables
- 9 DcRTST. VFSM Expected 0 Read Back 87  
 FF " " 87

Lab Bug Report

Student / Team name: *miles/Parks*

Bug Number: *1 A* *new Bug* *5/25/88*

What were the symptoms?

*graphics - Flight Fails*  
*ARCH Fails*

*white vertical lines on screen.*

What component would you replace?

*Replace BP3 Slot 13*

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

*Replaced cables List of 1*

- 1 Ran Flight*
- 2 Ran arch*
- 3 Ran Bpcd2 - Plane 19 write error, 0010 0000 6060 0000*  
*-D1 - Slot 13*

*Ran Stripe Test - D1(1) Failed*

*Swapped switches 12 → 13*

*Ran - Bpcd2 again 1B Failed*

*Ran - Stripe Test D3(1) Failed*

*Replace Bd at 13*

**Lab Bug Report**Student / Team name: *Parks/miles*Bug Number: *6A 5/25/88*

What were the symptoms? *System error message at Boot*  
*Defaulted to Terminal*  
*Fbc est: cant Reset GE PIPE*  
*Fbc Dead meat*

What component would you replace?

*GF2*

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

*Rm - Bpctest GF2 - GE's missing - GF2 Failed 1err.*

*93 - wont load*

*GFTest - GE's or passers missing*

*E19 - Gone!!*

## Lab Bug Report

Student / Team name: Parks/Miles

Bug Number: 2A 5/25/88

What were the symptoms?

Try going into multi-user mode.

v

What component would you replace?

Found /ETC/RC.Local with a Reboot command  
imagine that!!

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

You see unmount as last command.

Look at user files

- mTab
- FS Tab
- fsck

Then more RC Files

Notice pat mods in RC.local



### Lab Bug Report

Student / Team name: *Miles / Parks*

Bug Number: *4A*

What were the symptoms? *another Bummer Display.*

What component would you replace?

*Replace UC4 Bd.*

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

*BCPDZ*

*Rectangle Test - Failed*

*Pixel test - 4220 Error*

*Trapezoid Test - Failed*

*G-3 - NO Display*

*Write enable code FFFFFFFF or 0 FFFF d 11 d  
1101*

*Ran\_BPTest - Dead*

*Can not load Bit Planes*

*Stripe Test - Dead.*

**Lab Bug Report**Student / Team name: *miles/parks*Bug Number: *5A*What were the symptoms? *Bummer Display.*

What component would you replace?

*Found DC4 → Panel cable Reversed  
J1*

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

*Tried TO Boot - No Good.**Plugged console plug**Ran all Test - Display Did NOT clear UP**BUT No Test Reported any error*

① Jp2Diag - memTest  
 FP Test  
 FP Diag  
 FP walkTest

② BpCd2 - OK

③ G3 -

## Lab Bug Report

Student / Team name: *P*

Bug Number: *31*

What were the symptoms?

*want more user*

What component would you replace?

*RC File is Boot wait  
multi node default*

List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.

## Lab Bug Report

**Student / Team name:**

**Bug Number:**

**What were the symptoms?**

**What component would you replace?**

**List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.**

## Lab Bug Report

**Student / Team name:**

**Bug Number:**

**What were the symptoms?**

**What component would you replace?**

**List the steps you took to isolate the problem. List any diagnostic, demo or special procedure you may have used.**

# Contents

<b>Appendix 7.1: Field Engineering Diagnostic Manual</b> .....	<b>1</b>
<b>Section 1: General information and Loading Instructions</b> .....	<b>2</b>
<b>A. General Information</b> .....	<b>2</b>
<b>B. Diagnostics: Component Tested Index</b> .....	<b>2</b>
<b>C. Tests: Component Tested Index</b> .....	<b>3</b>
<b>D. Loading a Test Program</b> .....	<b>3</b>
<b>E. Loading a Diagnostic Program</b> .....	<b>4</b>
<b>Section 2: Running Instructions for Bpcd, Ip2diag and Gf diagnostics</b> .....	<b>6</b>
<b>A. IP2diag</b> .....	<b>8</b>
<b>B. Bpcd diagnostic</b> .....	<b>18</b>
<b>C. GF2 Diagnostic</b> .....	<b>31</b>
<b>Section 3: Test and Diagnostic Man Pages</b> .....	<b>40</b>



## Appendix 7.1: Field Engineering Diagnostic Manual

This appendix is a manual "of sorts", in that, it gives Manual Help Pages for several tests and diagnostics that have been gathered together and released on a cartridge tape to the field.

It is an *un-supported* document and cannot be ordered.

Several of the programs on the tape already exist as part of the basic workstation software release, but several other programs are tests and diagnostics collected from test floor or engineering personnel.

The appendix contains three sections:

- **SECTION 1:**

This section supplies general information and loading instructions for the Tests and Diagnostics.

- **SECTION 2:**

This section contains running instructions for diagnostics *bpcd, ip2diag & gf*.

- **SECTION 3:**

This section contains the manual pages for all the tests and diagnostics.



## Section 1: General information and Loading Instructions

### A. General Information

The programs contained on the tape are grouped into *Tests* and *Diagnostics*.

- **Tests:**  
Tests are programs that run under UNIX. They are simple to run and the manual pages in section III supply you with all the information you will require.
- **Diagnostics:**  
Diagnostics are *stand-alone* programs that are booted from the Prom Monitor. They are much more complex and difficult to run than the Tests. Section II supplies you with the running instructions for *ip2diag*, *gf*, and *bpcd* diagnostics. Lab project 4.3 illustrates how to run the *fex* programs.

The tape contains tests and diagnostics for 2400, 2400T, 2500T, 3020, 3030, and 3130 workstations. The 68020 based program names normally end with a 2.

i.e. *ipfex* for 2500, *ipfex2* for 2500T, 3030 or 3130

### B. Diagnostics: Component Tested Index

*Stand alone*

NAME	TYPE	MODEL	COMPONENT TESTED
*bmtest	diag	2400, 2500	PM2M memory, PM2 processor memory
*bpcd	diag	2400, 2500	Graphic Subsystem
*bpcd2	diag	2400T, 2500T, 3020, 3030	Graphic Subsystem
*dust	diag	2400, 2500	Serial Ports
*gf	diag	2400, 2500	GF2 graphic board
*gf2	diag	2400T, 2500T, 3020, 3030	GF2 graphic board
ipfex	diag	2500	474meg Hard Disk
ipfex2	diag	2500T	474meg Hard Disk
*ip2diag	diag	2400T, 2500T, 3020, 3030	IM1 memory, FP1 fpa board
mdfex	diag	2400	72meg Hard Disk
mdfex2	diag	2400T, 3020	72meg Hard Disk
*memtest	diag	2400, 2500	PM2M memory, PM2 processor memory
sifex2	diag	3030	174meg Hard Disk

\* These programs need an ASCII Terminal connected to port 2.

diag are in /usr/tmp

C. Tests: Component Tested Index

~ under unix

NAME	TYPE	MODEL	COMPONENT TESTED
bpad	test	2400, 2500	Digitizer Pad
bpad2	test	2400T, 2500T, 3020,3030	Digitizer Pad
cprint	test	ALL	Color Printer
cttest	test	ALL	Quarter Inch Cartridge Tape Drive
dev	test	2400, 2500	Mouse and Keyboard Devices
dev2	test	2400T, 2500T, 3020, 3030	Mouse, Keyboard
enet	test	ALL	Ethernet Communications
fptest	test	2400, 2500	Floating Point Board (Skyboard)
fptest2	test	2400T, 2500T, 3020, 3030	Floating Point Board (FP1)
knobs	test	2400, 2500	Dial & Button Box
knobs2	test	2400T, 2500T, 3020, 3030	Dial & Button Box
lpen	test	2400, 2500	Lightpen
lpen2	test	2400T, 2500T, 3020, 3030	Lightpen
mon	test	2400, 2500	Monitor alignment
mon2	test	2400T, 2500T, 3020, 3030	Monitor alignment
mttest	test	ALL	Half Inch Reel Tape Drive
pattern	test	2400, 2500	Monitor check
pattern2	test	2400T, 2500T, 3020, 3030	Monitor check
timeout	test	2400, 2500	Display Blanking
timeout2	test	2400T, 2500T, 3020, 3030	Display Blanking
zclip	test	2400, 2500	GF2 graphics board with z-clipping
zclip2	test	2400T, 2500T, 3020, 3030	GF2 graphics board with z-clipping

D. Loading a Test Program

1. Boot UNIX from disk while in the Prom Monitor.

Type **b** at the *iris>* prompt.

2. Enter multi-user mode, if system is in single-user mode.

Type **multi** at the *unix* prompt.

3. Login in as guest.

Type **guest** at the *Login:* prompt.

4. Change directory to /usr/tmp. (If there is no /usr/tmp; then type **mkdir /usr/tmp**)

Type **cd /usr/tmp**

5. Insert Tests & Diagnostics tape into tape drive.

6. Load test program(s) from tape.

Type **cpio -ivhmud1 <program\_name>** to load a specific program  
or,

type **cpio -ivhmud1** to load all programs from tape.

**NOTE:** To load all the tests and diagnostics, *9 megabytes* of disk space are required.

7. Run selected test program.

Type **<program\_name>** of the program you want to run.

**NOTE:** If you get error: *Segmentation violation -- Core dump*, you are trying to execute a diagnostic program.

If you get error: *Illegal instruction -- Core dump*, you are trying to execute a test for another model.

## E. Loading a Diagnostic Program

1. Boot diagnostic program from tape while in the Prom Monitor.

Type **b mt0:<program\_name>** at the iris prompt. (if 2400)

Type **b ct0:<program\_name>** at the iris prompt. (if 68020 based)

2. Boot diagnostic from disk if they are in */usr/tmp*.

Type **b si0f:/tmp/<program\_name>** (if 2400T and 3020)

**NOTE:** If you get error: *bad magic number* in iris boot header, you are trying to boot a test program.

If you get error: *FATAL ERROR: bus error*, you are trying to boot a diagnostic for another model.

**NOTE:**

**Disk boots from the hard disk will not work if you are on a 2400 while running the diagnostics. You will always boot from the tape.**

## Section 2: Running Instructions for Bpcd, Ip2diag and Gf diagnostics

*Bpcd2, IP2diag & G3*

This section illustrates how to test system boards **IP2**, **IM1**, **FP1**, **BP3**, **DC4**, **UC4**, and **GF2** using the following diagnostics: *ip2diag*, *bpcd* or *gf*.

- **ip2diag**

This diagnostic is used to test the processor boards; IM1 memory board and FP1 floating point point.

- **bpcd<sub>2</sub>**

This diagnostic is used to test the boards of the graphic system; BP3 bit plane, DC4 display controller, UC4 update controller and GF2 geometry engine/frame buffer controller.

- **gf<sub>3</sub>**

This diagnostic is used to test the GF2 geometry engine board. The actual test **gf** is difficult to use, so three tests called *g3*, *g21* and *g2* were created from the commands of **gf** to test the GF2 board. The three tests are comprised of several *macro* commands which are made up of several of the individual commands of the **gf** test.

The test are *go-nogo* tests. If they fail, you must correct the problem by replacing a board, cable or reseating a board or cable.

The instructions show you the output associated with each command and or test executed. The commands are highlighted in **bold** print and the resulting output is highlighted in *italics*.

All the test examples illustrated were run on a 3130 with the following configuration: 8 meg of memory, an FP1 board, and 32 bit planes.

The sequence illustrated should be the sequence that you follow when attempting to isolate a system failure:

1. Processor boards first. (IM1 and FP1)
2. Graphics system (BP3, DC4, UC4)
3. Geometry Engine/Frame buffer Controller. (GF2)

**NOTE:**

**While running the individual tests you may experience system halts (the halt light on "J" box will light) or test loops that do not end. If you encounter one of these conditions, depress reset to terminate the condition and enter PROM monitor. Reboot the test and try again.**

**A. IP2diag**

- The system must have an ASCII terminal installed on port two in order to use this diagnostic.
- While running this diagnostic you will see three (3) different prompts:
  - [n]->  
This is the normal prompt, where n equals the command number.
  - <q>  
This prompt is used by the diagnostic to inform you that additional information is waiting to be displayed if you depress return.
  - ->  
This prompt is used after you have requested help information concerning a specific command. After the diagnostic displays the help file, it places this prompt in front of the command to be executed.

**STEP 1: Booting the diagnostic.**

The diagnostic can be booted from the tape or disk if you had created a */usr/tmp* directory and loaded the tests into the directory.

If your customer is pressed for disk space then the latter method should not be used since the tests require *9 meg* of disk space.

If coming from tape use the following boot command:

- **b ct0:ip2diag**

The example will show the boot from the disk (it was a 3030).

```
iris > b si0f:/tmp/ip2diag
```

```
SGI Extent Filesystem  
Loading: md.0c:/tmp/ip2diag  
Text: 00b284 bytes  
Data: 005240 bytes  
Bss : 0074ac bytes (cleared)  
Jumping to loaded program ~ 20000000.
```

The following will appear at the ASCII terminal on port two (2).

```
pteinit: j=0, *RTCC = fffff0
pteinit. IP2 PCB board
IP2 Diagnostic: Thu Sep 12 15:51:55 PDT 1985
Type <space> to abort the current command
[1]->
```

The normal prompt is displayed and the diagnostic is telling you that if you want to stop (abort) a running command, use the *space bar*.

### STEP 2: Displaying the command menu.

Use **?** to display the available commands.

```
[1]-> ? <return>
```

<i>badbyte</i>	<b>fpdiag</b>	<b>lowtest</b>	<i>readbyte</i>	<i>toutest</i>	<b>?</b>
<i>defmacro</i>	<b>fpwalktest</b>	<i>macro</i>	<i>readword</i>	<i>writebyte</i>	<b>!</b>
<i>errors</i>	<i>getperror</i>	<i>mbmap</i>	<i>readlong</i>	<i>writeword</i>	
<b>fpctest</b>	<b>help</b>	<b>memtest</b>	<i>revision</i>	<i>writelong</i>	
<i>fpreginit</i>	<b>hightest</b>	<i>partest</i>	<i>rtc</i>	<b>\$</b>	
<i>fpooverlap</i>	<i>history</i>	<b>quit</b>	<i>seemacro</i>	<b>.</b>	

The commands that are highlighted in bold print are the ones used by Field Engineers. The other commands are used by test tech's to set up loops and isolating to a failing chip.

The tests that begin with *fp* are floating point tests (FP1 board).

### STEP 3: Displaying help menu.

The following illustrates how to display the help file.



[2]-> help?

help

BACKSPACE	Delete character
DELETE	" "
CTRL-W	Delete argument
TAB	Use default argument
CTRL-L	Redraw command line
CTRL-U	Erase command line
?	Print help test - type at any time
SPACE	Interrupt command execution

<q> <return>

A command-line consists of a command string followed by zero or more arguments. Partially typed command strings are expanded, if possible, when a <space> is typed. Arguments are signed hex numbers separated by a single <space>. Additional spaces between arguments prompt for additional arguments, until none are left. Typing any character other than a space causes all prompts to disappear. Arguments that are not typed are given (compiled-in) default values, which are displayed after the carriage return is typed.

<q> <return>

Please try again. Thanks..

-> help

When you type a command followed by the question mark (?), the help file is displayed. Notice the prompt ( <q> ) is displayed and the diagnostic waits for you to depress return before it displays the remainder of the help message.

Also, after the help file is displayed, the diagnostic displays the command (in the example, it is HELP) and waits for you to enter arguments, depress return to accept the default arguments, or enter control-U ( ^u ) to erase the command.

I will accept and execute the command (help) by depressing return.

-> help <return>

type <help?> for program information.

[3]->

**STEP 4: Running Memtest with default arguments (parameters):**

This test is used to check memory (the IM1 boards).

Before we run it, lets display the help file for this test.

[3]-> memtest?

*memtest <tests> <verbose> <count>*

*Test memory from LOWTEST to HIGHTEST (see those commands). Tests to run are specified as bits in the first argument. Verbose mode reports errors as they occur; otherwise, only the error count total is reported.*

<q> <return>

Tests are:

- 0 All 00000000
- 1 All FFFFFFFF
- 2 Alternate 00000000 and FFFFFFFF
- 3 Alternate 55555555 and AAAAAAAAAA
- 4 Data = Address
- 5 Data = -Address
- 6 Rotate 00000001
- 7 Rotate FFFFFFFE

<q> <return>

-> memtest

After displaying the help file the diagnostic is waiting for you to enter arguments, depress return to accept the defaults or ^u to erase the command and enter another command.

We will accept the default arguments by depressing return.

-> memtest <return>

-> memtest ff 1 1

*Testing memory from 00019000 to 007fffff  
pass 1: 01234567, 0 errors, 0 total.*

[4]->

The test told you where the testing began and ended. Remember, the examples were generated on a workstation with 8 meg of memory, therefore the *007ffff* ending address.

- **Defaults:**

*ff* = select all tests.

*1* = verbose error reporting.

*1* = number of times test will run.

- **Output**

Pass 1: = The pass count of the test.

01234567 = These numbers are output as the individual tests are run.

0 errors, 0 total = The error summary.

If errors were detected during the test, then additional error messages would be output indicating what and where.

#### STEP 5: Running Memtest with supplied arguments.

Let's run only tests 2 & 4, three (3) times using verbose error reporting.

```
[4]-> memtest 14 1 3 <return>
```

```
Testing memory from 00019000 to 007ffff
```

```
pass 1: 24, 0 errors, 0 total
```

```
pass 2: 24, 0 errors, 0 total
```

```
pass 3: 24, 0 errors, 0 total
```

```
[5]->
```

#### NOTE:

Notice that the test begins at location 00019000. The test code resides below this limit. If you wanted to test these locations you would need another memory board with its edge address switches set for the low order addresses in the system, and change the switches on the existing low order board to be the high order board.

Also, notice the test argument value of 14 was used to select tests 2 and 4. The following table illustrates the test bit weight within the test argument.

Test	Bit weight
0	01
1	02
2	04
3	08
4	10
5	20
6	40
7	80

Handwritten notes:   
 9 = Test 0 + 01 + 08 = 0108  
 104 = Test 2 + 4  
 4

**STEP 6: Changing the address test limits of Memtest.**

The following command shows how to change the address test limits.

Remember, the system used for these examples had 8 meg of memory.

I will now change the lower address test location to 400000. Memtest will begin testing at the second meg (400000). The address is changed using the `lowtest` command.

Lets first request help for the command....

[5]-> `lowtest?`

`lowtest <address>`

*Prints the lower memory test bound if no argument is given. A single argument replaces the current lower bound. The low bound of memory (to avoid the test program) is always printed.*

-> `lowtest`

-> `lowtest 200000400000`

`lowtest: 00400000`

`lowmem : 00019000`

Lets again run memtest, selecting only test 3 with verbose error reporting and two (2) passes.

[6]-> memtest 08 1 2

*Testing memory from 00400000 to 007ffff*

*pass 1: 3, 0 errors, 0 total*

*pass 2: 3, 0 errors, 0 total*

[7]->

Remember, these tests are *go-nogo*. If they do not find an error that you are experiencing while UNIX is running, margin the power supply voltages of +5v and -5v about 5% and run tests again.

### STEP 7: Testing the FP1 board.

We will run three floating point tests to test the FP1 board: **fpctest**, **fpwalktest** & **fpdiag**.

These tests are *go-nogo* as far as an F.E. is concerned. We will use the default parameters and replace boards or cables if failures occur. (Boards that could affect test are: IP2, FP1 and IM1)

As in the previous examples, lets first display the help file for the test about to be run...

[7]-> **fpctest?**

*fpctest <verbose> <count> <option reg>*

*Exercises all numerical floating point operations.*

*If any errors occur fpctest displays a summary of errors.*

*If verbose is 1, errors are reported in detail along with the summary.*

*If verbose is 2, a summary of acceptable rounding errors is also displayed.*

-> **fpctest**

We will accept the default arguments for this test...

-> fptest <return>

fptest 1 1 1

ERROR REPORT - 2c (hex) FP OPERATIONS COMPLETED:

opcode	SGL errors	SGLUPD errors	DBL errors	DBLUPD errors
ADD	0	0	0	0
SUB	0	0	0	0
REVSUB	0	0	0	0
MUL	0	0	0	0
DIV	0	0	0	0
REVDIV	0	0	0	0
INT-FLOAT	0	0	0	0
FLOT-INT	0	0	0	0
SGL<->DBL	0	0	0	0
NEGATE	0	0	0	0
TST	0		0	
CMP	0		0	

[8]->

*How To Tell  
When Test Is Passed*

The above error output is normal, indicating a healthy board.

Lets run the next test.... After first displaying the help file.

[8]-> fpwalktest?

fpwalktest <pattern> <targets> <verbose> <count>

Rotates the 32-bit <pattern> and writes it to one or more of the  
1) result register, 2) register file, 3) ALU, 4) multiplier,  
and 5) division hardware. Any 32 bit pattern can be used,  
but 1 and -2 are suggested. Setting any bits of <verbose> causes error  
messages to be printed: setting <verbose> to two makes the fpwalktest  
stop when an error occurs, allowing you to further probe the FPA registers.

<q> <return>

Setting these bits of <target> exercises the following hardware:

- 0 Result register
- 1 Register file
- 2 ALU
- 3 Multiplier
- 4 Divide Hardware

-> fpwalktest

We will accept the defaults by depressing <return>...

-> fpwalktest <return>

-> fpwalktest 1 1f 1 1

Will test X and Y bus for:

result reg

reg file

alu

mul

division

0 errors this pass, 0 errors total.

The last test is fpdiag. Lets again display the help file, then run it accepting the default parameters...

[9]-> fpdiag?

fpdiag <verbose> <count>

->fpdiag

-> fpdiag <return>

-> fpdiag 1 1

The following hardware areas are:

	OK	Worth Checking
Acknowledge	X	
Outer Transceivers	X	
Special Reg Transceivers	X	
Mask Register	X	
Error Register	X	
Option Register	X	
Bus Error Register	X	

<i>Condition Register</i>	X
<i>Inner Transceivers</i>	X
<i>X Register File</i>	X
<i>Y Register File</i>	X
<i>X Result Register</i>	X
<i>Y Result Register</i>	X
<i>X ALU input path</i>	X
<i>Y ALU input path</i>	X
<i>X MUL input path</i>	X
<i>Y MUL input path</i>	X
<i>X BUFFER from ALU/MUL path</i>	X
<i>Y BUFFER from ALU/MUL path</i>	X
<i>DIV SINGLE LOOKUP path</i>	X
<i>DIV DOUBLE LOOKUP path</i>	X

*Fault info  
vector off 1 007  
Exceptional  
Vector #*

[a]->

And that is all there is to testing the FP1 board.

#### STEP 8: Exiting IP2diag.

[a]-> q <return>

really...

really...y

The system now boots into the PROM monitor.



**B. Bpcd diagnostic**

- The system must have an ASCII terminal installed on port 2 in order to use this diagnostic.
- While running this diagnostic you will see three (3) different prompts:
  - [n]->  
This is the normal prompt, where n equals the command number.
  - <q>  
This prompt is used by the diagnostic to inform you that additional information is waiting to be displayed if you depress return.
  - ->  
This prompt is used after you have requested help information concerning a specific command. After the diagnostic displays the help file, it places this prompt in front of the command to be executed.

**STEP 1: Booting the diagnostic.**

The diagnostic can be booted from the tape or disk if you had created a */usr/tmp* directory and loaded the tests into the directory.

If your customer is pressed for disk space then the latter method should not be used since the tests require *9 meg* of disk space.

If coming from tape use the following boot command:

- **b ct0:bpcd2** If 68020 based workstation.
- **b mt0:bpcd** If 2400. (see note below)

The example will show the boot from the disk (it was a 3030).

```
iris > b si0f:/tmp/bpcd2
```

```
SGI Extent Filesystem  
Loading: md.0c:/tmp/bpcd2  
Text: 01840c bytes  
Data: 014ac8 bytes  
Bss: 006a62 bytes (cleared)  
Jumping to loaded program ~ 20000000.
```

The following will appear at the ASCII terminal on port 2.

```
BPC Diagnostic: Wed Mar 12 14:47:44 PST 1986  
UC4  
DC4  
Interface 3 (UC4 Multibus Interface)  
IP2 (If this had been a 2400, this would be PM2)  
Type <space> to abort the current command  
[1]->
```

As you can see, the process is very similar to the IP2diag, in fact, both diagnostics were written by the same person.

**NOTE:**

If you are booting the diagnostic on a 2400 workstation, you **MUST** disconnect the primary monitor keyboard at the "J"- box and depress reset before you enter the boot command. This will force the output to the ASCII terminal. The following output will appear as the diagnostic is loaded.

```
iris > b mt0:bpcd
```

```
Boot:mt.0:bpcd  
OK.  
Loading 0x1832C+0x157B0+0x0 at 0x1000  
Received 187132/4:  
Stack at 0xF6F61C; Starting at 0x1000
```

The remainder of the output at the ASCII terminal is the same as it is for the 68020 based workstations.

**STEP 2: Displaying the command menu:**

Use ? to display the available commands.

[1]-&gt; ? &lt;return&gt;

<i>ball</i>	<i>endmacro</i>	<i>lded</i>	<i>readdda</i>	<i>swap</i>
<b>bpctest</b>	<b>fillrect</b>	<i>ldfmaddr</i>	<i>readfm</i>	<i>time</i>
<b>bplamptest</b>	<i>filltrap</i>	<i>ldmode</i>	<i>readloop</i>	<b>trapctest</b>
<b>bptest</b>	<b>fmttest</b>	<i>ldrepeat</i>	<i>readmap</i>	<i>uclamptest</i>
<b>chartest</b>	<b>gftest</b>	<i>idxe</i>	<i>readword</i>	<i>ucr</i>
<i>clear</i>	<b>help</b>	<i>ldxs</i>	<i>read</i>	<i>wecode</i>
<i>cmdndlist</i>	<i>history</i>	<i>ldye</i>	<b>recttest</b>	<i>we</i>
<i>colorcode</i>	<i>init</i>	<i>idys</i>	<i>request</i>	<i>writfm</i>
<b>colorwetest</b>	<i>joescale</i>	<b>linetest</b>	<i>restore</i>	<i>writemap</i>
<i>color</i>	<i>ldconfig</i>	<i>loopclearword</i>	<i>revision</i>	<i>writeloop</i>
<b>dclamptest</b>	<i>ldddasaf</i>	<i>loop</i>	<i>rgbcolor</i>	<i>writeword</i>
<i>dcpal</i>	<i>ldddasai</i>	<i>macro</i>	<i>rotateloop</i>	<i>write</i>
<b>dcrtest</b>	<i>ldddasdf</i>	<i>mapcolor</i>	<i>save</i>	<b>\$</b>
<i>dcr</i>	<i>ldddasdi</i>	<b>maptest</b>	<i>screenmask</i>	<b>.</b>
<b>ddatest</b>	<i>ldddaeaf</i>	<b>memtest</b>	<b>scrmsktest</b>	<b>?</b>
<i>defmacro</i>	<i>ldddaeai</i>	<b>pixeltest</b>	<i>seemacro</i>	<b>!</b>
<i>depthcue</i>	<i>ldddaedf</i>	<b>quit</b>	<i>sigplanes</i>	
<i>display</i>	<i>ldddaedi</i>	<b>ramptest</b>	<i>steptest</i>	
<i>drawchar</i>	<i>lddec</i>	<i>randlines</i>	<b>stripetest</b>	
<i>drawline</i>	<i>lded</i>	<i>randrect</i>	<i>stripe</i>	

[2]-&gt;

**NOTE:**

Most of the tests are for use by the floor technicians when trying to isolate a component on a board.

The F.E. will use the commands highlighted in **bold** print.

The test **bpctest** will run several of the listed tests in a pre-defined sequence. This test should be the second test run and if it detects failures, then the individual failing test could be run.

If an individual test name is followed with the **?** character, that tests help file is displayed.

i.e. **bpctest?**

**STEP 3: Initializing the raster subsystem.**

The **first** command you must execute upon entering *bpcd* is, **init**.

**init** initializes the raster subsystem (before you begin testing) or returns the raster system to its initialized state (run **init** between tests).

Lets look at the help file....

[2]-> **init?**

*init* <mode>

*Initialize the raster subsystem, or return it to its initialized state.*

*Must be called before any UC command is executed. Arguments are*

<q> <return>

*none* return all values to defaults.

*1* change the default *init* values.

*2* size the bitplanes to determine the default values.

*3* interface *init* only, minimum state change to the raster system.

*4* like 'none', but bitplanes are not altered.

-> *init*

Notice that the test is waiting for you to enter arguments, depress return to accept the defaults, or **ctrl-c** to erase the command.

If this is the beginning of testing (which it is), depress return to accept the default for the *mode* argument. The *init* process will prompt you to enter specific values that will define the configuration of your raster subsystem.

-> *init* <return>

-> *init* 0

configuration [0,ffff], df:

The above parameter can be set to a value from 0-ffff. It's default is **df**. You will **always** accept the default for this argument.

configuration [0,ffff], df: <return>

mode [0,7], 0:

This parameter can be set to a value from 0-7. It's default is 0. You will **always accept the default** for this argument.

mode [0,7], 0: <return>

*dc register [0,ffff], 0:*

This parameter can be set to a value from 0-ffff. It's default is 0. This parameter will define the type of monitor you are using as your primary or secondary monitor. See the table below for the correct code that will define your monitor:

- If you are using the **STANDARD CONFIGURATION:**

Primary monitor is 60 Hz (NI),      enter: 3000

Secondary monitor is 30 Hz (I),      enter: 0

- If you are using an **OPTIONAL CONFIGURATION:** (configured in pairs)

Primary monitor is 60 Hz (NI),      enter: 1000

Secondary monitor is RS170A,      enter: 2800

Primary monitor is 30 Hz (I),      enter: 1000

Secondary monitor is RS170A,      enter: 2800

Primary monitor is 60 Hz (NI),      enter: 1000

Secondary monitor is EURO,      enter: 2800

Primary monitor is 30 Hz (I),      enter: 1000

Secondary monitor is EURO,      enter: 2800

The system I was on when this document was created was set for the Standard Configuration and the Primary Monitor selected, therefore, I will enter 3000 to the *dc register* prompt.

dc register [0,ffff], 0: 3000 <return>

*color index [0,fff], 0:*

This prompt is asking you what color the test should use as it's index (reference) color. There are 7 colors you can select, see the following table:

Value	Color
0	Red (default color)
1	Green
2	Blue
3	White
4	Black

The color is not important, so you can select any color, I accepted the default by depressing return...

color index [0,fff], 0: <return>

write enable code [0,ffffff], fffffff:

The value for this parameter is between 0-ffffff. This is a mask for the bit planes to which you are going to allow diagnostic writing. There are 32 bits in the mask, each significant to one of the possible 32 bit planes. If the write enable (we) bit is set ( 1 ), then the corresponding bit plane can be written. To select which planes you want to write into, see the following table:

*address  
switch*

*These are Decimal  
But report Bad Plane  
in Hex.*

SLOT NUMBER	BIT PLANE NUMBER																															
	3	3	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	0	9	8	7	6	5	4	3	2	1	0		
17																																
16																																
15																																
14																																
13																																
12																																
11																																
10																																

*MSB*

*LSB*

i.e. If you only wanted to write into the bit planes of the board in slot 17, enter 303 to the *write enable* prompt.

Test slots 16 and 17, enter f0f.

Test slots 15,16 and 17, enter 3f3f.

I have all 32 bit planes installed and I want to test them, so I will accept the default by pressing <return>...

write enable code [0,ffffff], fffffff: <return>

*significant bitplanes* [0,ffffff], fffffff:

This prompt is telling you what bit planes the diagnostic was able to communicate with (what planes are on line). The example implies that all 32 bit planes are present and the diagnostic can see them. This is what I would expect since 32 bit planes are installed.

**NOTE:**

If you know how many planes are present in your system and this parameter indicates fewer than expected, then the diagnostic may have already found a problem for you.

With this parameter you can select what planes you want tested. The value entered is the same as the write enable code (we). If this is your first pass, it is best to test all the installed planes, so depress return..

significant bitplanes [0,ffffff], fffffff: <return>

*cfr: display ab, update ab, viewport, backline, ldlinestp*

*mdr:*

*deflags: pipe 4, prom, vsm-clear*

*significant bitplanes: fffffff*

*installed bitplanes: fffffff*

*color index: 0*

*write enable code: fffffff*

[3]->

Your screen will now be initialized to the color you selected with the *color index* prompt. If you had selected 0 (the default), your screen should be *Red*.

You are now ready to run a test. Remember, you must execute *init* between tests to return all

the parameters to the selected or default values. If you wish to enter new parameter values, then you must execute `init 2` to force the diagnostic to prompt you again for the individual parameters.

**STEP 4: Testing the complete graphics subsystem.**

As mentioned earlier, the best approach for isolating errors in the graphics subsystem is running the test `bpctest` after executing the `init` command.

This test runs several of the individual tests against all the boards in the graphics subsystem. If errors are detected, then the individual failing test(s) could be run to verify the failures and narrow the possibilities.

Lets look at the help file for `bpctest`...

[3]-> `bpctest?`

`bpctest <iterations> <verbose> <tests>`

7654 321 0 -DC4

Run an assortment of tests on the entire raster system. All tests are run non-verbose. Each test reports its error total if verbose is true (not 0). Otherwise only the error total of all of the tests is printed. Unlike other test commands, bits in the 'tests' argument specify the boards to be tested (rather than specific tests to run).

<q> <return>

bit meaning

--- -----

- 0 Run tests that are applicable to display controller.
- 1 Run tests that are applicable to update controller.
- 2 Run tests that are applicable to the bitplane boards.
- 3 Run `bpctest` (display on and off). This test is very long with UC3.
- 4 Run the processor memory test (PM2 and IP2 only).
- 5 Run the `gf2` test (GL2 systems only - others will crash).
- 6 Run the floating point test. IP2 with FP1 only.

-> `bpctest`

**NOTES:**

- Running `bpctest` using only bit 3 (<tests> argument = 08) does not work due to a programming error.



- The default for <tests> is 3f. This runs all the tests except the floating point test because not all systems are shipped with an FP1 board.
- If the test is running on a 2400, only six choices are available; the FP1 is not available on 2400 systems.
- The test *memtest* is the same memtest that is part of the IP2diag.

lets now run **bpctest** accepting the defaults...

-> bpctest <return>

-> bpctest 1 1 3f

Pass 1: 01234567012345670123456701234567 global errors: 0

read errors: 0

write errors: 0

weird errors: 0

(round errors: 0)

memtest complete, 0 errors

gftest complete, 0 errors

bptest complete, display ab, 0 errors

bptest complete, no display, 0 errors

stripetest complete, display ab, 0 errors

stripetest complete, display ab, sbuf, 0 errors

color/we test complete, 0 errors

fmtest complete, 0 errors

dcrtest complete, 0 errors

maptest complete, 0 errors

linetest complete, 0 errors

viewporttest complete, 0 errors

(scrmsktest)

rectangle test complete, 0 errors

(recttest)

character test complete, 0 errors

(chartest)

dda test complete, 0 errors

pixel test complete, 0 errors

trapezoid test complete, 0 errors

(traptest)

errors this pass = 0, total errors = 0

[4]->

#### NOTE:

Notice that four of the test names in the output are different from those listed in the commands menu. The actual names are highlighted in bold print next to the used name.

If one or more of the tests reports errors, you could run those tests individually.

**STEP 5: Running individual Bpcd tests.**

The next few examples illustrate running tests against the individual boards in the graphics subsystem. Remember, the individual boards can be tested by changing the default value of the `<tests>` argument of `bpctest`.

Because the boards of the graphics subsystem are so tightly coupled, you have to keep in mind that when testing the individual boards, other boards in the subsystem can affect the board function that you are testing. In each of the following examples, you are shown what board is being tested and what boards you should suspect if the individual test fails. (Test output is not shown for the following examples)

The first example illustrates how to test the DC4 board.

```
[4]-> init <return>
```

```
[5]-> bpctest 1 1 1
```

This configuration runs tests *stripetest*, *color/we test*, *fmttest*, *dcrtest*, and *maptest*.

- If *dcrtest* and/or *maptest* detect failures, DC4 is defective.
- If *stripetest*, *color/we test*, or *fmttest* detect failures, then cables, DC4, UC4, or a BP3 could be defective.

The next configuration runs tests *stripetest*, *color/we test*, *fmttest*, *linetest*, *viewporttest*, *rectangle test*, *character test*, *dda test*, *pixel test*, and *trapezoid test*. It checks the Update Controller (UC4).

```
[6]-> init <return>
```

```
[7]-> bpctest 1 1 2
```

- If *linetest*, *viewporttest*, *rectangle test*, *character test*, *dda test*, *pixel test*, or *trapezoid test* detect failures, then UC4 is the most probable cause.
- If *stripetest*, *color/we test*, or *fmttest* detect failures, then cables, DC4, UC4, or a BP3 could be defective.

The next configuration runs tests *stripetest*, *color/we test*, and *pixel test*. It is a basic check of the bit plane boards.

[8]-> **init <return>**

[9]-> **bpctest 1 1 4**

- If **pixel test** detects failures, then a bad bit plane is probably the cause.
- If **stripetest** and/or **color/we test** fail, then cables, DC4, UC4, or a BP3 board could be the problem.

The configuration (**bpctest 1 1 8**) is invalid due to a programming bug. The test *bpctest*, which this should run, will run when the default 3f is accepted for **bpctest**, but will not run when you try to select it with bit three of the tests mask.

The following three examples show you how to exercise the IM1 memory boards, GF2 and FP1 boards using *bpctest*.

- |                   |                   |
|-------------------|-------------------|
| 1. bpctest 1 1 10 | Test memory (IM1) |
| 2. bpctest 1 1 20 | Test GF2 board    |
| 3. bpctest 1 1 40 | Test FP1 board    |

Two tests that can be run individually and are of interest, are *stripetest* and *bpctest*. Both tests are basic tests of the bit planes, but they test the planes differently.

- **Stripetest** uses a specific pattern that is displayed on the monitor. The DC4 board and cables are in the loop and should be suspect if failures occur.
- **Bpctest** is really a memory test, remember, the bit planes are nothing more than RAM memory devices. The P2 bus is used to write and read the data into and out of the planes. Video appears at the monitor for this test, but it has no real pattern and it would be hard to spot a display problem in the pattern.

Lets run *stripetest*...

[a]-> **init** <return>

[b]-> **stripetest** <return>

[b]-> **stripetest 1 1 40 0**

pass 1: cannot test planes f0f00000, C:4567, D:4567

a0a1a2a3a4a5a6a7b0b1b2b3b4b5b6b7c0c1c2c3d0d1d2d3 badplanes: 0, cumulative: 0

[c]->

ANY # means ...  
 90(0123) → Error  
 91 - Top cable  
 23 - Bottom cable

The test is telling you it cannot test the bit planes in slots 10 and 11. This is because there are no cables on these boards, remember, they are used for Z-buffering.

Errors are reported as characters in parentheses after the plane name.

The next example illustrates running *bptest*...

[c]-> **init** <return>

[d]-> **bptest** <return>

Errors reported in Hex

[d]-> **bptest 1 1 3f 0 0**

pass 1: 012345, errors this pass = 0, total errors = 0

[e]->

While this test is running, you will notice dots (...) appearing between the test numbers as the individual tests of *bptest* are run.

Lets look at the help file for *bptest*...

[e]-> **bptest?**

[e]-> **bptest <iterations> <verbose> <tests> <chip report> <diag>**

Test the operation of the bitplane memory. Errors are reported as they occur if verbose is true (not 0). Otherwise error total is reported. Individual tests are selected with bits of the test argument.

<q> <return>

BIT	TEST
0	All 0 test.
1	All 1 test.
2	Word = address.
3	Word = compliment of address. UC4 only.
4	Word takes random value.
5	Exercise saveword/drawword hardware.

-> *bptest*

As you can see, *bptest* is nothing more than a memory test.

This ends the examples of *Bpcd*. The following illustrates how to get back to the PROM monitor.

-> *bptest ^u* (This clears the command line)

-> *q*

*type 'y' to confirm y*

**NOTE:**

Sometimes the system halts when *q* is typed, if this occurs, just depress reset to get back to the PROM monitor.

### C. GF2 Diagnostic

- The system must have an ASCII terminal installed on port two to use this diagnostic.
- While running this diagnostic you will see the following prompt, !. It is the exclamation point.
- Three diagnostics, *g3*, *g21*, and *g2* will be used to test the GF2 board. These tests are comprised of macro commands built using individual commands from the parent diagnostic *gf2*. The macro commands are much easier to use than attempting to perform the same function using *gf2*.
  - *g3* is used on 68020 based workstations.
  - *g21* and *g2* are used on 2400 workstations. These two tests perform the same testing as *g3*.

**NOTE:**

Before you boot either *g21* or *g2*, you must disconnect the primary monitor's keyboard at the J-box and depress reset. This will force the output to the ASCII terminal on port two.

#### STEP 1: Booting the diagnostic.

By now you should have a good idea what type of data is displayed when you boot a diagnostic, so only the initial display of the diagnostic will be illustrated. Also illustrated are the different options you have to boot the diagnostics.

System	Media	Command	Comment
2400	Tape	b mt0:g21	Loads g21 from tape.
2400	Tape	b mt0:g2	Loads g2 from tape.
68020	Tape	b ct0:g3	Loads g3 from tape.
3020	Disk	b md0c:/tmp/g3	Loads g3 from disk /usr/tmp.
3030	Disk	b si0f:/tmp/g3	Loads g3 from disk /usr/tmp

The system that was used to create the output for this document was a 2400 Turbo with the following configuration: 8 meg of memory, 32 bit planes, an FP1 board, and all 14 macro chips on the GF2 board (Z-buffering was installed).

After the boot command was executed, the following appeared at the ASCII terminal when g3 was booted.

**GF2/IP2/UC4/DC4 CONSOLE V1.9**

*changes:*

*'t' no longer does 'bt'*

*EPROM version 1.4 kluge: > emb 3500000b change to 0f  
use ^C to escape macro loops!*

!

If you are on a 2400, g21 must be booted first to allow you access to the same commands that this document illustrates. The following would appear after the boot command is executed:

**GF2/PM2/UC4/DC4 CONSOLE V1.2**

*changes:*

*use gt0-7 then gt9 Gv*

!

In both of the initial messages you see the word *changes* followed by some information about command changes; it does not concern the F.E. What is important, is that the prompt ( ! ) is displayed.

Once you have g3 (68020 based) or g21 (2400) loaded and the ( ! ) is displayed, the flow is the same.

The tests are of the *go-nogo* type. If any of the following tests fail, correct the problem by replacing the GF2 board, cables, or reseating the board and/or it's cables.

Remember, other boards in the system could cause these tests to fail. Before replacing GF2 you should run diagnostics against other boards in the system.

All the output of the examples show normal status (0 errors). If you experience errors, your output will be different.

**NOTE:**

Understanding how the GF2 board functions is not important, following the recommended order of initializing and testing the board is!

**STEP 2: Initializing the GF2 board.**

The first two commands (**w** and **SI**) initialize the GF2 board and must be executed before any testing occurs.

```
! w <return>
```

```
OK
all bkpts cleared.
!
```

```
! SI <return> (upper case S and I)
```

```
findge head GA ..... tail GA found: 14 mask: 3fff
scratchsize: 0fff version: 2.3
!
```

The program (**SI**) is informing you that it sees all 14 of the GF2 macro chips: the two Geometry Accelerators (head GA and tail GA) and 12 Geometry Engines (.....).

Two of the Geometry Engines are optional and your system may not have them; they are GE5 and GE10 and perform near and far clipping (Z-clipping).

**STEP 3: Testing the Geometry Engines.**

The next few commands test the Geometry Engines with various data patterns.

```
! cp <return>
```

```
0 errors in 482 words.
!
```



**! cP <return>**

*0 errors in 196 words.*

|

**! ct <return>**

*0 errors in 993 words.*

*0 errors in 1047 words.*

*0 errors in 1101 words.*

*0 errors in 721 words.*

*0 errors in 728 words.*

**5 not installed**

*0 errors in 735 words.*

*0 errors in 751 words.*

*0 errors in 767 words.*

*0 errors in 755 words.*

*0 errors in 780 words.*

**10 not installed**

*0 errors in 700 words.*

*0 errors in 482 words.*

The above output is what you would see for a GF2 with all 12 Geometry Engines. If your system only has 10, then the statements that are highlighted in bold print would be part of the output.

**! cT <return>**

*0 errors in 1063 words.*

*0 errors in 1117 words.*

*0 errors in 1171 words.*

*0 errors in 791 words.*

*0 errors in 798 words.*

**5 not installed**

*0 errors in 805 words.*

*0 errors in 821 words.*

*0 errors in 837 words.*

*0 errors in 825 words.*

*0 errors in 850 words.*

**10 not installed**

*0 errors in 770 words.*

*0 errors in 552 words.*

**STEP 4: Testing the Geometry Accelerators.**

The next two commands will test the Geometry Accelerators using various data patterns.

**! cg <return>**

*0 errors in 588 words.*  
*0 errors in 262 words.*  
*0 errors in 262 words.*  
*0 errors in 262 words.*  
*0 errors in 262 words.*  
*0 errors in 262 words.*  
*0 errors in 262 words.*  
*0 errors in 262 words.*  
*0 errors in 262 words.*  
*0 errors in 456 words.*  
*0 errors in 428 words.*  
*0 errors in 173 words.*  
*0 errors in 278 words.*  
*0 errors in 173 words.*  
*0 errors in 133 words.*

**! cG <return>**

*0 errors in 658 words.*  
*0 errors in 332 words.*  
*0 errors in 332 words.*  
*0 errors in 332 words.*  
*0 errors in 332 words.*  
*0 errors in 332 words.*  
*0 errors in 332 words.*  
*0 errors in 332 words.*  
*0 errors in 332 words.*  
*0 errors in 526 words.*  
*0 errors in 498 words.*  
*0 errors in 243 words.*  
*0 errors in 348 words.*  
*0 errors in 243 words.*  
*0 errors in 203 words.*

The previous tests checked the GF2 hardware without any data being output to the monitor. If failures occurred during one of them, chances are good that the problem is in the GF2 board.

The following tests display graphics at the monitor, so failures that occur could be other components in the system.

#### **STEP 5: Booting *g2* if a 2400.**

##### **NOTE:**

If you were on a 68020 based workstation, you could continue with the testing flow (jump to STEP 6), but if you were on a 2400, then you must boot the graphics test *g2* before continuing.

Execute the *quit (q)* command to return to the PROM monitor so that you can boot *g2*.

```
! q <return>
```

```
iris> b mt0:g2 <return>
```

Again the boot message is not shown, but the initial diagnostic output is:

```
GF2/PM2/UC4/DC4 CONSOLE V1.7
```

```
changes:
```

```
use ^c to escape macro loops!
```

```
gm tests: gm1 draws 2 red squares
```

```
!
```

#### **STEP 6: Graphics Testing.**

Depending on the type of monitor you have selected, from this point on, there are minor differences. These differences will be highlighted as the flow continues.

Lets again initialize the GF2 board.

**! w <return>**

OK  
all bkpts cleared.  
!

**! SI <return>**

*findge head GA ..... tail GA found: 14 mask: 3fff*  
*scratchsize: 0fff version: 2.3*  
!

The next command is used only if your monitor is a 60 Hz monitor. You only need to execute it once. If your monitor is a 30 Hz, omit this command.

**! m <return>**

1-> 3020  
!

The next command runs several tests and then draws two red squares connected by a green, red, blue and gray line.

**! t <return>**

*Multibus: micro ram: micro test: done*  
OK  
*init: scratchsize: 0fff version: 2.3*  
*< testing 4096 words*  
*<<< scratch test done.*  
*< testing 4096 words*  
*<<< scratch test done.*  
*< testing 4096 words*  
*<<< scratch test done.*  
*< testing 4096 words*  
*<<< scratch test done.*  
*< testing 4096 words*  
*<<< scratch test done.*  
*draw: scratchsize: 0fff version: 2.3*  
*findge head GA ..... tail GA found: 14 mask: 3fff*

*byteswap: token test: OK*

*fifo test: 153 wds*

*GE test:*

*Tests done. (485 vector words)*

!

The next command draws blinking colored lines (solid and dashed) within 4 white boxes. The two boxes on the left side of the screen are smaller than the two boxes on the right side. When the test is complete, the boxes are completely white.

! cd <return>

*findge head GA ..... tail GA found: 14 mask: 3fff*  
*scratchsize: 0fff version: 2.3*

The next series of commands will initialize the GF2 board, clear the screen and display **Red and White** squares that spiral in on themselves.

! ii <return>

*findge head GA ..... tail GA found: 14 mask: 3fff*  
*scratchsize: 0fff version: 2.3*

! gC <return>

! gt <return>

! Gbd <return>

The last test displays a box in the lower left corner of the screen that has colored bars moving through it from bottom to top. When the test is complete, the box is left a solid color.

! ii <return>

*findge head GA ..... tail GA found: 14 mask: 3fff*  
*scratchsize: 0fff version: 2.3*

**! gC <return>**

**! gt <return>**

**! Gd <return>**

This completes the GF2 testing. To return to the PROM monitor, execute the quit command.

**! q <return>**

### **Section 3: Test and Diagnostic Man Pages**

- This section contains manual help pages for all the tests and diagnostics contained on the diagnostic tape.
- They are generally useful for running tests (UNIX), but not so when running diagnostics (stand-alone).
- They were added to the end of this appendix, therefore they do not have page numbers, but they are in alphabetical order.





### **Section 3: Test and Diagnostic Man Pages**

- This section contains manual help pages for all the tests and diagnostics contained on the diagnostic tape.
- They are generally useful for running tests (UNIX), but not so when running diagnostics (stand-alone).
- They were added to the end of this appendix, therefore they do not have page numbers, but they are in alphabetical order.

**DIAG**

**bmtest** - Exercise PM2M memory boards for proper operation.

**SYNOPSIS**

**bmtest**

**DESCRIPTION**

The *bmtest* program exercises PM2M memory board for proper operation.

Press **h** to display the following command listing:

- Clear all patterns/tests
- cross test direct/multibus (True)
- direct memory tests (True)
- Error list
- enable Parity (False)
- fullmem test (True)
- help
- low mem to test
- Memory select
- multibus tests (True)
- pattern select
- quit
- Refresh rate
- run
- summary each pass (True)
- verbose (True)

Type **e** and then **t** to enable parity testing.

Type **r** to run the program.

Press **RETURN** to abort the current test.

**BMTEST** is a memory exerciser for both the PM2M expansion memory, and the PM2 on-board memory. The program verifies proper memory operation and data under various patterns. The program also tests proper operation of the multibus map, memory access via program will detect parity failures and report them, but it will not identify exactly where the parity error(s) occurred.

**BMTEST**

**- continued -**

**BMTEST**

**BMTEST** tries to identify failures to the memory chip which is at fault. If there are grounded or shorted memory data in/out lines, this will probably lead to whole rows of chips reported as bad. The correlation between failures and chip position is theoretical and should be treated as a clue only.

Press the **q** key to quit the program.

**TEST**

*bpad* - test the digitizer pad

**SYNOPSIS**

*bpad*

**DESCRIPTION**

The *bpad* program tests the digitizer pad.

A viewport screen should appear on the screen in the color grey. Within the grey screen there should be a diagonal image of a pad. You should move the image of the pad around on the digipad and press the four buttons to make sure the pad-peripheral works.

After testing the pad-peripheral then turn the power off and insert the pen-peripheral then turn power back on. The same pad like image should appear on the screen. Move the pen-peripheral around and then press the pen on the digi-pad; this should display a dark image within the pad image.

**HARDWARE REQUIREMENTS**

digitizer pad

**DIAG**

bpcd - diagnose the graphic subsystem

**SYNOPSIS**

bpcd

**DESCRIPTION**

The *bpcd* program is used to diagnose the graphic subsystem.

Generally, the display controller performs two distinguishable functions. The first of which is memory and monitor timing signal generation; The second is display generation with reference to the color maps, rgb data paths, DAC's etc.. The objective of this document is to provide an efficient, yet complete, test procedure for the DC4.

Enter ? to display the following command listing:

ball	endmacro	ldfmaddr	readda	swap
bpctest	fillrect	ldmode	readfm	time
bplamptest	filltrap	ldrepeat	readloop	traptest
bptest	fmttest	ldxe	readmap	uclamptest
chartest	gftest	ldxs	readword	ucr
clear	help	ldye	read	wecode
cmdnlist	history	ldys	recttest	we
colorcode	init	linetest	request	writefm
colorwetest	joescale	loopclearword	restore	writemap
color	ldconfig	loop	revision	writeloop
dclamptest	ldddasaf	macro	rgbcolor	writeword
dcpal	ldddasai	mapcolor	rotateloop	write
dcrtest	ldddasdf	maptest	save	\$
dcr	ldddasdi	memtest	screenmask	.
ddatest	ldddaeaf	pixeltest	scrmsktest	?
defmacro	ldddaeai	print	seemacro	!
depthcue	ldddaedf	quit	sigplanes	
display	ldddaedi	rampptest	steptest	
drawchar	ldec	randlines	stripetest	
drawline	lded	randrect	stripe	

**Important Note**

You must first type init to initialize program.

**bptest** <iterations> <verbose> <tests> <chip report> <diag>

Test the operation of the bitplane memory. Errors are reported as they occur if verbose is true (not 0). Otherwise only an error total is reported. Individual tests are selected with bis of the 'tests' argument.

BIT	TEST
0	All 0 test.
1	All 1 test.
2	Word = address.
3	Word = complement of address. UC4 only.
4	Word takes random value.
5	Exercise saveword/drawword hardware.

**clear** <x stipple> <y stipple>

Clear the 1024x1024x bitplane memory to current colorcode. A 16x16 (8x16 for uc3) pattern that is the bitwise OR of the horizontal and vertical patterns is used. Wecode is not changed.

**colorcode** <code>

Set the (up to) 32-bit colorcode. Current color index becomes invalid. Code bits are always ordered D7-0, C7-0, B7-0, A7-0, regardless of the current swizzle or double values. Cfr and mdr are restored after the code has been written.

**color** <index>

The given color index is given into a code, taking into account the current cfr, dcr, and sigplane values. This code becomes the current colorcode, and is written into the bitplanes. Nothing is changed if no argument is given. The current (new) value is always printed.

**dclamptest** <iterations>

Individually turn on each of the display control LED's. DC4 'optclk' stays on longer for some obscure reason.

**dcpal** <value>

Set the VFSM (video finite state machine) to the specified value (range 0-7) (no change if no argument is given). Then sample the VFSM outputs during the following vertical retrace periods and print the results.

**dcrtest** <iterations> <verbose>

Test the display controller multibus register.

**dcr** <dcr>

Set the display control register to a new value. The contents are not changed if no argument is given. The current (or new) value is printed.

BIT	DC4	BIT	DC4	BIT	DC4	BIT	DC4
0	reg0	4	bus	8	pal2	c	pipe4
1	reg1	5	reg	9	optclk	d	prom
2	reg2	6	rgb	a	pipe4	e	lk
3	reg3	7	highmap	b	prom	f	mbit

**display** <x word address> <y address>

The color indexes of s word-aligned 64x20 screen area are printed. Each index is masked by 0xf so that it can be printed as a single character. The special character ':' is printed if and only if ALL bits of that pixel are 0.

**drawline** <x1> <x2> <y1> <y2> <stipple>

Draws a line from x1,y1 to x2,y2 with the specified stipple.

**endmacro**

endmacro

**fillrect** <x> <y> <width> <height> <x stipple> <y stipple>

Fills the specified rectangle with the current color. A 16x16 pattern is specified as the bitwise OR of the given horizontal and vertical patterns.

**help**

BACKSPACE	Delete character
DELETE	" "
CTRL-W	Delete argument
TAB	Use default argument
CTRL-L	Redraw command line
CTRL-U	Erase command line
?	Print help text - type at any time
SPACE	Interrupt command execution

A command-line consists of a command string followed by zero or more arguments. Partially typed command strings are expanded, if possible, when a <space> is typed. Arguments are signed hex numbers separated by a single <space>. Additional spaces typed between arguments prompt for additional arguments, until none are left. Typing any character other than a space causes all prompts to disappear. Arguments that are not typed are given (compiled-in) default values, which are displayed after the carriage return is typed. BPCD supports nested (not recursive) user-defined macros, and includes both history and argument substitution mechanisms.

**history <command number>**

A history of previous commands is maintained. Typing 'history' with no argument causes the list of recently typed commands to be printed. Each command in this list is preceded by its number. A remembered command can be re-executed by typing 'history <number>'. Alternately, the n'th previous command can be re-executed by typing 'history -n'.

**init <mode>**

Initialize the raster subsystem, or return it to its initialized state. Must be called before any UC command is executed.

Arguments are:

- none return all values to defaults.
- 1 change the default init values
- 2 size the bitplanes to determine the default values
- 3 interface init only, minimum state change to the raster system
- 4 like 'none', but bitplanes are not altered

**joescale**

Draws a pattern useful for monitor adjustment.

**ldconfig <value>**

Set the uc configuration register to a new value. The current value is not changed if no argument is given. The current (or new) value is printed.

BIT	UC3	UC4
0	display a	display a
1	display b	display b
2	update a	update a
3	update b	update b
4	screenmask	screenmask
5	enable cd	invert
6	finish line	finish line
7	ldlinestip	ldlinestip
8		pfi cd
9		pfi read
a		pfi column
b		pfi x down
c		pfi y down
d		all pattern
e		pattern 32
f		pattern 64



**loop** <command> <data>

Continuous requests until interrupted. Read-requests do read, but they do not print a value. See 'request' for a list of command assignments.

**macro** <number> <iterations>

Execute a user-defined macro a specified number of times.

**mapcolor** <index> <red> <green> <blue>

All colormap entries that correspond to index are changed to the new red, green, and blue values. An index corresponds to multiple map entries if not all of the planes connected to the colormap are Index is in the range  $0..(2^{**n})-1$ , where n is the number of significant planes that are currently connected to the colormap. Red, green, and blue are 8-bit intensities, ranging from 0 (off) to FF (brightest). All arguments are given in hex. For example, if only bitplane A0 is significant (sigplanes = 1) then the command 'mapcolor 0 ff ff ff' will set every other colormap entry (0, 2, 4, etc) to 'ff ff ff'. Calling mapcolor with no significant planes sets the entire colormap.

**maptest** <iterations> <verbose> <tests> <diag>

Test the operation of the colormap. Errors are reported as they occur if verbose is true (not 0). Otherwise only an error total is printed. Individual tests are selected with bits in the 'tests' argument.

BIT	TEST
-----	------

0	Red, green, blue = 0.
---	-----------------------

1	Red, green, blue = FF.
---	------------------------

2	Red, green, blue = address & FF.
---	----------------------------------

3	Red, green, blue = address & F00.
---	-----------------------------------

4	Red, green, and blue = different random values.
---	---

**ramptest** <red> <green> <blue> <rgb>

Fill the screen with a ramp of intensity increasing from left to right. Each of red, green, and blue is either ramped or not depending on whether the corresponding argument is false (0) or true (not 0). RGBmode disables the colormap; it requires 24 bitplanes for correct operation.

**randlines** <seed> <number> <stipple>

Draws the specified number of lines with the specified stipple pattern in the current color. 'Seed' initializes the random number generator determines the line endpoints. Any value will do. Be sure to clear the screen to a color that is different from the current color, or you won't see anything.

**randrect** <log2 size>

Continuous random fill of the screen with rectangles. Color indexes 0, 1, 2, and 3 are used with 4 different patterns. Rectangle dimensions range from 1 through 2 raised to the power of the passed argument. For example,

**readloop** <i/o address>

Read continuously any multibus i/o space location, and do NOT print the 16-bit results. The i/o space is a 16-bit space. Address bits 0 and 16..31 are ignored. Word access only (no byte reads).

**readmap** <address> <locations>

Print consecutive colormap values. The 'address' argument is an absolute colormap address in the range 0..FFF. It is NOT a color index.

**read** <i/o address>

Read any multibus i/o space location, and print the 16-bit result. The i/o space is a 16-bit space. Address bits 0 and 16..31 are ignored. Word access only (no byte reads).

**rgbcolor** <red> <green> <blue>

Sets the colorcode to the given red, green, and blue values (range 0..ff). Color index becomes invalid.

**sigplanes** <planes>

Specify which bitplanes are significant, in terms of index-to-code conversion and colormap changes. Sigplanes is not changed if no argument is given. The current (new) value is always printed.

**steadtest** <red> <green> <blue> <rgb>

Like ramptest, but draws 10 vertical stipes of increasing intensity, instead of 256.

**stripetest** <iterations> <verbose> <fields> <singlebuf>

Fills each significant plane with a 3-off/3-on pattern, then uses the VFSM to determine whether this pattern is being correctly received at the output of the colormap. The colormap is adjusted to output only a function of the bitplane being tested. Errors are reported as characters in parentheses after the plane name. DC3 characters are 'e' and 'o', indicating errors in even and/or odd planes. DC4 errors are '0', '1', '2', and '3' to allow pipe4 error reporting.

**stripe** <width> <step>

Fill the screen with screen-height rectangles of the specified width, on the specified horizontal pitch. For example, 'stripe 3 6' draws rectangles that are 3 pixels wide, with 3-pixel spaces (pitch of  $6=3+3$ ).

**writemap** <address> <red> <green> <blue>

Change a single colormap location. The 'address' argument is an absolute colormap address in the range 0..FFF. It is NOT a color index.

**writeloop** <i/o address> <data>

Write continuously to any multibus i/o space address. Interrupt to stop. The i/o space is a 16-bit space. Address bits 0 and 16..31 are ignored. Word access only (no byte writes).

**writeword** <x word address> <y address> <data word>

All planes of the specified bitplane word are loaded with value. This is accomplished by first setting all 16 pixels to all 0's, then setting the appropriate pixels to 1's. After the write is completed, a UC SAVEWORD command is executed so that the bitplane registers are also set to the specified value (for probe/scope purposes).

**write** <i/o address> <data>

Write to any multibus i/o space address. The i/o space is a 16-bit space. Address bits 0 and 16..31 are ignored. Word access only (no byte writes).

### SAMPLE RUN

The sequence of tests are shown below. Each step must follow consecutively as listed. If an error occurs, or if a retest is desired at a specific point in this list, restart the command sequence at the PREVIOUS 'init 2' command statement.

### INTERLACED MONITOR, SINGLE BUFFER MODE, MAPPED

```

init 2           ; The DC is mapped into the multibus address space.
dclamptest 1    ; The lamp test checks for address acknowledge and
dcr 0           ; gives visual feedback of correct DCR operation.
dcrtest 1 1     ; If bus errors result, use a logic probe and the
maptest 1 1 1f 0 ; DCR command to trace the acknowledge path.
fmtest 1 1 1f 0 ; DCRTEST checks for stuck and shorted bits in DCR.
ldconfig 0f     ; Performs a rudimentary test on the video fsm PAL.
bptest 1 1 3f 0 0 ; MAPTEST looks for both data and address errors in
ldconfig C     ; the color maps. READMAP and WRITEMAP can be used to
bptest 1 1 3f 0 0 ; interrogate and alter the colormap contents.
init 2         ;
dcr 0         ;
ldconfig 0f   ; including refresh.
stripetest 1 1 40 1 ; STRIPETEST requires 24 bitplanes for correct operation.

```

**INTERLACED MONITOR, DOUBLE BUFFER MODE, MAPPED**

ldconfig 9

stripetest 1 1 40 0 ;STRIPETEST double buffered and single buffered  
;checks both data paths for each mode.

**INTERLACED MONITOR, SINGLE BUFFER MODE, MAPPED**

ldconfig 0f

ramptest 1 0 0 0 ; Mapped mode, Use an oscilloscope to individually  
; check each DAC output. All 256 levels should blend  
; into a smooth ramp. The ramp amplitude (with only  
; the on-board 75 ohm resistor as termination) should  
; be 1.5 volts peak-to-peak. As shown, RED is active,  
; the DAC for the RED output can be checked. Requires  
; 24 bitplanes for correct operation.

ramptest 0 1 0 0

ramptest 0 0 1 0 ; Colors can be check on the interlaced monitor.

ramptest 1 1 1 0

**INTERLACED MONITOR, SINGLE BUFFER MODE, RGB**

init 2

dcr 40

ldconfig 0f

ramptest 1 0 0 1 ; The colormap is disabled in RGB mode. Ramptest

ramptest 0 1 0 1 ; requires 24 bitplanes for correct operation.

ramptest 0 0 1 1 ; Each of the colors can be checked on the

ramptest 1 1 1 1 ; interlaced monitor.

**NON-INTERLACED MONITOR, SINGLE BUFFER MODE, MAPPED**

init 2

dcr 3000

dcrtest 1 1

maptest 1 1 1f 0

fmttest 1 1 1f 0

ldconfig 0f

bptest 1 1 3f 0 0

ldconfig C

bptest 1 1 3f 0 0

init 2

dcr 3000

ldconfig 0f

stripetest 1 1 40 1

**NON-INTERLACED, DOUBLE BUFFER, MAPPED**

ldconfig 9

stripetest 1 1 40 0

**NON-INTERLACED, SINGLE BUFFER, MAPPED**

ldconfig 0f

ramptest 1 0 0 0

ramptest 0 1 0 0

ramptest 0 0 1 0

ramptest 1 1 1 0

**NON-INTERLACED, SINGLE BUFFER, RGB**

dcr 3040

ramptest 1 0 0 1

ramptest 0 1 0 1

ramptest 0 0 1 1

ramptest 1 1 1 1

**A sample test run using each of the codes above is listed below:**

iris &gt; b mt:bpcd

BPC Diagnostic: Tue Nov 6 08:53:14 PST 1984

UC4

DC4

Interface 3 (UC4 Multibus Interface)

PM2

Type &lt;space&gt; to abort the current command

[1] -&gt; init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cfr: display ab, update ab, viewport, backline, ldlinestip

mdr:

dcflags: v fsm-clear

significant bitplanes: FFFFFFFF

installed bitplanes: FFFFFFFF

color index: 0

write enable code: FFFFFFFF

[2] -> dclamptest 1

[3] -> dcr 0

dcflags: v fsm-clear

[4] -> dcrtest 1 1

pass 1: errors this pass = 0, total errors = 0

[5] -> maptest 1 1 1F 0

pass 1: 01234, errors this pass = 0, total errors = 0

[6] -> fntest 1 1 1F 0

pass 1: 01234, errors this pass = 0, total errors = 0

[7] -> ldconfig 0f

cfr: display ab, update ab

[8] -> bptest 1 1 3F 0 0

pass 1: 012345, errors this pass = 0, total errors = 0

[9] -> ldconfig C

cfr: update ab

[A] -> bptest 1 1 3F 0 0

pass 1: 012345, errors this pass = 0, total errors = 0

[B] -> init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cfr: display ab, update ab, viewport, backline, ldlinestip

mdr:

dcflags: v fsm-clear

significant bitplanes: FFFFFFFF

installed bitplanes: FFFFFFFF

color index: 0

write enable code: FFFFFFFF

[C] -> dcr 0  
dcflags: v fsm-clear

[D] -> ldconfig 0f  
cfr: display ab, update ab

[E] -> stripetest 1 1 40 1  
pass 1: Cannot test planes FFFFC0C0, A:67 B:67 C:01234567 D:01234567  
a0a1a2a3a4a5b0b1b2b3b4b5 badplanes: 0, cumulative: 0

[F] -> ldconfig 9  
cfr: display a, update b

[10] -> stripetest 1 1 40 0  
pass 1: Cannot test planes F0F00000, C:4567 D:4567  
a0a1a2a3a4a5a6a7b0b1b2b3b4b5b6b7c0c1c2c3d0d1d2d3 badplanes: 0,  
cumulative: 0

[11] -> ldconfig 0f  
cfr: display ab, update ab

[12] -> ramptest 1 0 0 0

[13] -> ramptest 0 1 0 0

[14] -> ramptest 0 0 1 0

[15] -> ramptest 1 1 1 0

[16] -> init 2  
configuration [0,FFFF], DF:  
mode [0,7], 0:  
dc register [0,FFFF], 0:  
color index [0,FFF], 0:  
write enable code [0,FFFFFFFF], FFFFFFFF:  
significant bitplanes [0,FFFFFFFF], FFFFFFFF:  
cfr: display ab, update ab, viewport, backline, ldlinestip  
mdr:  
dcflags: v fsm-clear  
significant bitplanes: FFFFFFFF  
installed bitplanes: FFFFFFFF  
color index: 0  
write enable code: FFFFFFFF

[17] -> dcr 40  
dcflags: rgb, v fsm-clear

[18] -> ldconfig 0f  
cfr: display ab, update ab

[19] -> ramptest 1 0 0 1

[1A] -> ramptest 0 1 0 1

[1B] -> ramptest 0 0 1 1

[1C] -> ramptest 1 1 1 1

[1D] -> init 2  
configuration [0,FFFF], DF:  
mode [0,7], 0:  
dc register [0,FFFF], 0:  
color index [0,FFF], 0:  
write enable code [0,FFFFFFFF], FFFFFFFF:  
significant bitplanes [0,FFFFFFFF], FFFFFFFF:  
cfr: display ab, update ab, viewport, backline, ldlinestip  
mdr:  
dcflags: v fsm-clear  
significant bitplanes: FFFFFFFF  
installed bitplanes: FFFFFFFF  
color index: 0  
write enable code: FFFFFFFF

[1E] -> dcr 3000  
dcflags: pipe4, prom, v fsm-clear

[1F] -> dcrtest 1 1  
pass 1: errors this pass = 0, total errors = 0

[20] -> maptestdcrtest 1 1  
pass 1: errors this pass = 0, total errors = 0



[21] -> maptest 1 1 1F 0  
pass 1: 01234, errors this pass = 0, total errors = 0

[22] -> fmtest 1 1 1F 0  
pass 1: 01234, errors this pass = 0, total errors = 0

[23] -> ldconfig 0f  
cfr: display ab, update ab

[24] -> bptest 1 1 3F 0 0  
pass 1: 012345, errors this pass = 0, total errors = 0

[25] -> ldconfig C  
cfr: update ab

[26] -> bptest 1 1 3F 0 0  
pass 1: 012345, errors this pass = 0, total errors = 0

[27] -> init 2  
configuration [0,FFFF], DF:  
mode [0,7], 0:  
dc register [0,FFFF], 0:  
color index [0,FFF], 0:  
write enable code [0,FFFFFFFF], FFFFFFFF:  
significant bitplanes [0,FFFFFFFF], FFFFFFFF:  
cfr: display ab, update ab, viewport, backline, ldlinestip  
mdr:  
dcflags: v fsm-clear  
significant bitplanes: FFFFFFFF  
installed bitplanes: FFFFFFFF  
color index: 0  
write enable code: FFFFFFFF

[28] -> dcr 3000  
dcflags: pipe4, prom, v fsm-clear

[29] -> ldconfig 0f  
cfr: display ab, update ab

[2B] -> stripetest 1 1 40 1  
pass 1: Cannot test planes FFFFC0C0, A:67 B:67 C:01234567 D:01234567  
a0a1a2a3a4a5b0b1b2b3b4b5 badplanes: 0, cumulative: 0

[2C] -> ldconfig 9  
cfr: display a, update b

[2D] -> stripetest 1 1 40 0  
pass 1: Cannot test planes F0F00000, C:4567 D:4567  
a0a1a2a3a4a5a6a7b0b1b2b3b4b5b6b7c0c1c2c3d0d1d2d3 badplanes: 0,  
cumulative: 0

[2E] -> ldconfig 0f  
cfr: display ab, update ab

[2F] -> ramptest 1 0 0 0

[30] -> ramptest 0 1 0 0

[31] -> ramptest 0 0 1 0

[33] -> ramptest 1 1 1 0

[34] -> dcr 3040  
dcflags: rgb, pipe4, prom, v fsm-clear

[36] -> ramptest 1 0 0 1

[37] -> ramptest 0 1 0 1

[38] -> ramptest 0 0 1 1

[39] -> ramptest 1 1 1 1

[3A] -> quit

**TEST**

`cprint` - test color printer

**SYNOPSIS**

`cprint` [file]

**DESCRIPTION**

The `cprint` program tests the Tektronix 4692 color printer.

The program uses a file which is the data for a color picture. The two files which are available are `CHESS` and `FLUTE`. The specified file is sent to the color printer.

**TEST**

cttest - test cartridge tape drive

**SYNOPSIS**

cttest

**DESCRIPTION**

The *cttest* program tests the quarter inch cartridge tape drive.

The program first does a TAR test using the 'tar' format and then does a CPIO test using the 'cpio' format.

The program writes to tape 22 data files, of different data types, then reads them into a separate directory. The program then compares the original files with the ones that were copied and then read back from tape. If an error occurs, the file the program was comparing at the time of the error will be printed out.

The following is the normal output for the tapetest:

```
QUARTER INCH CARTRIDGE TAPE TEST
  TAR Test
    Tar write test... complete
    Tar read test... complete
    Comparing Input with Output... complete
  CPIO Test
    Cpio write test... complete
    Cpio read test... complete
    Comparing Input with Output... complete
TAPE TEST COMPLETE
```

Type ^c (control c) to abort the test.

DEV

## Tests & Diagnostics

DEV

### TEST

dev - test mouse and keyboard input devices

### SYNOPSIS

dev

### DESCRIPTION

The *dev* program tests the mouse and keyboard devices.

Any movement of the mouse or pressing of the mouse buttons is simulated in the lower portion of the screen. Pressing keyboard keys causes the corresponding keys on the simulated keyboard to be highlighted.

Pressing the ESC key exits the program.

## DIAG

dust - test serial ports

## SYNOPSIS

dust

## DESCRIPTION

The *dust* program diagnose the processor serial ports.

Pressing the **h** key to display the following command listing:

- 1-4 - Test selected line
- a - All lines tested
- l - Loop toggle
- m - Modem test only toggle
- q - Quiet toggle
- v - Verbose toggle
- x - EXit

Connect loop back connector to port to be tested and then type 1, 2, 3, 4 to select the port and start the test.

Press the Space Bar to abort the current test.

Type x to exit the program.

enet - test ethernet communications

enet <hostname>

The *enet* program tests the ethernet communications.

If the workstation is running 'xns ethernet protocol' then the program will do the following:

The program first does a 'xx host date' to get the date from a host. Then it does a 'xcp file host:dir' to copy a file to the host directory. Then it does a 'xcp host:file newfile' to copy the same file from the host. If the test passes with no errors, the program will display 'END OF TEST'.

If the workstation is running 'tcp ethernet protocol' then the program will do the following:

The program first does a 'rsh host date' to get the date from a host. Then it does a 'rcp file host:dir' to copy a file to the host directory. Then it does a 'rcp host:file newfile' to copy the same file from the host. If the test passes with no errors, the program will display 'END OF TEST'.

The following is the normal output of the program:

START OF XX TEST

```
1: <date>
    xx host ok
2: <date>
    xx host ok
3: <date>
    xx host ok
4: <date>
    xx host ok
5: <date>
    xx host ok
6: <date>
    xx host ok
7: <date>
    xx host ok
8: <date>
    xx host ok
9: <date>
    xx host ok
10: <date>
    xx host ok
END OF XX TEST
```

START OF XCP TEST

```
copying data to host
    copy to host ok
copying data from host
    copy from host ok
END OF XCP TEST
```

Type ^c (control c) to abort the test.

## PROC

fex - generic procedure for fex programs

## SYNOPSIS

fex

## DESCRIPTION

The *fex* programs (*mdfex*, *sifex*, *ipfex*) diagnose the disk for bad blocks which cause hard errors. When a hard error occurs, note the bad block indicated in the error message. This bad block should be added to the bad block list as shown below in the disk restoration procedure.

## READ IN FEX FROM TAPE:

- Insert bootable tape.
- Enter *h* for command list.
- Enter proper command to boot from tape.

Example: *b mt:mdfex*

## ENTER FIELD ENGINEER PORTION OF PROGRAM:

- Type *Z* to get the password prompt.
- Enter the correct password.

Example: *donl*

## SELECT DISK DRIVE UNIT:

- Type *s* to set miscellaneous variables.
- Type *u* to select drive.
- Enter *0* or *1* to select drive to be restored.
- Type *i* to initialize selected drive.

## CHECK BAD BLOCK LIST:

NOTE: The Map Error Table is a list of already known bad blocks.

Insure these bad blocks are included in the bad block list.

- Type *b* to enter bad block edit mode.
- Type *a* to add bad blocks.
- Now enter any additional bad blocks, in the format of *cylinder/head*.  
(i.e. *31/0* or *921/6*).
- Press RETURN to end.
- Type *q* to quit bad block edit mode.



**FORMAT DISK:**

- Type **f** to format disk.
- Enter **go** to start.

NOTE: Formatting takes 8 minutes.

**EXERCISE DISK:**

- Type **s** to set miscellaneous variables.
- Type **w** to set write lock.
- Enter **off** to set write lock off.
- Type **e** to enter exercise mode.
- Type **c** for complete write/read test.
- Press RETURN to repeat test forever.
- Press RETURN for no alternate units.

NOTE: One test cycle takes 8 minutes. This test should be run for at least 12 hours and preferably 24 hours for a thorough check of all blocks; but do what time allows.

- Press Esc to abort test.
- Type **e** for error display.
- Type **d** to display errors.
- Now record list of bad blocks, if any, on paper.
- Type **q** to quit exercise mode.

**ADD NEW BAD BLOCKS TO BAD BLOCK LIST:**

- Type **b** to enter bad block edit mode.
- Type **a** to add bad blocks.
- Now enter each entry from your list recorded on paper.
- Press RETURN to end.
- Type **q** to quit bad block edit mode.

**REFORMAT DISK TO SET ALTERNATE TRACKS:**

- Type **f** to format disk.
- Enter **go** to start.

NOTE: Formatting takes 8 minutes.

**COPY ROOT FILE SYSTEM FROM TAPE TO DISK:**

- Type **t** for tape to disk utility.
- Press RETURN to copy from tape file (2). (root file system)
- Press RETURN to copy to selected unit.

- Press RETURN to copy to file system (a). (root file system)
- Enter go to begin copying.  
NOTE: Copying takes 4 minutes.
- Type q to quit FEX.
- Type y to confirm.

#### COPY /USR FILE SYSTEM FROM TAPE TO DISK:

- Boot UNIX from disk.  
Example: *b* or *b md0:vmunix*
- Enter *newfs md0c* to make /usr file system on unit 0  
or enter *newfs md1c* to make /usr file system on unit 1.  
NOTE: The command *newfs* does a *mkfs* using the  
disk size specifications for the file system specified.
- Enter *y* to confirm.
- Enter *mount /dev/md0c /usr* to mount /usr.
- Enter *cd /usr* to change directory to /usr.
- Enter *smt rew* to rewind tape.
- Enter *smt fsf* to position tape to file #2.
- Enter *smt fsf* to position tape to file #3.
- Enter *cpio -iahduv2* to copy /usr from tape to disk.

#### RESTORE DISK DRIVE LABELS:

- Enter the *sgilabel* command to label the disk drive.  
Usage: *sgilabel [-n name] [-s serial#] dv#*  
Example: *sgilabel -n "Release GL1-W2.1" -s 14596 md0*
- Use the *labelit* command to label each file system.  
Usage: *labelit /dev/r???? [fsname volume [-n]]*  
NOTE: The question marks designate the file system to relabel;  
and the -n option skips the label check before relabeling.
  - a. Label usr file system first.  
Example: *labelit /dev/rmd0c usr W2.1*
  - b. Label root file system last.  
Example: *labelit /dev/rmd0a root W2.1*
- Press the reset button to reboot the system.  
NOTE: Do not enter *reboot* or else the root label will not be updated.
- Boot up unix again to ensure proper operation.

TEST

*fpctest* - test floating point board

SYNOPSIS

*fpctest*

DESCRIPTION

The *fpctest* program tests the floating point board.

For 2400 or 2500:

Just run the test and the program will indicate if the floating point passed or failed the test.

For 2400T, 2500T, 3020, 3030:

*fpctest* exercises all numerical floating point operations.

If any errors occur, *fpctest* returns a non-zero exit code and displays a summary of errors.

USAGE: *fpctest* <verbose> <# of operations> <Option Reg>  
(arguments shown in <> are in hex)

If verbose is set to 1, each error is reported in detail in addition to the summary.

If verbose is set to 2, an extended summary is displayed.

Default value for Option Reg is hex '01'.

ERROR REPORT - 2c (hex) FP OPERATIONS COMPLETED:

opcode	SGL errors	SGLUPD errors	DBL errors	DBLUPD errors
ADD	0	0	0	0
SUB	0	0	0	0
REVSUB	0	0	0	0
MUL	0	0	0	0
DIV	0	0	0	0
REVDIV	0	0	0	0
INT-FLOAT	0	0	0	0
FLOAT-INT	0	0	0	0
SGL<->DBL	0	0	0	0
NEGATE	0	0	0	0
TST	0		0	
CMP	0		0	

divbyzero = 0, overflow = 1, underflow = 3, nan = 0

denormalize = 0, inexact = 16, illegalop = 0

## DIAG

gf - diagnose the graphic board

## SYNOPSIS

gf

## DESCRIPTION

The gf program is used to diagnose the graphic board.

Enter ? to display the following command list:

bpc op <type> <code> <data>  
Breakpoint set [no LOADOUT]  
call macro <name>  
Callable macro [#n] display  
dismiss interrupt  
fifo/file/test <which>  
geom. engine cmd <which>  
Geometry engine test spiral  
init hardware <runmode>  
Init tables after ir  
monitor format toggle  
Mask GE's force to <n> after init  
open microcode loc. <n>  
print <which> <how>  
<CR> print next loc.  
quit this command level  
^Re-init screen  
ram test <which>  
store <which> <how>  
Setup for vector testing ('SD' for dummy FIFOs)  
test gf board  
Token flag test  
Unset breakpoint(s): Ua = all UI <n> loc. n  
vector testing  
Version of microcode force to <n>  
write microcode  
Write microcode, preserve breakpts

Enter q to quit program.

## DIAG

ip2diag - diagnose the processor boards

## SYNOPSIS

ip2diag

## DESCRIPTION

The *ip2diag* program is used to diagnose the processor boards; IM1 memory board, FP1 floating point board.

Enter ? to show the command listing:

badbyte	fpdiag	lowtest	readbyte	touttest	?
defmacro	fpwalktest	macro	readword	writebyte	!
errors	getperror	mbmap	readlong	writeword	
fpctest	help	memtest	revision	writelong	
fpreinit	hightest	partest	rtc	\$	
fpooverlap	history	quit	seemacro	.	

## HELP SCREEN

BACKSPACE	Delete character
DELETE	" "
CTRL-W	Delete argument
TAB	Use default argument
CTRL-L	Redraw command line
CTRL-U	Erase command line
?	Help
SPACE	Interrupt command execution

Type <command>? to get an explanation of command.

A command-line consists of a command string followed by zero or more arguments. Partially typed command strings are expanded, if possible, when a <space> is typed. Arguments are signed hex numbers separated by a single <space>. Additional spaces typed between arguments prompt for additional arguments, until non are left. Typing any character other than a space causes all prompts to disappear. Arguments that are not typed are given (compiled-in) default values, which are displayed after the carriage return is typed.

**DIAG**

*ipfex* - 474meg disk utility

**SYNOPSIS**

*ipfex*

**DESCRIPTION**

The *ipfex* program is used to diagnose bad blocks on the 474meg disk drive. Refer to the generic procedure *fex* for program operation.

**TEST**

knobs - test dial & button box

**SYNOPSIS**

knobs

**DESCRIPTION**

The *knobs* program tests the dial & button box.

The purpose of the program is two-fold:

1. To exercise all the dials and all the buttons on the two boxes as well as the text display and the leds for each of the buttons.
2. To show a simple example of interfacing to the Dial & Button Box using the graphics library.

The program is basically an enhanced version of "etch-a-sketch". The dials are used in pairs (one for X movement, one for Y movement).

**Program start up:**

The program begins by clearing the screen to a neutral "etch-a-sketch grey" and then proceeds to light up all of the leds in the button array one at a time. It starts out with the led in the top left hand corner, goes across the row to the right, goes to the left side of the next row, and proceeds until it gets to the bottom right hand corner, and then starts all over again. While this is happening, the text display should read "PRESS 31". Press the right button (button 31) of the bottom row, to proceed with the test.

**Dial pair selection:**

There are four pairs of valuator (dials) available to move and draw on the screen. Only one of the pairs is available at any one time. Which pair is active is determined by the four buttons on the left-most column of the button array. The top button of the column selects the top pair, the next button selects the next pair down, and so on. The left dial of the pair controls the X movement while the right one controls the Y movement.

Choosing a pair will cause the led for the appropriate button to light up.

**Drawing color selection:**

Lines can be drawn on the screen in any of 7 different colors. There are two ways to change the current line drawing color. One is to press one of the color selection buttons in the button array (the four buttons on the top row of the array and the three left-most buttons on the bottom row). The other way to change the color is to press the right mouse button. For each time the mouse button is pressed the color increments by one. When the color is color 7 (cyan) and the button is pressed, the color starts back at color 1 (black).

In either case, choosing a new color will cause the button led for that color to light up.

**Clearing the screen:**

To clear the screen, press the right-most button of the bottom row.

**Using the mouse buttons:**

The left mouse button allows you to draw a line from the current screen position to the position of the mouse cursor when the button is pressed. The line is drawn in the current line drawing color. Although you can place the mouse cursor anywhere on the screen, the line will not extend beyond the X and Y limits given below:

	MIN	MAX
	-----	-----
X:	100	924
Y:	100	678

The middle mouse button allows you to move the current screen position anywhere within the limits given above, but no line will be drawn from the previous screen position to this new screen position.

The right mouse button, as discussed before, allows you to cycle through the available drawing colors by pressing the button. Each time the button is pressed, the next color becomes the current drawing color. The led for the color selection buttons changes as the mouse button is pressed to indicate which color is the current drawing color.



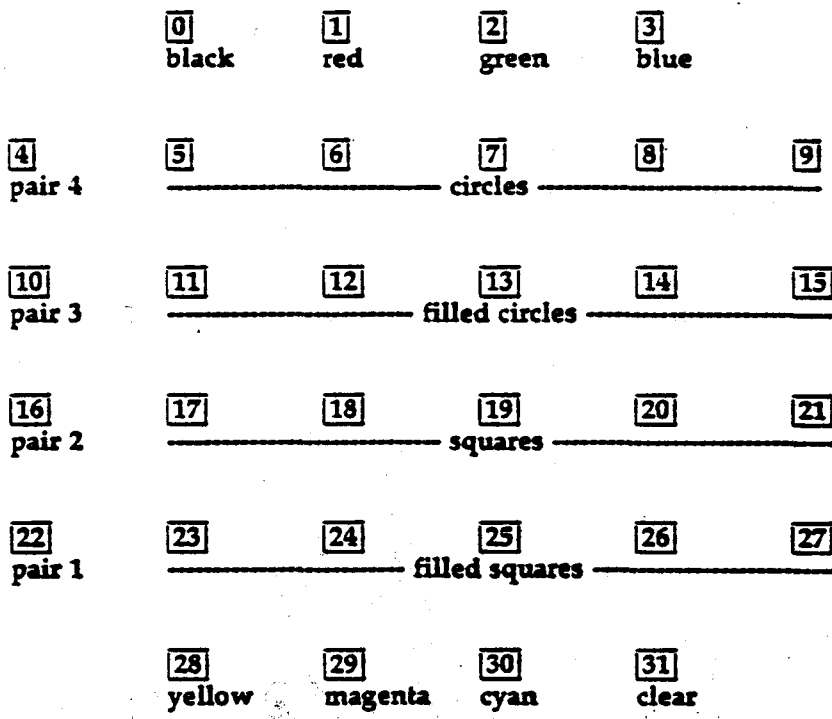
**Drawing circles and squares using the buttons:**

The five buttons in the second row from the top draw circles in various sizes. The five buttons in the third row draw filled circles of various sizes. The five buttons in the fourth row draw squares of various sizes. The five buttons in the fifth row draw filled squares of various sizes.

For each of these rows, the buttons on the left draw the larger objects, while the buttons on the right draw the smaller objects. Each time any of these buttons is pressed, the text display will show either "CIRCLE" or "SQUARE" depending on the button you pushed.

**Summary of button functions:**

Switch (button) numbers are shown in each location.



**TEST**

lpen - test the lightpen

**SYNOPSIS**

lpen

**DESCRIPTION**

The *lpen* program tests the lightpen.

This test should be performed inside the window manager (*mex*). The screen will turn all one color. To operate the light pen, point the lightpen at the screen. Small squares should appear, if not, move the lightpen closer to the screen. Pushing the light pen to the screen should draw a larger square each time. The intensity can be changed by pushing the mouse buttons, left for up, middle for down.

Point the pen away from the screen. Depress the tipswitch, and check for the track software response. The tipswitch value should toggle 0/1. There shouldn't be any dots written to the display screen at this point. If spurious dots are displayed when the pen aperture is covered, while pressing the tip switch up/down, the pen is transmitting erroneous data and should be marked "FAILED".

Press the ESC key to exit the program.

**HARDWARE REQUIREMENTS**

lightpen, lightpen board, associated cables

**MDFEX**

**Tests & Diagnostics**

**MDFEX**

**DIAG**

*mdfex* - 72meg disk utility

**SYNOPSIS**

*mdfex*

**DESCRIPTION**

The *mdfex* program is used to diagnose bad blocks on the 72meg disk drive. Refer to the generic procedure *fex* for program operation.

**DIAG**

memtest - Exercise PM2M memory boards for proper operation.

**SYNOPSIS**

**memtest**

**DESCRIPTION**

The *memtest* program exercises PM2M memory board for proper operation.

The following is the list of the available tests:

- Test 1: constant 0s
- Test 2: constant 1s
- Test 3: stripes 0xAAAA
- Test 4: stripes 0x5555
- Test 5: random data
- Test 6: write-read-read test
- Test 7: write-delay-read test
- Test 8: random byte data
- Test 9: reverse byte data

The following are the questions asked by the program:

- Bottom of memory block <CR for default of 0x410C>?
- Top of memory block <CR for default of 0x40000>?
- Test number? <CR for all>
- Suppress errors? <CR for no>
- Continue on error? <CR for no>
- Loop on soft error? <CR for no>
- Disable parity? <CR for no>

Type ^c (control c) to abort test in progress.

Memtest is a memory exerciser for both the PM2M expansion memory, and the PM2 on-board memory. The program verifies proper memory operation and data under various patterns. The program also tests proper operation of the multibus map, memory access via program will detect parity failures and report them, but it will not identify exactly where the parity error(s) occurred.

Memtest tries to identify failures to the memory chip which is at fault. If there are grounded or shorted memory data in/out lines, this will probably lead to whole rows of chips reported as bad. The correlation between failures and chip position is theoretical and should be treated as a clue only.

Press the RESET switch to quit the program.

**TEST**

pattern - checks monitor adjustments

**SYNOPSIS**

pattern

**DESCRIPTION**

The *pattern* program checks the monitor adjustments.

This program shows various test screens. Hold down the left mouse button to show the menu. Move the mouse to select a pattern and then release the mouse button to display the selected pattern. The menu consists of the following selections:

- \* 100% white field
- \* 90% white field
- \* Focus 1
- \* Focus 2
- \* Focus 3
- \* Raster regulation
- \* Gray scale
- \* Convergence
- \* Black level 1
- \* Black level 1
- \* Quit

Select Quit to exit the program.

SIFEX

Tests & Diagnostics

SIFEX

DIAG

sifex - 170meg disk utility

SYNOPSIS

sifex

DESCRIPTION

The *sifex* program is used to diagnose bad blocks on the 170meg disk drive. Refer to the generic procedure *fex* for program operation.

**timeout** – sets the time for display blanking

**timeout** <number>

100 = 1.6 seconds  
1000 = 16 seconds  
10000 = 160 seconds  
100000 = 1600 seconds

The *timeout* program sets the time for display blanking.

To use, type '*timeout tics*'. This will determine how much time the display will show on the screen before it goes blank.

If you type '*timeout 1000*' on a 2400, the screen will stay on for ~20 seconds.



**TEST**

*zclip* - test z-clipping

**SYNOPSIS**

*zclip*

**DESCRIPTION**

The *zclip* program tests the z-clipping geometry engines.

This program displays an object which can be moved towards and away from the point of view. As the object is moved, it quickly encounters the z-clipping planes, if the z-clipping hardware exists.

Without pressing the mouse buttons, moving the mouse from left to right changes the location of the object. Pressing any mouse button restarts the object in the initial position.

Press the ESC key to exit the program.

**HARDWARE REQUIREMENTS**

z-clipping option

# Preliminary DC4 Test Procedure

Anthony F. Roaque

Silicon Graphics, Inc.

## 1. INTRODUCTION

### 1.1 Display Controller

Generally, the display controller performs two distinguishable functions. The first of which is memory and monitor timing signal generation; The second is display generation with reference to the color maps, rgb data paths, DAC's etc.. The objective of this document is to provide an efficient, yet complete, test procedure for the DC4.

### 1.2 BPCDXXXX.r Overview

The program bpcdXXXX.r contains the routines required to test the GL2 or GL1 hardware. Revision four of the display controller, or DC4, is backward compatible with any previous vintage of control/bitplane boards. The diagnostic code as described in this document contains routines which test the bitplanes, update controller, and the DC4. Other choices include routines to display a variety of test patterns, read/write control registers, waveform generation, etc.. As an example (referencing the DC4), programs such as ramptest generate test patterns in the DAC area. Using an oscilloscope, ramptest can be a useful tool in troubleshooting a display problem. Only the routines for testing the DC4 are described in this document. The entire command list is shown below:

```
[15] -> cmdndlist
ball          drawline      lded          randrect      stripetest
bpctest      endmacro      ldfmaddr     readdda       stripe
bplamptest   fillrect      ldmode       readfm        swap
bptest       filltrap      ldrepeat     readloop      time
chartest     fmtest        ldxe         readmap       traptest
clear        help          ldxs         readword      uclamptest
cmdndlist    history       ldye         read          ucr
colorcode    init          ldys         recttest     wecode
colorwetest  joescale     linetest     request       we
color        ldconfig     loopclearword restore        writefm
dclamptest   ldddasaf     loop         revision      writemap
dcpal        ldddasai     macro        rgbcolor     writeloop
dcrtest      ldddasdf     mapcolor     rotateloop   writeword
dcr          ldddasdi     maptest      save          write
ddatest     ldddaeaf     memtest      screenmask   $
defmacro     ldddaeai     pixeltest    scrmsktest   .
depthcue     ldddaedf     quit         seemacro     ?
display      ldddaedi     ramptest     sigplanes    !
drawchar     ldec         randlines    steptest
```

### 1.3 DC4 System Hardware

The DC4 can be tested in a limited bitplane environment, however, in order to facilitate complete testing, the maximum amount of bitplanes must be utilized. The tests provided by the diagnostic program require the use of an update control board, an interlaced color monitor, and a non-interlaced color monitor. The GF board is not required for testing the DC4. Note that the maximum amount of bitplanes is thirty-two, or eight bitplane boards. Only twenty-four of those can actually be displayed. The hardware configuration required for testing the DC4 is listed below:

- o Processor
- o Ethernet
- o Memory
- o 8 Bitplanes
- o Display Controller
- o Update Controller
- o interlaced color monitor
- o non-interlaced color monitor
- o Visual 50 Terminal
- o Misc. cables, etc.

## 2. PROCEDURE

### 2.1 System Configuration

A table shows the cable connections for the DC3 and DC4:

BP3 to Display Controller Cable Connections							
BP3 settings		UC3/DC3 mode		UC4/DC4 mode			
Bd #	Bitplanes	BP3	DC3	BP3	DC4	BP3	DC4 Non-Int*
0	A0,A1,B0,B1	J1	J1	J1	J1	J3	J4
1	A2,A3,B2,B3	J1	J1	J1	J1	J3	J4
2	A4,A5,B4,B5	J1	J1	J1	J1	J3	J4
3	A6,A7,B6,B7	J1	J1	J1	J1	J3	J4
0	C0,C1,D0,D1	J1	J4	J1	J5	J3	J6
1	C1,C2,D1,D2	J1	J4	J1	J5	J3	J6
2	C3,C4,D3,D4	-	-	-	-	-	-
3	C6,C7,D6,D7	-	-	-	-	-	-

NOTE: \* In non-interlaced mode, use *both* "DC4" and "DC4 Non-Int" cable connections.

The four position switch settings for the DC4 are listed as follows:

- Configuration for -01 version -

Switch: A15 = 1 (open)  
 A14 = 0 (closed)  
 A13 = 1 (open)  
 A12 = 1 (open)  
 A11 = 1 (open)

The 'jumpers' that are required on the board are specified on the assembly drawing for the board. Both interlaced and non interlaced monitors must be cabled to the 1400. The system is setup to boot from the Visual 50 terminal. This will provide the user with the capability to test each of the color monitors independently from one another. A Null modem cable is required to connect port 2 to the Visual 50.

### 2.2 Diagnostic Program Test Procedure

The diagnostic program is down-loaded via ethernet and depending on the test system configuration, one of the following versions of the test programs are executed:

- bpcd3332.r for UC3, DC4, PM2 systems
- bpcd4431.r for UC4, DC4, PM1 systems
- bpcd4432.r for UC4, DC4, PM2 systems

The following is a list of routines that are exercised at one point or another while testing the DC4. These routines are derived from the command list described in section 1.0.

bptest  
clear  
colorcode  
color  
dclamptest  
dcpal  
dcrtest  
dcr  
display  
drawline  
endmacro  
fillrect  
fmttest  
help  
history  
init  
joescale  
ldconfig  
loop  
macro  
mapcolor  
maptest  
ramptest  
randlines  
randrect  
readloop  
readmap  
read  
rgbcolor  
sigplanes  
steptest  
stripetest  
stripe  
writemap  
writeloop  
writeword  
write

Several modes must be tested such as double buffered mode, interlaced, non-interlaced mode, single buffered mode, etc.. These test modes are described in the sample test run section of this document. When typing a '?' after the command line, a description of the command is printed. The command lines that pertain to this document, have been recorded as part of this document, to aid in the understanding of the diagnostic program. It is recommended that the following is read and understood before reading the data shown in the sample test run. The description for each of the pertinent commands are shown below:

[1] -> bptest

bptest <iterations> <verbose> <tests> <chip report> <diag>

Test the operation of the bitplane memory. Errors are reported as they occur if verbose is true (not 0). Otherwise only an error total is reported. Individual tests are selected with bits of the 'tests' argument.

bit	test
1	0 All 0 test.
2	1 All 1 test.
4	2 Word = address.
8	3 Word = complement of address. UC4 only.
10	4 Word takes random value.
20	5 Exercise saveword/drawword hardware.

*3F all 7-7-*

[2] -> clear

clear <x stipple> <y stipple>

Clear the 1024x1024 bitplane memory to the current colorcode. A 16x16 (8x16 for uc3) pattern that is the bitwise OR of the horizontal and vertical patterns is used. Wecode is not changed.

[3] -> colorcode

colorcode <code>

Set the (up to) 32-bit colorcode. Current color index becomes invalid. Code bits are always ordered D7-0, C7-0, B7-0, A7-0, regardless of the current swizzle or double values. Cfr and mdr are restored after the code has been written.

[4] -> color

color <index>

The given color index is converted into a code, taking into account the current cfr, dcr, and sigplane values. This code becomes the current colorcode, and is written into the bitplanes.

Nothing is changed if no argument is given. The current (new) value is always printed.

[5] -> dclamptest

dclamptest <iterations>

Individually turn on each of the display control LEDs. DC4 'optclk' stays on longer for obscure reasons.

[6] -> dcpal

dcpal <value>

Set the VFSM (video finite state machine) to the specified value (range 0-7) (no change if no argument is given). Then sample the VFSM outputs during the following vertical retrace periods and print the results.

[7] -> dcrtest

dcrtest <iterations> <verbose>

Test the display controller multibus register.

[9] -> dcr

dcr <dcr>

Set the display control register to a new value. The contents are not changed if no argument is given. The current (or new) value is printed.

bit DC3    DC4

```

---  ---
0 bus  reg0
1 reg  reg1
2 rgb  reg2
3 maint reg3
4 reg0 bus
5 reg1 reg
6 reg2 rgb
7 reg3 highmap
8 pal0 pal0
9 pal1 pal1
a pal2 pal2
b      optclk
c      pipe4
d      prom
e      1k
f      mbit

```

[A] -> display

display <x word address> <y address>

The color indexes of a word-aligned 64x20 screen area are printed.

Each index is masked by 0xf so that it can be printed as a single character.

The special character '.' is printed if and only if ALL bits of that pixel are 0.

[B] -> drawline

drawline <x1> <y1> <x2> <y2> <stipple>

Draws a line from x1,y1 to x2,y2 with the specified stipple.

[C] -> endmacro

endmacro

[E] -> fillrect

fillrect <x> <y> <width> <height> <x stipple> <y stipple>

Fills the specified rectangle with the current color. A 16x16 (8x16 for uc3) pattern is specified as the bitwise OR of the given horizontal and vertical patterns.

[10] -> help

help

BACKSPACE Delete character

DELETE " "

CTRL-W Delete argument

TAB Use default argument

CTRL-L Redraw command line

CTRL-U Erase command line

? Print help text - type at any time

SPACE Interrupt command execution

A command-line consists of a command string followed by zero or more arguments. Partially typed command strings are expanded, if possible, when a <space> is typed. Arguments are signed hex numbers separated by a single <space>. Additional spaces typed between arguments prompt for additional arguments, until none are left. Typing any character



other than a space causes all prompts to disappear. Arguments that are not typed are given (compiled-in) default values, which are displayed after the carriage return is typed.

BPCD supports nested (not recursive) user-defined macros, and includes both history and argument substitution mechanisms.

BPCD was written by Kurt Akeley, who is happy to hear positive comments and suggestions. Complaints (on the other hand) should be directed to Dave Koontz, who is now responsible for the code.

Take drugs and prosper ...

[12] -> history

history <command number>

A history of previous commands is maintained. Typing 'history' with no argument causes the list of recently typed commands to be printed.

Each command in this list is preceded by its number. A remembered command can be re-executed by typing 'history <number>'. Alternately, the n'th previous command can be re-executed by typing 'history -n'.

[13] -> init

init <mode>

Initialize the raster subsystem, or return it to its initialized state.

Must be called before any UC command is executed. Arguments are:

none return all values to defaults.

1 change the default init values

2 size the bitplanes to determine the default values

3 interface init only, minimum state change to the raster system

4 like 'none', but bitplanes are not altered

[14] -> joescale

joescale

Draws a pattern useful for monitor adjustment. Named in honor of  
Jok'in Eddie.

[15] -> ldconfig

ldconfig <value>

Set the uc configuration register to a new value. The current value is  
not changed if no argument is given. The current (or new) value is  
printed.

bit	uc3	uc4
---	---	---
0	display a	display a
1	display b	display b
2	update a	update a
3	update b	update b
4	screenmask	screenmask
5	enable cd	invert
6	finish line	finish line
7	ldlinestip	ldlinestip
8		pfi cd
9		pfi read
a		pfi column
b		pfi x down
c		pfi y down
d		all pattern
e		pattern 32

**[16] -> loop**

loop <command> <data>

Continuous requests until interrupted. Read-requests do read, but they do not print a value. See 'request' for a list of command assignments.

**[17] -> macro**

macro <number> <iterations>

Execute a user-defined macro a specified number of times.

**[18] -> mapcolor**

mapcolor <index> <red> <green> <blue>

All colormap entries that correspond to index are changed to the new red, green, and blue values. An index corresponds to multiple map entries if not all of the planes connected to the colormap are 'significant' (see sigplanes).

Index is in the range  $0..(2^{*n})-1$ , where n is the number of significant planes that are currently connected to the colormap.

Red, green, and blue are 8-bit intensities, ranging from 0 (off) to FF (brightest). All arguments are given in hex.

For example, if only bitplane A0 is significant (sigplanes = 1) then the command 'mapcolor 0 ff ff ff' will set every other colormap entry (0, 2, 4, etc) to 'ff ff ff'. Calling mapcolor with no significant planes sets the entire colormap.

**[19] -> maptest**

maptest <iterations> <verbose> <tests> <diag>

Test the operation of the colormap. Errors are reported as they occur if verbose is true (not 0). Otherwise only an error total is printed.

Individual tests are selected with bits in the 'tests' argument.

bit test  
--- ----

- 0 Red, green, blue = 0.
- 1 Red, green, blue = FF.
- 2 Red, green, blue = address & FF.
- 3 Red, green, blue = address & F00.
- 4 Red, green, and blue = different random values.

[1A] -> ramptest

ramptest <red> <green> <blue> <rgb>

Fill the screen with a ramp of intensity increasing from left to right.

Each of red, green, and blue is either ramped or not depending on whether the corresponding argument is false (0) or true (not 0). RGBmode disables the colormap; it requires 24 bitplanes for correct operation.

[1B] -> randlines

randlines <seed> <number> <stipple>

Draws the specified number of lines with the specified stipple pattern in the current color. 'Seed' initializes the random number generator determines the line endpoints. Any value will do.

Be sure to clear the screen to a color that is different from the current color, or you won't see anything.

[1C] -> randrect

randrect <log2 size>

Continuous random fill of the screen with rectangles. Color indexes 0, 1, 2, and 3 are used with 4 different patterns. Rectangle dimensions range from 1 through 2 raised to the power of the passed argument. For example, 'randrect 8' draws rectangles with dimensions in the range 1..255.

[1D] -> readloop

readloop <i/o address>

Read continuously any multibus i/o space location, and do NOT print the 16-bit results. The i/o space is a 16-bit space. Address bits 0 and 16..31 are ignored. Word access only (no byte reads).

[1E] -> readmap

readmap <address> <locations>

Print consecutive colormap values. The 'address' argument is an absolute colormap address in the range 0..FFF. It is NOT a color index.

[1F] -> read

read <i/o address>

Read any multibus i/o space location, and print the 16-bit result. The i/o space is a 16-bit space. Address bits 0 and 16..31 are ignored. Word access only (no byte reads).

[1] -> rgbcolor

rgbcolor <red> <green> <blue>

Sets the colorcode to the given red, green, and blue values (range 0..ff).

Color index becomes invalid.

[20] -> sigplanes

sigplanes <planes>

Specify which bitplanes are significant, in terms of index-to-code conversion and colormap changes. Sigplanes is not changed if no argument is given. The current (new) value is always printed.

[2] -> steptest

steptest <red> <green> <blue> <rgb>

Like ramptest, but draws 10 vertical stripes of increasing intensity, instead of 256.

[21] -> stripetest

stripetest <iterations> <verbose> <fields> <singlebuf>

Fills each significant plane with a 3-off/3-on pattern, then uses the VFSM to determine whether this pattern is being correctly received at the output of the colormap. The colormap is adjusted to output only a function of the bitplane being tested.

Errors are reported as characters in parentheses after the plane name.

DC3 characters are 'e' and 'o', indicating errors in even and/or odd planes. DC4 errors are '0', '1', '2', and '3' to allow pipe4 error reporting.

[22] -> stripe

stripe <width> <step>

Fill the screen with screen-height rectangles of the specified width, on the specified horizontal pitch. For example, 'stripe 3 6' draws rectangles that are 3 pixels wide, with 3-pixel spaces (pitch of  $6=3+3$ ).

[23] -> writemap

writemap <address> <red> <green> <blue>

Change a single colormap location. The 'address' argument is an absolute colormap address in the range 0..FFF. It is NOT a color index.

[25] -> writeloop

writeloop <i/o address> <data>

Write continuously to any multibus i/o space address. Interrupt to stop.

The i/o space is a 16-bit space. Address bits 0 and 16..31 are ignored.

Word access only (no byte writes).

[1] -> writeword

writeword <x word address> <y address> <data word>

All planes of the specified bitplane word are loaded with value.

This is accomplished by first setting all 16 pixels to all 0's, then

setting the appropriate pixels to 1's. After the write is completed,

a UC SAVEWORD command is executed so that the bitplane registers are

also set to the specified value (for probe/scope purposes).

[2] -> write

write <i/o address> <data>

Write to any multibus i/o space address. The i/o space is a 16-bit space.

Address bits 0 and 16..31 are ignored. Word access only (no byte writes).

### 2.3 Sample Test Run

The sequence of tests for the DC4 are shown below. Each step must follow consecutively as listed. If an error occurs, or if a retest is desired at a specific point in this list, restart the command sequence at the PREVIOUS 'init 2' command statement.

## INTERLACED MONITOR, SINGLE BUFFER MODE, MAPPED

```
init 2 ; The DC is mapped into the multibus address space.
dclamptest 1 ;The lamp test checks for address acknowledge and
dcr 0 ;gives visual feedback of correct DCR operation.
dcrtest 1 1 ;If bus errors result, use a logic probe and the
maptest 1 1 1f 0 ;DCR command to trace the acknowledge path.
fmtest 1 1 1f 0 ;
; DCRTEST checks for stuck and shorted bits in DCR.
ldconfig 0f ;Performs a rudimentary test on the video fsm PAL.
bptest 1 1 3f 0 0 ;
;MAPTEST looks for both data and address errors in
ldconfig C ;the color maps. READMAP and WRITEMAP can be used to
bptest 1 1 3f 0 0 ;interrogate and alter the colormap contents if
;problems are encountered.

init 2 ;
dcr 0 ;
ldconfig 0f ; FMTEST tests the dfsm generation of memory signals
stripetest 1 1 40 1 ;including refresh.
;
; STRIPETEST requires 24 bitplanes for correct
;operation
```

## INTERLACED MONITOR, DOUBLE BUFFER MODE, MAPPED

```
ldconfig 9
stripetest 1 1 40 0 ;STRIPETEST double buffered and single buffered
;checks both data paths for each mode.
```

## INTERLACED MONITOR, SINGLE BUFFER MODE, MAPPED

```
ldconfig 0f
ramptest 1 0 0 0 ;Mapped mode. Use an oscilloscope to individually
;check each DAC output. All 256 levels should blend
;into a smooth ramp. The ramp amplitude (with only
;the on-board 75 ohm resistor as termination) should
;be 1.5 volts peak-to-peak. As shown, RED is active.
;the DAC for the RED output can be checked. Requires
ramptest 0 1 0 0 ;24 bitplanes for correct operation.
ramptest 0 0 1 0 ;Colors can be check on the interlaced monitor.
ramptest 1 1 1 0
```

## INTERLACED MONITOR, SINGLE BUFFER MODE, RGB

```
init 2
dcr 40

ldconfig 0f
ramptest 1 0 0 1 ; The colormap is disabled in RGB mode. Ramptest
ramptest 0 1 0 1 ;requires 24 bitplanes for correct operation.
```



rampstest 0 0 1 1 ;Each of the colors can be checked on the  
rampstest 1 1 1 1 ;interlaced monitor.

#### NON-INTERLACED MONITOR, SINGLE BUFFER MODE, MAPPED

init 2  
dcr 3000  
dcrtest 1 1  
mapstest 1 1 1f 0  
fntest 1 1 1f 0

ldconfig 0f  
bptest 1 1 3f 0 0

ldconfig C  
bptest 1 1 3f 0 0

init 2  
dcr 3000

ldconfig 0f  
stripetest 1 1 40 1

#### NON-INTERLACED, DOUBLE BUFFER, MAPPED

ldconfig 9  
stripetest 1 1 40 0

#### NON-INTERLACED, SINGLE BUFFER, MAPPED

ldconfig 0f  
rampstest 1 0 0 0  
rampstest 0 1 0 0  
rampstest 0 0 1 0  
rampstest 1 1 1 0

#### NON-INTERLACED, SINGLE BUFFER, RGB

dcr 3040  
rampstest 1 0 0 1  
rampstest 0 1 0 1  
rampstest 0 0 1 1  
rampstest 1 1 1 1

A sample test run using each of the codes above is listed below:

PM2/GL2 prom monitor V3.0 October 24, 1984

screen modes = 0x0, default = HIGH

dc HW register set to 0x0

Memory scan:X.XX.....

clearing memory:X.XX.....

MB mem starts at physical page 0x10

Memory mapped logically contiguous, rwx.

Multibus memory at 10000 to 80000

10000 bytes MB i/o space mapped at F70000

180000 bytes of memory, mapped logically contiguous

Switches are: 1E, hostspeed: 19200, bootenv: 3

iris> n bpcd4432.r

booting: olympus:/usr/local/boot/bpcd4432.r

Downloading.....:

BPC Diagnostic: Tue Nov 6 08:53:14 PST 1984

UC4

DC4

Interface 3 (UC4 Multibus Interface)

PM2

Type <space> to abort the current command

[1] -> init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cfr: display ab, update ab, viewport, backline, ldlinestip

mdr:

dcflags: v fsm-clear

significant bitplanes: FFFFFFFF

installed bitplanes: FFFFFFFF

color index: 0

write enable code: FFFFFFFF

[2] -> dclamptest 1

[3] -> dcr 0

dcflags: v fsm-clear

[4] -> dcrtest 1 1

pass 1: errors this pass = 0, total errors = 0

[5] -> maptest 1 1 1F 0

pass 1: 01234, errors this pass = 0, total errors = 0

[6] -> fptest 1 1 1F 0

pass 1: 01234, errors this pass = 0, total errors = 0

[7] -> ldconfig 0f

cfr: display ab, update ab

[8] -> bptest 1 1 3F 0 0

pass 1: 012345, errors this pass = 0, total errors = 0

[9] -> ldconfig C

cfr: update ab

[A] -> bptest 1 1 3F 0 0

pass 1: 012345, errors this pass = 0, total errors = 0

[B] -> init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cfr: display ab, update ab, viewport, backline, ldlinestip

mdr:

dcflags: v fsm-clear

significant bitplanes: FFFFFFFF

installed bitplanes: FFFFFFFF

color index: 0

write enable code: FFFFFFFF

[C] -> dcr 0

dcflags: v fsm-clear

[D] -> ldconfig 0f

cfr: display ab, update ab

[E] -> stripetest 1 1 40 1

pass 1: Cannot test planes FFFFC0C0, A:67 B:67 C:01234567 D:01234567

a0a1a2a3a4a5b0b1b2b3b4b5 badplanes: 0, cumulative: 0

[F] -> ldconfig 9

cf: display a, update b

[10] -> stripetest 1 1 40 0

pass 1: Cannot test planes F0F00000, C:4567 D:4567

a0a1a2a3a4a5a6a7b0b1b2b3b4b5b6b7c0c1c2c3d0d1d2d3 badplanes: 0,  
cumulative: 0

[11] -> ldconfig 0f

cf: display ab, update ab

[12] -> ramptest 1 0 0 0

[13] -> ramptest 0 1 0 0

[14] -> ramptest 0 0 1 0

[15] -> ramptest 1 1 1 0

[16] -> init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cf: display ab, update ab, viewport, backline, ldlinestip

mdr:

pass 1: Cannot test planes F0F00000, C:4567 D:4567

a0a1a2a3a4a5a6a7b0b1b2b3b4b5b6b7c0c1c2c3d0d1d2d3 badplanes: 0,  
cumulative: 0

[2E] -> ldconfig 0f

ctr: display ab, update ab

[2F] -> ramptest 1 0 0 0

[30] -> ramptest 0 1 0 0

[31] -> ramptest 0 0 1 0

[33] -> ramptest 1 1 1 0

[34] -> dcr 3040

dcflags: rgb, pipe4, prom, v fsm-clear

[36] -> ramptest 1 0 0 1

[37] -> ramptest 0 1 0 1

[38] -> ramptest 0 0 1 1

[39] -> ramptest 1 1 1 1

[3A] -> quit

pass 1: 012345, errors this pass = 0, total errors = 0

[9] -> ldconfig C

cfr: update ab

[A] -> bptest 1 1 3F 0 0

pass 1: 012345, errors this pass = 0, total errors = 0

[B] -> init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cfr: display ab, update ab, viewport, backline, ldlinestip

mdr:

dcflags: v fsm-clear

significant bitplanes: FFFFFFFF

installed bitplanes: FFFFFFFF

color index: 0

write enable code: FFFFFFFF

[C] -> dcr 0

dcflags: v fsm-clear

[D] -> ldconfig 0f

cfr: display ab, update ab

[E] -> striptest 1 1 40 1

pass 1: Cannot test planes FFFFC0C0, A:67 B:67 C:01234567 D:01234567

a0a1a2a3a4a5b0b1b2b3b4b5 badplanes: 0, cumulative: 0

[F] -> ldconfig 9

cfr: display a, update b

[10] -> striptest 1 1 40 0

pass 1: Cannot test planes F0F00000, C:4567 D:4567

a0a1a2a3a4a5a6a7b0b1b2b3b4b5b6b7c0c1c2c3d0d1d2d3 badplanes: 0,  
cumulative: 0

[11] -> ldconfig 0f

cfr: display ab, update ab

[12] -> ramptest 1 0 0 0

[13] -> ramptest 0 1 0 0

[14] -> ramptest 0 0 1 0

[15] -> ramptest 1 1 1 0

[16] -> init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cfr: display ab, update ab, viewport, backline, ldlinestip

mdr:



dcflags: v fsm-clear  
significant bitplanes: FFFFFFFF  
installed bitplanes: FFFFFFFF  
color index: 0  
write enable code: FFFFFFFF

[17] -> dcr 40

dcflags: rgb, v fsm-clear

[18] -> ldconfig 0f

cfr: display ab, update ab

[19] -> ramptest 1 0 0 1

[1A] -> ramptest 0 1 0 1

[1B] -> ramptest 0 0 1 1

[1C] -> ramptest 1 1 1 1

[1D] -> init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cfr: display ab, update ab, viewport, backline, ldlinestip

mdr:

dcflags: v fsm-clear

significant bitplanes: FFFFFFFF

installed bitplanes: FFFFFFFF

color index: 0

write enable code: FFFFFFFF

[1E] -> dcr 3000

dcflags: pipe4, prom, v fsm-clear

[1F] -> dcrtest 1 1

pass 1: errors this pass = 0, total errors = 0

[20] -> maptestdcrtest 1 1

pass 1: errors this pass = 0, total errors = 0

[21] -> maptest 1 1 1F 0

pass 1: 01234, errors this pass = 0, total errors = 0

[22] -> fntest 1 1 1F 0

pass 1: 01234, errors this pass = 0, total errors = 0

[23] -> ldconfig 0f

cfr: display ab, update ab

[24] -> bptest 1 1 3F 0 0

pass 1: 012345, errors this pass = 0, total errors = 0

[25] -> ldconfig C

cfr: update ab

[26] -> bptest 1 1 3F 0 0

pass 1: 012345, errors this pass = 0, total errors = 0

[27] -> init 2

configuration [0,FFFF], DF:

mode [0,7], 0:

dc register [0,FFFF], 0:

color index [0,FFF], 0:

write enable code [0,FFFFFFFF], FFFFFFFF:

significant bitplanes [0,FFFFFFFF], FFFFFFFF:

cfr: display ab, update ab, viewport, backline, ldlinestip

mdr:

dcflags: v fsm-clear

significant bitplanes: FFFFFFFF

installed bitplanes: FFFFFFFF

color index: 0

write enable code: FFFFFFFF

[28] -> dcr 3000

dcflags: pipe4, prom, v fsm-clear

[29] -> ldconfig 0f

cfr: display ab, update ab

[2B] -> stripetest 1 1 40 1

pass 1: Cannot test planes FFFFC0C0, A:67 B:67 C:01234567 D:01234567

a0a1a2a3a4a5b0b1b2b3b4b5 badplanes: 0, cumulative: 0

[2C] -> ldconfig 9

cfr: display a, update b

[2D] -> stripetest 1 1 40 0

# Contents

<b>LESSON 8: ADJUSTMENTS</b> .....	<b>1</b>
WHAT COLOR IS THE APPLE .....	<b>2</b>
COLOR EXPRESSION .....	<b>3</b>
TWO RED BALLS .....	<b>4</b>
ACCURATE COLOR EXPRESSION .....	<b>5</b>
COLOR COMPONENTS .....	<b>6</b>
NUMERICAL COLOR EXPRESSION .....	<b>7</b>
FULL WHITE COLOR VALUES .....	<b>8</b>
(x,y) CHROMATICITY DIAGRAM .....	<b>9</b>
MONITOR PIXEL ORGANIZATION .....	<b>10</b>

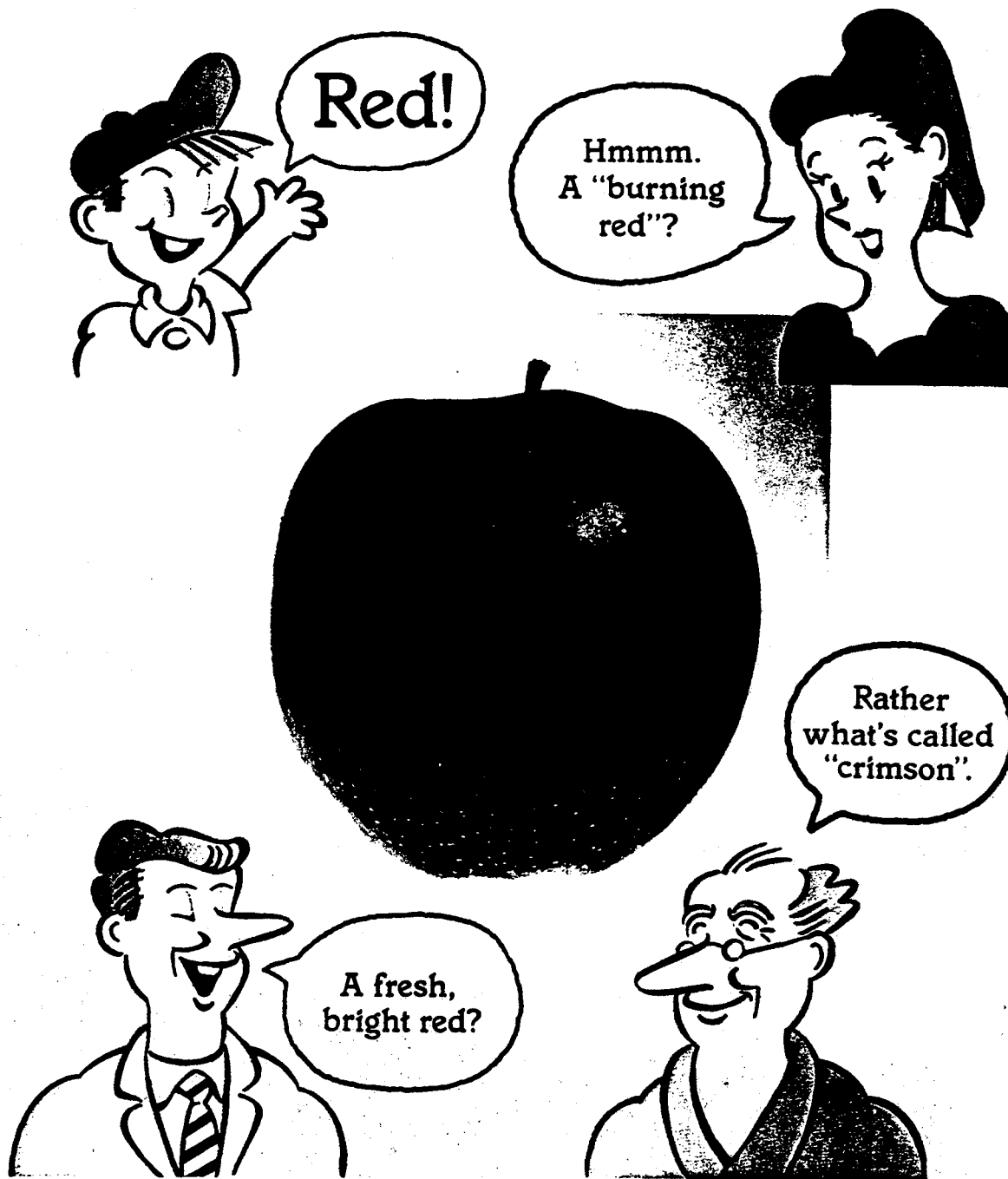


## **LESSON 8: ADJUSTMENTS**

Upon completion of this lesson, the student, using available documentation and test equipment, must correctly perform the required adjustments to an IRIS color monitor and workstation power supply.

The adjustments must be performed per current adjustment procedures and the equipment must be left in normal operational condition once the adjustments have been made.

WHAT COLOR IS THE APPLE



**COLOR EXPRESSION****Color expression often means ten different colors from ten different people.**

**If you show the same apple to four different people, you are bound to get four different answers.**

Color is a matter of perception, of subjective interpretation. To express the same color, different people will draw upon different references and express the exact same color in different words. It is because there are such varied expressions that communicating a particular color is so difficult - and vague. Can we tell someone "the apple was a burning red" and then expect them to be able to exactly duplicate that color? Verbal expression of color is simply too complicated and difficult. But... if there were a standard method by which colors could be accurately expressed, color communication would become much simpler and more straightforward. Precise color communication would, in a word, simply eliminate color troubles.

**Common color names and general color names:****To what extent can words express color?**

Words that express colors have always changed with the times. If we consider, for instance, the red we've been talking about, there's "vermillion", "cinnabar", "crimson", "scarlet". These are called "common color names".

Today a somewhat more precise expression is obtained by using such adjectives as "bright", "dull", and "deep". Terms such as "bright red" used by the man in the picture are called "general color names".

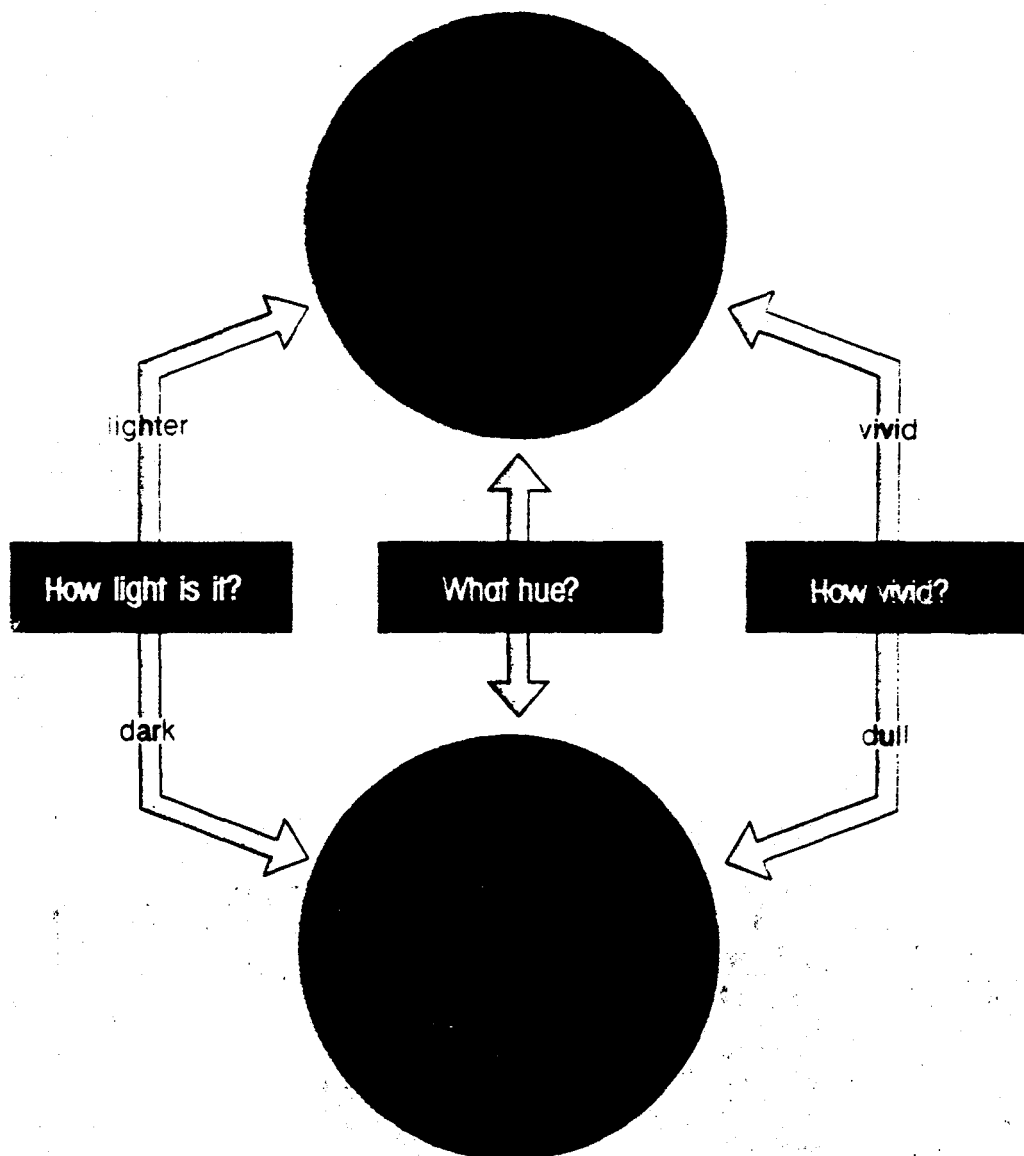
Although there are many such devices used for a slightly better definition, different people will still interpret "crimson" or "bright red" in different ways. This is obviously still not clear.

Well then, how should colors be verbalized without misunderstanding one another?



**TWO RED BALLS**

Two red balls.  
Now tell someone the  
difference between them.



**ACCURATE COLOR EXPRESSION**

**To better understand accurate color expression, let's take a look at color.**

Two red balls. At first glance they look the same, but upon closer examination you know they are different. But in how many ways? The COLOR of both is red, but the upper ball is somewhat BRIGHTER than the lower ball - and of course the lower one is darker. At the same time, the upper appears VIVID - which the bottom ball certainly isn't. When colors are classified, they can be broken down into three primary elements of their *hue, lightness,* and *saturation.* -CLOMA

## COLOR COMPONENTS

Color is "hue",  
lightness is "value",  
saturation is "chroma":  
this is the world of color.

**Hue:** "Red", "yellow", "green", "blue"...

Apples are red, lemons are yellow, the sky is blue - and that's how we all perceive color in everyday language. The term used to distinguish these red, yellow, and blue colors is called *hue*.

**Value:** The Lightness of colors.

Colors can be separated into "light" and "dark" colors when their values are compared. Take for instance the yellow of a lemon and that of a grapefruit. Without a doubt, the lemon yellow is much lighter, wouldn't you say. If so, what about lemon yellow and the red of a sweet cherry? Again, the lemon yellow is lighter, right? This lightness, which can be measured independently of color hue, is *value*.

**Chroma:** Vivid colors, dull colors. (Saturation)

Going back to the example of yellow, how do the colors in a lemon and a banana differ? You might say the lemon is "light", as we did above, but more to the point in this case, it is "vivid" while that of the banana is "dull" - another big difference, but this time one of color intensity, or clarity. This third property of color that is distinguishable from both value and hue is called *chroma*.

---

**NUMERICAL COLOR EXPRESSION**

## Expressing colors more precisely than with words: Numerical color communication.

*Capital Y*

If the color of the apple was expressed numerically, it would be:

$$Y = .287 \quad x = 0.460 \quad y = 0.287$$

*Value*

*↑ Hue*

*↑ Chroma*

**Y = Value:** Measured in ft. lamberts.

**x and y = hue and chroma**

By using a numerical system, no matter what we may say the color is verbally, the expressed values will always define the same color.

Also, by assigning colors numbers, test equipment can be used to monitor colors and tell us exactly what color is occurring.

**FULL WHITE COLOR VALUES**

Table 1 lists the values for "x" and "y" for the color white. They are industry standards and insure that the color white will be the same from manufacture to manufacture of color generating or color monitoring devices.

*The way SGI sets it up  
with the analyzer*

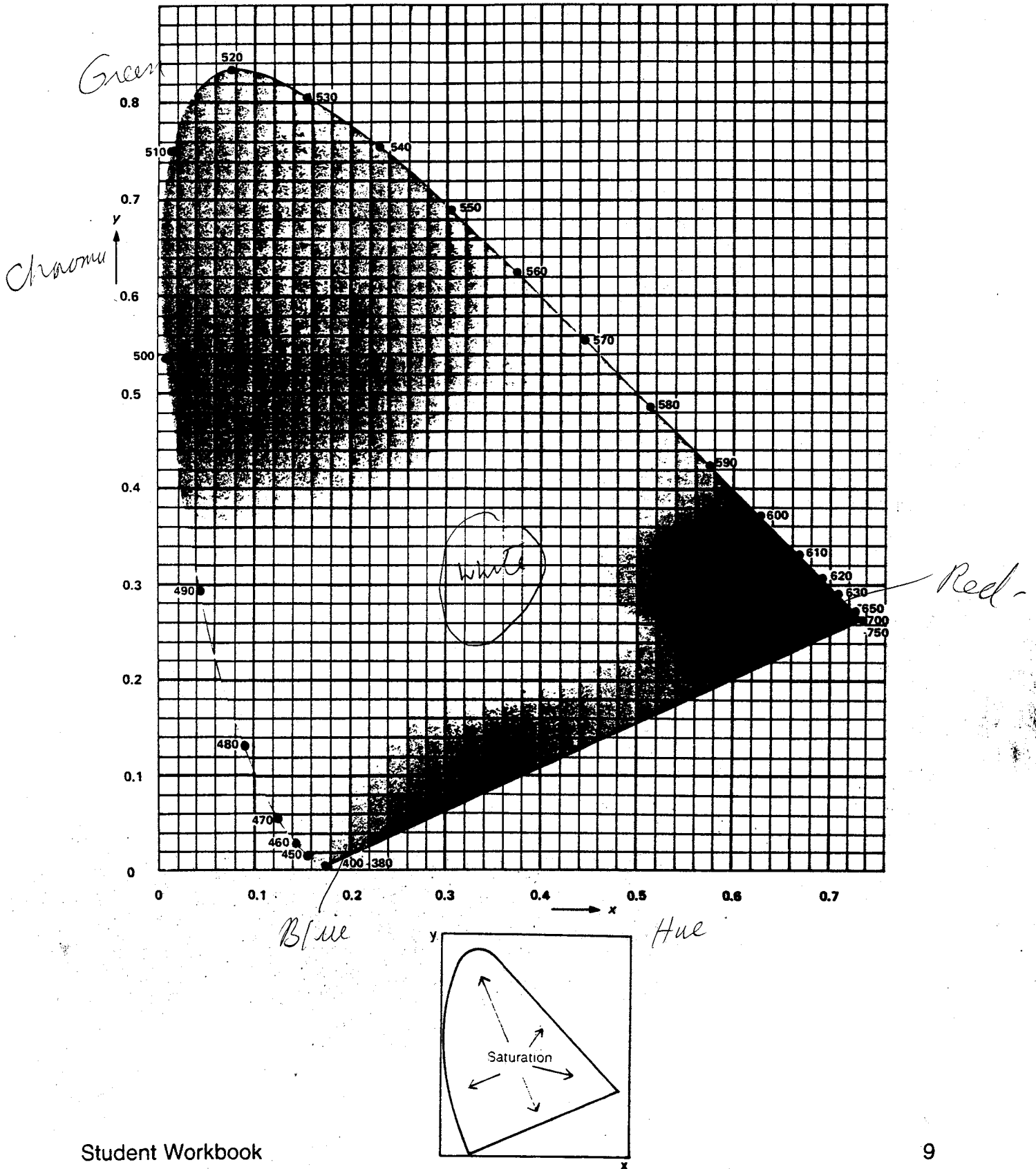
**TABLE 1**

Color Channel	Reference Data		
	x	y	Y
W	313	329	25.0
1	313	329	25.0
2*	0	0	0
3	281	311	10.0

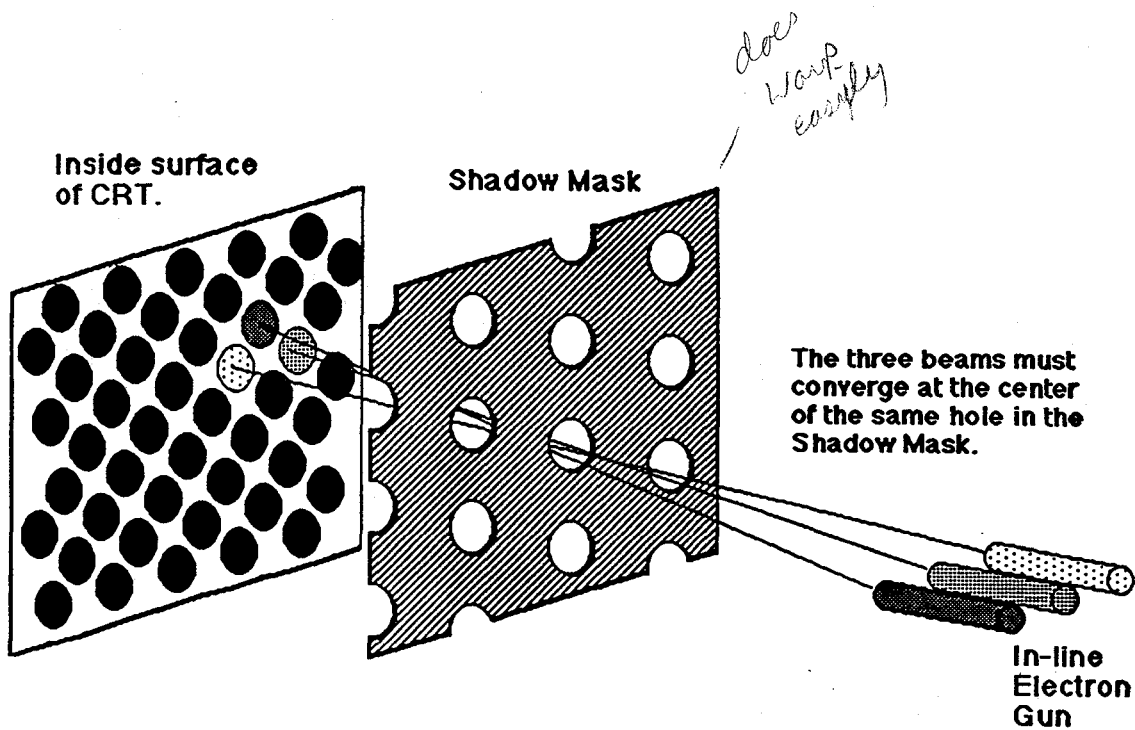
*Color*

*White*

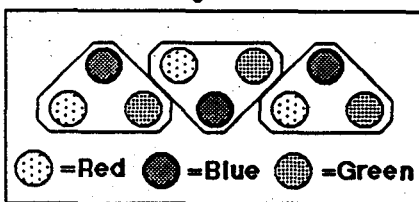
(x,y) CHROMATICITY DIAGRAM



MONITOR PIXEL ORGANIZATION



Pixel Organization



# Contents

<b>LAB 8.1: Adjustments</b> .....	1
I. OBJECTIVE .....	2
II. PURPOSE .....	2
III. EQUIPMENT. ....	2
IV. PROCEDURE .....	3
SECTION 1: The Minolta Color Analyzer .....	4
SECTION 2: Adjustment Procedure of HM-4619-C Monitor .....	12
SECTION 3: HM-4619-C 19 Inch Monitor Convergence Adjustment .....	21
SECTION 4: Adjustment Procedure of CM-2073-A Monitor .....	25
SECTION 5: Power Supply Adjustments .....	31
SECTION 6: Diagrams .....	32
<b>DIAGRAM 1: Minolta at Power On</b> .....	33
<b>DIAGRAM 2: Minolta - Depressing O-CAL</b> .....	34
<b>DIAGRAM 3: Minolta Channel Data</b> .....	35
<b>DIAGRAM 4: Minolta - Channel Pair W-1 with MR depressed</b> .....	36
<b>DIAGRAM 5: Patterns Help Menu</b> .....	37
<b>DIAGRAM 6: Monitor Rear View without cover</b> .....	38
<b>DIAGRAM 7: Screen Position and Sync Adjustment Pots</b> .....	39
<b>DIAGRAM 8: Monitor Side View - Adjustment Pots</b> .....	40
<b>DIAGRAM 9: Display Centering</b> .....	41
<b>DIAGRAM 10: Monitor Rear View - Color Adjustment Pots</b> .....	42
<b>DIAGRAM 11: Monitor Top View - Color Adjustment Pots</b> .....	43
<b>DIAGRAM 12: Convergence</b> .....	44
<b>DIAGRAM 13: Convergence Test Pattern</b> .....	45
<b>DIAGRAM 14: Connecting Convergence Tester</b> .....	46
<b>DIAGRAM 15: Convergence Tester Panel</b> .....	47
<b>DIAGRAM 16: Test Patterns</b> .....	48
<b>DIAGRAM 17: Convergence Examples</b> .....	49
<b>DIAGRAM 18: Hitachi CM2073A Color Adjustment Pots</b> .....	50
<b>DIAGRAM 19: CM2073A Background Adjustment Pots</b> .....	51
SECTION 7: Review .....	52





## LAB 8.1: Adjustments

This lab project contains seven sections:

- **Section 1: The Minolta Color Analyzer**

This section is used to teach you how to use and program the Minolta TV-2150 color analyzer. It also gives the student information concerning the adjustments that are made using the analyzer.

*What we need*

- **Section 2: Adjustment procedure for HM-4619-C 19" color monitor**

This section is an instructional procedure for the color adjustments, sync, raster, and picture centering adjustments.

- **Section 3: Convergence adjustment procedure for HM-4619-C 19" monitor**

This section is an instructional procedure for making the convergence adjustment on the HM-4619-C 19" color monitor.

- **Section 4: Adjustment procedure for CM-2073-A 19" color monitor**

This section is an instructional procedure for color adjustments, sync, raster, and picture centering adjustments.

*The one we have  
Tilt + Swivel*

- **Section 5: Power supply adjustments**

This section gives information concerning primary power supply adjustments and adjustments made to primary and add-on supply when the second supply is added to a 2400 Turbo workstation.

- **Section 6: Diagrams**

This section contains all the diagrams and tables referred to throughout this lab project.

- **Section 7: Questionnaire**

This is a questionnaire that you will complete and hand in after performing this lab project.

## I. OBJECTIVE

When you finish this lab project you should be able to:

- A. Correctly make color, sync, and raster/picture size adjustments to either 19" color monitor using recommended procedures. The monitor shall be left in good operational condition and with all adjustments in spec.
- B. Correctly make the convergence adjustment to HM-4619-C 19" color monitor using the recommended procedures. The monitor shall be left in good operational condition with the convergence adjustment in spec.
- C. Given the three values of color reference (x, y, and Y), program the channel pair of W-1 of the Minolta color analyzer with the supplied values.

## II. PURPOSE

Teach the student how to make color and convergence adjustments to the "old" style 19" color monitor, and color adjustments to the "new" style color monitor.

The project also attempts to help the student understand what is being done when the adjustments are being performed.

## III. EQUIPMENT.

- A. A functional IRIS 3030 or 3130 workstation.
- B. Classroom tool kit.
- C. HM-4619-C 19" color monitor (old style).
- D. CM-2073-A 19" color monitor (new style).
- E. Minolta TV-2150 color analyzer.
- F. Hitachi DDC Console Type II convergence tester.
- G. Small magnifying glass (this is optional).

The values define a color somewhere in the *white* range of colors and is an industry accepted standard for white. The *Y* value ensures we drive our monitors to a maximum of 25 ft. lamberts.

**NOTE:**

The monitors could be set for a higher *ft. lambert* drive, but damage to the *shadow mask*, in the form of permanent warpage, caused by the excess heat resulting from over driving the monitor, could result. Once the shadow mask becomes warped, screen distortions will occur.

With a defined *standard* color and amount of drive stored inside the Minolta, the Minolta's micro computer will insure that every monitor adjusted will be the same. This is accomplished by sampling color (through a light probe) from the monitor, comparing that color against the stored standards, and indicating to the user when the color of the monitor matches the stored standards. The user will adjust specific color gains (while the Minolta is monitoring the color) to achieve the desired color.

The Minolta stores another piece of required data; the characteristics of the phosphor of the individual monitors. This information is initially programmed into the Minolta and is held in it's memory along with the standard color values and maintained by a nickel-cadmium battery that will hold a charge for about two months.

**NOTE:**

Once you program your Minolta to hold your specific color values and phosphor characteristics, you should ensure that it is powered on several times a week to maintain the battery charge. If you do not maintain the charge, you will always be re-programming the Minolta when you want to use it.

We will now check our Minolta and verify that it is programmed for the type of monitor we will adjust, and if not, program the correct data into it's memory.

**STEP 1: Power on the Minolta.**

When the Minolta powers on, the word *CAP* should be displayed in the top row of digital lights. See figure #1.

Also, check the switch labeled *luminance*, located underneath the Minolta, to be set in the *ft-L* position.

*on the bottom*

**STEP 2: Calibrate the Minolta.**

By displaying the word *CAP*, the Minolta is asking you to place a dark lens cap over the probe or to place the probe into the dark cup located on top of the tester. This is designed to block light from entering the probe while the self-calibration process occurs.

Now, place the probe into the cup on top of the tester and depress the **O-CAL** button. The word *CAL* will now be displayed while the unit undergoes a self-calibration. (See figure #2)

When the displays start blinking, self-calibration is complete and the probe can be removed from the cup.

**NOTE:**

You will notice that the probe probably has small pieces of electrical tape around the monitoring end. These pieces of tape will keep the probe in place on the CRT screen leaving your hands free to make the required adjustments. As the tape looses it's ability to stick, you will have to replace it with new tape.

**STEP 3: Verify if correct reference values are stored in Minolta.**

The Minolta has the ability to store four sets of color reference data ( *x*, *y*, and *Y* ) and the phosphor characteristics of four monitors.

The color reference data is stored in the four channels *W*, 1, 2, and 3. Using the button *COLOR*, you can sequentially step the channel indicator to allow you access to the stored color reference data. The LED's under the letter *W*, and under the numbers 1-3 will inform you as to the channel you have selected.

The CRT phosphor characteristic data is stored in the channels numbered 1-4 and controlled by the button *CRT*. Again, the LED's under the numbers will inform you as to what CRT channel you have selected.

When you adjust a monitor, you will have one *COLOR* channel and one *CRT* channel selected.

The following convention of storing reference data into the Minolta storage channels is used at Silicon Graphics manufacturing facilities. Since a limited number of Minolta testers exist, and they are shipped around the country, originating at SGI, Mt. View, your tester will most likely be programmed in the following manner:

**See diagram #3**

LAB 8.1: Adjustments

---

We shall now check the reference data.

Since we will be adjusting only 19" monitors in class, we want to check the *W color channel*, and the *1 CRT channel*.

Use the *COLOR* and *CRT* buttons to step to these channels.

Now, set the toggle switch on the left side of the Minolta to the *CHROMA* position. (This will allow you to read the data stored in the color channel memory)

Now, depress and hold the *MR* (memory read) button. The values displayed should be equal to the data shown in Table 1 of diagram 3, or diagram 4. (Release *MR* when you are satisfied)

If the numbers are correct, switch the toggle *mode* switch on the left side of the tester to the *ANALYZER* position.

Now, depress and hold the *MR* button again. The value **100** should be displayed in each row of display lights.

These values are percentages that were established by the Minolta when this channel was programmed with the color reference data. What they indicate is the percentage to which the color that you will monitor and adjust, will be compared against the stored reference color data. In other words, using the Minolta with the pre-programmed data will allow you to adjust the monitor color to 100% of the reference color.

**NOTE:**

The Minolta will only show you the percentage readings when channel pair *W-1* is selected. The other channels may be programmed, but the percentage data cannot be displayed for those channels.

Also, in order to use other channel pairs other than *W-1*, the *W-1* channel pair must be programmed with valid reference data. So, if you have to program the Minolta, and you only have one type of monitor, you must select and use the *W-1* channel pair.

After displaying the programmed values and they are what you require, then move to the section in the lab project that presents the procedure for the monitor type you are going to adjust, otherwise, continue with this section to learn how to program the Minolta with color reference data.

**STEP 4: Programming the Minolta.**

If you need to program the Minolta in the field, remember, the *W-1* pair must be programmed if you want to program all the channel pairs, otherwise, just program *W-1*.

Also, the *W-lock* switch, found underneath the Minolta, must be in the **off** position to allow new values to be programmed into the Minolta's memory. Do not forget to return the switch to the **on** position when programming is complete.

You must first create the required reference values for the monitor that you will perform your adjustments on. The way you will create these values may seem a little odd, but it is quick and easy and it works.

You are going to use the monitor (that you are about to adjust) to help you produce the reference values.

- Login to the system and cd to */usr/diag*.
- Perform an *ls* command to verify that you have a program called *colors*.
- Execute *colors*.

You will get a black screen with eight columns of data at the bottom of the screen. The information in each column will look like the following example:

```
0
R  0
G  0
B  0
```

Each column will have a different number, but the data and what happens within the column is the same. The column runs from screen top to screen bottom where the above information is displayed.

Using the mouse, position the arrow cursor into one on the middle columns, place the arrow in the black area above the displayed information.

Now, depress and hold the right mouse button. Notice that a shade of blue fills the column. By holding the right mouse button depressed, the intensity of blue can be varied by moving the cursor up and down in the column. Also, notice that the program gives you the digital value of the color, 255 being the maximum value.

## LAB 8.1: Adjustments

Create the maximum intensity of blue and then, using the middle mouse button, create the maximum intensity of green. Notice, as you add in the green color, you produce other shades of color. With a maximum of blue and green, you produce *cyan*.

Now, using the left mouse button, create the maximum intensity red. This will produce full white.

Once you have created full white in the selected column, move the cursor to the adjacent column and repeat the procedure of producing white.

**NOTE:**

If you need help with the program, depress the **h** key. This will display the help message. Ignore the commands *c*, *f*, *F*, and *p*.

The message tells you how to terminate the program when you are finished; move the cursor to the lower left corner and depress all three mouse buttons.

You now have a field of *full white* to work with.

- Place the Minolta probe into the cup on top of the tester.
- Depress the *O-CAL* button.

After a few seconds random numbers will appear at the digital read outs which will also begin to blink.

- Attach the probe to the CRT in the center of the full white field that you created.
- Set the Minolta to *CHROMA* mode.
- Using the *COLOR* and *CRT* buttons, select channel pair *W-1*.
- Now, the object is to adjust the displayed white field (using the mouse) so that the Minolta *x*, *y*, and *Y* displays are equal to the values shown in the table on diagram 3 for the monitor type you are going to adjust.

The *x* and *y* values are chromaticity coordinates, and together define a particular color, in this case white. These numbers can be changed by adjusting the relative amounts of red, green, and blue intensities. Although, varying any one color can affect both *x* and *y*, the *x* value will track red changes, and the value *y* will track blue level changes.

You will notice that slight movements of the mouse make great changes in the values. Your body presence around the CRT will also slightly affect the values, but patience will



be rewarded with the desired reference values.

The desired value of  $Y$  is obtained by using the monitor front panel brightness control.

**NOTE:**

If you cannot get enough drive from the monitor to achieve the desired value for  $Y$ , place your color cable terminators into the *high* position for the three colors. This will over drive the monitor, but should give the required drive to complete the adjustments.

Also, if the desired values cannot be achieved using the above method, then the drive adjustments of the monitor will have to be turned up. But, since you were going to adjust the monitor anyway, go ahead. This should be done as a last resort.

Once you create the proper color on your monitor that gives the desired reference values, depress the *HOLD* button. This will lock in the values you just created. Note: The value may drift as you depress hold, if this is the case, depress hold again to remove the lock and try again.

- With the *hold* light on, and the correct values displayed, depress and hold the *MS* (memory store) button. While *MS* is depressed, push and release the *COLOR* button. After releasing the *color* button, release the *MS* button. The *color* and *CRT* led's will momentarily blink. The data is now stored and the *hold* button can now be toggled to turn off the led.

**NOTE:**

If you had to set the termination switches to high, at this time return them to the normal position.

- The next step will be to store the CRT phosphor characteristics into the selected *CRT* channel, which is 1.
  - Now, display a full red field. If you cannot do this under program control, simply disconnect the green and blue cables from the monitor.

You should be able to use *colors* to create a full red field. The above method of removing the cables is just an alternative.

At this point, set the front panel *brightness* control fully clockwise for a red field of maximum intensity. (NOTE: The brighter the red, the greater the accuracy of the reading, but only adjust the brightness control to a point where the background raster begins to show, then back off slightly)

Now, using the *CAL* button, select the *R* led. With the red field displayed and the *R* led selected to the left of the *CAL* button, depress and hold the *MS* button. With the *MS* button depressed, depress and release the *CAL* button. After releasing the *CAL* button, release the *MS* button. The *R* led should blink while the *CAL* button is depressed.

- Now, in the same manner that you created the red field, create a green field. Use the *CAL* button to select the *G* led. With a green field displayed and the *G* led selected, depress and hold *MS*. With *MS* depressed, depress and release the *CAL* button. After releasing the *CAL* button, release the *MS* button. The *G* led should have blinked while *CAL* was depressed.
- Now, in the same manner as before, create a blue field. Use the *CAL* button to select the *B* led. With a blue field displayed, depress and hold the *MS* button. With *MS* depressed, depress and release the *CAL* button. After releasing *CAL*, release *MS*. All the lights should have blinked.

The analyzer is now programmed. The *x*, *y* and *Y* values you programmed could be displayed by depressing the *MR* button with the Minolta in the ~~MINOLTA~~ <sup>analyzer</sup> mode, and channel *W-1* selected. In the analyzer mode with the *MR* button depressed, the RED, BLUE, and GREEN values displayed should equal 100, 100, 100.

After going through this programming procedure, you should see how important it is to keep a good charge on you Minolta battery.

You are now ready to adjust the monitor.

## SECTION 2: Adjustment Procedure of HM-4619-C Monitor

If your customer is complaining about monitor color quality, before you attempt any adjustments, verify that the signal cables are terminated properly.

SGI uses *current drive* ( D to A ) drivers. Therefore, the length of our cable will not effect the signal. The cable termination must be **75 ohms**.

Both ends of the signal line must be terminated with **75 ohm** terminators. If several monitors are sharing the same line, then only the last monitor on the line will supply the termination. The other monitors on the line will be set for *high* termination.

The terminators not being set correctly will result in too great a drive for the line not terminated correctly.

To make a point of how important correct termination is, perform the following before continuing with the lab project:

- Login to your workstation.
- Cd to */usr/diag*.
- Execute the program called *colors*.
- Using the mouse, create maximum intensity colors of RED, BLUE, and GREEN in several of the screen columns.
- With the colors displayed, switch the signal line terminators to *high* and watch how the colors are affected.
- Place the terminators into the correct position when finished with this little demonstration.

Now, on with the monitor adjustments....

### NOTE:

Before attempting any adjustment of the color monitor, it is necessary to insure that it has been powered on for a *minimum* continuous period of *at least 30 minutes*.

**STEP 1: Display a full white field.**

In order to perform the color adjustment, you need a *full white* field displayed.

To get a full white field, perform the following:

- Cd to */usr/diag*.
- Execute the program called *patterns*.

When *patterns* begins, the screen is cleared and you will see an *arrow cursor* somewhere in the center of the dark screen.

Depressing the left mouse button (hold it depressed) will display a menu that lists the various test patterns available by this program. See **diagram #5**

Move the cursor to *100% white field* and release the left button. A full white field will now be displayed. To select another pattern or quit the program, depress and hold the left mouse button while moving the cursor to the desired pattern or quit function, again releasing the button to make your selection.

**NOTE**

A file called *README.diag* exists at */usr/diag*. This file is a document that gives a brief explanation of each pattern and how the pattern might be used. Not all the patterns are for use by Field Engineers.

**STEP 2: Degauss CRT.**

Immediately prior to beginning the adjustment procedure, hold the monitor degauss button in for a period of **10 seconds**. The degauss button is located on the front panel of the monitor.

If after using the front panel degauss function, any local impurities persist, then **degauss the CRT** using an external degaussing coil.

**NOTE:**

As you can see from the above statement, you may need the degaussing coil. **This must be obtained from the F.E. parts distribution center, therefore, it would be wise to plan ahead and request the coil be sent to you before you schedule time with your customer.**

**If your region does not have the Minolta Analyzer, then at the time you request the analyzer be sent, also request the coil.**

- If you need to degauss the CRT using the external coil, follow these instructions:
  - Turn on the hand-held degaussing coil at a distance of *3 feet* from the CRT screen.
  - Move it to within about *4 inches* of the top and side of the screen. Make a slow circular movement a few times around the screen.
  - Move the coil gradually away from the screen and turn off the coil. **DO NOT turn off the coil within 3 feet of the screen.**

If distortions cannot be removed by degaussing, then the monitor must be returned to SGI manufacturing.

### STEP 3: Remove monitor cover.

To remove the protective cover from the monitor chassis, locate the two screws on each side of the monitor (at the bottom of the cover) and remove.

Stand behind the monitor and lift the cover up and away from the chassis. **DO NOT** move the cover from side-to-side as you back it off the chassis. You could accidentally touch the high voltage fly-back transformer if you do so.

### CAUTION:

It is imperative that all of the following adjustments are made with a **NON-METALLIC, NON-CONDUCTIVE** screwdriver! It is very easy to cause permanent damage with a metal screwdriver.

### STEP 4: Synchronization Adjustment.

If your monitor *Horizontal* and *Vertical* sweeps are in sync., then skip this step and goto **STEP 5**, else follow the instructions:

- First, you must select a *cross-hatch* screen pattern. Using the mouse, depress the left button and select the *convergence* pattern.
- Next, locate the position of the adjustment pots using diagrams 6, 7, and 8. The pots you may use are: *VR1, VR4, and VR404*.

## LAB 8.1: Adjustments

Diagram #6 shows a rear view of the monitor.

Diagram #7 shows the rear view with the protective plate, that covers the logic board, removed. The position of the pot (on this board) that you may need access to ( *VR404* ) is shown by this diagram.

Diagram #8 shows the side view of the monitor where the deflection PCB is located. This drawing shows the location of *VR1* and *VR4*.

- Adjust *VR1* ( Horizontal Hold ) gradually so that the vertical lines of the cross-hatch pattern become stable.
- Adjust *VR4* ( Vertical Hold ) gradually so that the pattern of cross-hatch becomes stable.
- Adjust *VR1* ( Horizontal Hold ) gradually again to hold the synchronization in the middle of the range.

**NOTE:**

When the pattern is not stable in spite of the adjustments as per the above three steps, adjust *VR404* ( Sync Level ) gradually so that the picture becomes stable.

- When you are satisfied with the monitor sync, use the mouse to again select the full white field ( 100% white field ).

**STEP 5: Adjustment of Horizontal Screen Location.**

The next series of adjustments will insure that the center of the *raster field* is centered within the CRT screen and the *picture area* is centered in the *raster field*. Follow these instructions:

- First, using figures 7 and 8, locate the adjustment pots that you may require access to, and they are: *VR8*, *VR403* and *VR1*.
- Set *VR403* ( Horizontal Phase ) to it's minimum value.

**NOTE:**

The adjustment pots react as follows:

Counter-clockwise to *minimum* value.

Clockwise to *maximum* value.

The boundaries of the *raster field* are shown by bright vertical lines on left and right

sides.

- Adjust *VR8* ( Horizontal Center ) so that the center of the raster is centered within the CRT screen. ( See example 1 of diagram #9 )
- Now, again adjust *VR403* ( Horizontal Phase ) from minimum toward maximum value gradually so that the *picture* area aligns with the center of the *raster field*. ( See example 2 of diagram #9 )
- When the center of the picture cannot be set by the method in the above steps, the picture is being erased by blanking. *VR1* ( Horizontal Hold ) is not to be set in the center of it's range which holds synchronization. In this case, carefully adjust *VR1* so that the picture may be centered. ( The adjustment of *VR1* is covered in STEP 4 of this lab project. You may have skipped over STEP 4 because your monitor did not require Horizontal or Vertical sync adjustments.)

#### STEP 6: Adjustment of Vertical Screen Location.

This shows you how to vertically center your picture area.

- First, locate the adjustment pot that controls vertical picture location. The pot is *VR12*. ( See diagram #8 for location)
- Adjust *VR12* ( Vertical Center ) so that the picture center is aligned with the center of the CRT screen.

#### STEP 7: Adjustment of Picture Size.

The size of the picture field must meet the spec's defined by example 3 of diagram 9. The following adjustment pots are used to bring the picture size into spec.: *VR2*, *VR3*, *VR9*, and *VR10*.

- Using the mouse, select the "cross-hatch" *convergence* pattern.
- Using diagram #8, locate the adjustment pots.
- Adjust *VR3* ( Width ) so that the horizontal picture width is 14.17 inches.

- Adjust *VR9* ( Height ) so that the vertical picture height is **10.63 inches**.
- Adjust *VR2* ( SPCC ) so that the vertical lines of both right and left sides are straight.
- Adjust *VR10* ( Vertical Linearity ) so that the vertical spans of "cross-hatches" are equidistant. **NOTE: When linearity is adjusted, adjustment of picture height *VR9* may again be required due to changes of the picture size.**
- Using the mouse, select the full white field ( 100% white field ).

Now that all the preliminary adjustments have been checked or performed, you are ready to make the color adjustments.

#### STEP 8: Color Adjustments.

Remember, you should have a *full white field* displayed to make these adjustments.

#### NOTE:

As you make the following adjustments, you should keep in mind that the monitors *time-out* circuit will eventually shut off the video if no activity occurs at the keyboard. If, while during the procedure, you suddenly lose your Minolta readings, the cause is probably time-out. Just depress any key or move the mouse to get the video back.

- First, locate the positions of the 10 pots you will use during the procedure. The pots are: *VR101, VR201, VR301, VR103, VR203, VR303, G-Background, R-Background, B-Background, and G2 Bias*.

Use diagrams 10 and 11 to locate the pots.

- Set the following pots to their *fully counterclockwise* positions:
  - *G-BACKGROUND*
  - *R-BACKGROUND*
  - *B-BACKGROUND*
  - *R-LEVEL - VR101*
  - *G-LEVEL - VR201*



- **B-LEVEL - VR301**
- Set the following pots to their *fully clockwise* positions:
  - **R-BACK GAIN - VR103**
  - **G-BACK GAIN - VR203**
  - **B-BACK GAIN - VR303**
- Set the front panel *Brightness control* to its *full clockwise* position.
- Affix the Minolta analyzer probe to the center of the screen.

**NOTE:**

If you have bright light that shines or reflects onto your screen, cover the screen with a jacket or some other type of cover, or remove the source of the light from the screen. Very bright light could be picked up by the probe and corrupt the measurements.

Covering the screen no matter what amount of light is present will never hurt.

- Set the Minolta analyzer to *ANALYZER* mode.

At this point, separate red, green, and blue values should be displayed on the analyzer. (It is assumed here, that you verified the operation and programming of the Minolta, using section 1 of this lab project, before you began the color adjustments)

- **Sub-step A**

Adjust the monitor *G2 bias* pot so that the *lowest* of the three (3) Minolta readings is between **.5 and 1.0**.

- **Sub-step B**

Now, you are going to adjust two (2) of the three (3) color *BACKGROUND* pots. (see diagram 11)

The color that is producing the *highest* reading on the Minolta at this time will not be adjusted. Remember, all three of the pots were initially set to the *full counterclockwise* position.

What you must now do, is adjust the *BACKGROUND* pots for the other two colors. As you adjust an individual color, notice that the readings for the other two colors are also affected. You must bring up the readings of the two *lower* colors to be equal to

the level of the color that was producing the *highest* initial reading. In other words, produce three equal reading by adjusting the two colors with the *lower* initial readings to a level the third color seeks as you bring up the other two. (equal in the above instruction means very close)

- **Sub-step C**

**QUESTION:**

After performing *sub-step B*, are all 3 readings in the range of **.5 to 1.0** ? (If this is the first pass, the answer will most likely be no)

If the answer to the above question is **NO**, jump back to *sub-step A* and perform it again; adjusting the readings down from the levels obtained in *sub-step B*. After performing *sub-step A*, perform *sub-step B* again.

You will remain in this loop until the answer to the question is yes.

If the answer to the above question is **YES**, then goto the next step in the sequence. From this point on, the *BACKGROUND* adjustments are set and you will not have to touch these pots again.

- When you come out of the above loop with all the readings somewhere in the range of **.5 - 1.0**, you are ready for this step.

Now, adjust the *G2 bias* pot so that the *total* of all three (3) readings is equal to **.3**. This can be achieved with individual values of **.1, .1, .1**, or **0, .1, .2**, etc....

**NOTE:**

When the analyzer readings go below **.5**, the readings will start blinking, indicating it is out of range. Do not become alarmed. Simply read the values as they flash, and adjust *G2 bias* so that the total of the three readings is equal to **.3 MOST** of the time. Do not expect stable readings.

Once you achieve the **.3** level, *G2 bias* is set and can be left alone.

- Next, adjust the **RED, GREEN, and BLUE LEVEL** pots ( *VR101, VR201, and VR301* ) for three equal readings of **100, 100, and 100**. ( 99 - 101 is acceptable )

The **RED, GREEN, and BLUE** levels are now set and the pots can be left alone.

- Now, with the three readings at **100**, using the *Front Panel Brightness* control, drop the readings so that the *lowest* of the three drops to between **.5 - 1.0**.

Now, adjust two (2) of the three (3) **RED, GREEN, or BLUE Back Gain**

pots ( *VR103*, *VR203*, and *VR303* ) so that all three readings are equal to the color that displayed the initial *lowest* reading when the *brightness* control was lowered.

**NOTE:**

All three *back gain* pots were initially set to their full clockwise positions, therefore, one of these pots should remain at this setting: Whichever exhibited the lowest initial reading will not require adjustment. Again, as you move the pots, you will notice considerable interaction of the readings.

When you balance the three readings by moving two of the three pots, you will probably find it necessary to again adjust the *front panel brightness* control down to maintain a minimum reading of between .5 - 1.0 for the color that displayed the initial *lowest* reading.

If the above procedure sounds familiar, it is. Its the same approach that was used earlier when you set the BACKGROUND adjustments using the G2 bias.

Once you get all three readings in the .5 - 1.0 range, monitor adjustment is complete, and the following conditions should be met:

- The front panel brightness control should completely blank the display before reaching the mechanical end of travel.
- Maximum front panel brightness should not overdrive the display.

A full white field should read between 24 - 28 ft. lamberts with front panel brightness fully clockwise.

To measure this, switch the MINOLTA to *CHROMA* mode, and the *bottom* of the three displays will be an intensity reading in *ft. lamberts*.

- There should be NO visible color shift as the front panel brightness is adjusted through its entire range.

If the monitor cannot meet specification, and the customer is complaining, then it must be returned to SGI manufacturing.

### SECTION 3: HM-4619-C 19 Inch Monitor Convergence Adjustment

The convergence adjustment is much easier than the color adjustments, in that, no mechanical or electrical adjustments are made to the monitor.

Information or graphic shapes are produced on the CRT screen by exciting spots of phosphor with an electronic beam. When the beam strikes one of these phosphor spots, light is emitted by the phosphor, which in turn is picked up by our eyes.

The phosphor spots are arranged in a recurring pattern on the inside portion of the CRT screen. There are hundreds of thousands of these phosphor spots covering the inside screen surface.

There are three types of light emitting phosphor spots: Red, Green, and Blue. Spots of each color type are grouped into a tight formation called a *PIXEL*.

The monitor actually has three beam guns that it uses to excite the phosphor spots. Each gun is assigned to excite only one type of spot, therefore, we refer to the guns as the Red gun, Green gun, or Blue gun.

Located between the guns on the inside surface of the CRT is a device called the *Shadow Mask*. The shadow mask is located very near the inside surface. The shadow mask is full of tiny holes, through which, the beams will pass. Actually, there is a one-to-one correspondence between the number of holes and the number of pixels. The mask is positioned so that each hole is directly in front of its corresponding pixel.

The spots, mask, and guns are mechanically aligned, but the beams can be controlled to some extent electronically. Beam deflection can be controlled by logic that contains stored digital information that will cause the beam to line up with the individual mask holes. The information is stored on PROMs and it can be changed to correct for minor mis-alignment of the guns, mask, and screen.

This is what the *convergence* adjustment is all about. Using the external tester, you can program new beam deflection data into the monitor's ram memory to correct for mis-alignment of the beams. ( See diagram 12 for illustration of above explanation )

All three beams must *converge* on the same shadow mask hole.

**NOTE:**

**DO NOT** set your monitor to drive more than *25 ft. lamberts*. The beams will cause excess heating of the shadow mask, which could become permanently warped. Once the mask is warped, distortions in the display will occur.

**QUESTION:**

How do you tell if the convergence is out of spec at an area or areas of the screen?

**ANSWER:**

Start the program *patterns* (located at */usr/diag*) and select the convergence pattern. ( See diagram 13)

If some of the lines of the test pattern have Red, Blue, or Green halos, and it is very noticeable, then the convergence is out of spec. You may only be out of spec in specific areas of the screen, this is because the screen is divided up into small areas, with each area having it's own memory ram containing deflection data for that area.

If your monitor is out of spec, then the following procedure will guide you through the adjustment.

**STEP 1: Attaching the Hitachi Tester.**

Remove the monitor's protective cover and connect the tester to the test port on the back panel of the monitor. ( See diagram 14 for location of test connector, and diagram 15 for illustration of tester control panel )

**STEP 2: Set the tester power switch to ON.****STEP 3: Depress the ON/OFF button.****STEP 4: Remove the Green signal cable from the rear of the monitor.**

As you do this, notice that the color of your monitor display turns *magenta*. This is the compliment of green, produced by mixing Red and Blue.

**STEP 5: Selecting your test pattern.**

Depending on how out of spec your monitor is, select one of the test patterns shown by diagram 16.

Remember, the display may only be out of spec in specific areas of the screen, select the pattern that gives some test spots in these areas.

To select a pattern, depress one of the *zoning* buttons: *3x3*, *5x5*, or *7x17*.

**STEP 6: Depress the button *center*.**

This causes the cursor to be positioned at the center dot and start it flashing.

If you do not like the flashing, depress *flicker*. This stops the flashing. The *flicker* function can be toggled on and off.

**NOTE:**

The flashing is for your benefit. It marks the dot that you are going to work on.

**STEP 7: Position cursor to area out of spec.**

Using the *cursor keys*, position the cursor over the first dot you want to adjust.

**STEP 8: Adjustment of Red and Blue guns.**

Using the *DATA R-B* keys, adjust the Red and Blue guns to converge on the same spot.

You want to produce a spot that is as *white looking* as possible. Remember, the color is not white, but magenta, due to the absence of green, but it will appear white to your naked eye.

As you move the guns (using the R-B keys) you will notice distinct Red and Blue spots grow out of the white spot. ( Your spots will initially be this way, or why are you here)

You want to *converge* the Blue and Red spots into one.

Again, your naked eye will be able to perceive this convergence or lack of it, but a magnifying glass will give you a much better look at what's going on. ( See diagram 17 for an illustration of Red and Blue spots in and out of convergence.)

Once you are satisfied with this spot, move the cursor to the next spot and repeat the process. Continue this until all out of spec areas have been adjusted.

**STEP 9: Adjustment of the Green gun.**

Now, reconnect the Green signal cable.

Using the **DATA FB-G** keys, adjust the green at all the required spots. Remember, you want each spot looking as white as possible, with little or no Red, Blue, or Green halo.

**STEP 10: Storing the new deflection data.**

Once you are satisfied that all areas are in spec, depress **STORE**.

This writes the new convergence digital data into the monitor's RAM memory.

**END OF PROCEDURE**

---

**SECTION 4: Adjustment Procedure of CM-2073-A Monitor****STEP 1: PREPARATION.**

- A. Insure monitor has warmed up for at least 30 minutes, preferably with an image displayed. The gray scale pattern is suitable.
- B. Perform self calibration of Minolta, and check that Minolta is properly programmed.
- C. Affix probe of Minolta TV Color Analyzer II at center of screen.
- D. After warm up, but before proceeding with adjustments, degauss monitor by depressing front panel degauss button for 10 seconds.

**STEP 2: Red, Green, Blue CUTOFF & DRIVE Adjustments.**

- A. Set Minolta to the *ANALYZER MODE*.
- B. Set User *CONTRAST* control fully *CLOCKWISE*.
- C. Set User *BRIGHTNESS* control fully *CLOCKWISE*.

**NOTE:**

You should keep the room as dark as possible and/or shield the screen from any bright light source.

- mouse in upper right corner*
- D. Using the *PATTERNS* program, display the *RASTER REGULATION* pattern.
  - E. Adjust Red, Green and Blue *CUTOFF* controls for three equal readings of *3.0*.
  - F. Display *100%* white pattern.



- G. Adjust Red, Green and Blue *DRIVE* controls so that all three readings are equal to 100.

**NOTE:**

If Blue cannot be adjusted to 100, then bring up Red and Green to the level of Blue. The *maximum* level Blue can be is 120, and if this level cannot be achieved, return the monitor to manufacturing.

- H. If adjustment was required in step 2G, go back to step 2D, otherwise proceed. You will loop on steps 2D thru 2G until both adjustments are 3.0 and 100 (or what ever level you used if Blue could not be set to 100).

**STEP 3: Red, Green, Blue BACKGROUND & DRIVE Adjustments.**

- A. Display the 100% pattern.

- B. Set User BRIGHTNESS control fully *CLOCKWISE*.

- C. Set MINOLTA to *CHROMA* mode.

- D. Set User CONTRAST control for 10.0 fl. on the bottom scale of the MINOLTA.

- E. Set MINOLTA to *ANALYZER* mode.

- F. Adjust Red, Green and Blue *BACKGROUND* controls for three equal readings of 20.

- G. Set User CONTRAST control fully *CLOCKWISE*.

- H. Adjust Red, Green and Blue *DRIVE* controls for three equal readings of 100, or to the level of blue as you did in step 2G. Allow the values time to settle after you move user contrast.

This will probably have  
to be 5

## LAB 8.1: Adjustments

- I. If adjustment was required in step 3H, go back to step 3C, otherwise proceed. You will loop on steps 3C thru 3H until you have all the adjustments in spec. If you cannot bring them into spec, return the monitor to manufacturing.

**STEP 4: Red, Green and Blue CUTOFF & BACKGROUND Adjustments.**

- A. Display the *100%* white PATTERN.
- B. Set User CONTRAST control fully *CLOCKWISE*.
- C. Set User BRIGHTNESS control fully *COUNTER-CLOCKWISE*.
- D. The three readings at this point will probably be different. Select the middle value (of the three) as a reference. **Allow time for values to settle.**
- E. Leaving the reference color alone, adjust the other two *CUTOFF* adjustments to equal the reading of the reference color. The reference reading will probably change. **DO NOT** adjust the reference, simply adjust the other two *CUTOFFS* for three equal readings.
- F. Set User BRIGHTNESS fully *CLOCKWISE*.
- G. Adjust Red, Green, and Blue *BACKGROUND* controls for three equal readings of *100*, or the level of blue. **Allow time for values to settle.**
- H. If adjustment was required in step 4G, go back to step 4C, otherwise proceed. You will loop on steps 4C thru 4G until you bring the adjustments into spec. If you cannot bring them into spec, then return the monitor to manufacturing.

**STEP 5: Red, Green, Blue CUTOFF & DRIVE Adjustments.**

- A. Set the Minolta to *ANALYZER MODE*.

- B. Set User CONTRAST control fully *CLOCKWISE*.
- C. Set User BRIGHTNESS control fully *CLOCKWISE*.
- D. Display the *RASTER REGULATOR* pattern.
- E. Adjust Red, Green and Blue *CUTOFF* controls for three equal readings of 3.0. **Allow time for values to settle.**
- F. Display the *100% White* pattern.
- G. Adjust Red, Green and Blue *DRIVE* controls so that all three readings equal *100*, or the value of blue. **Allow time for values to settle.**
- H. **If** adjustment was required in step 5G, go back to step 5D, otherwise proceed. You will loop on steps 5D thru 5G until you bring the adjustments into spec. If you cannot bring them into spec, return the monitor to manufacturing.

**STEP 6: Background check.**

- A. Set User BRIGHTNESS fully *CLOCKWISE*.
- B. Set User CONTRAST fully *CLOCKWISE*.
- C. Display *Raster Regulation* pattern.
- D. There should be a lit background extending **AT LEAST 1/16** of an inch to the left of the left vertical column of dots. ( 1/16" is a **MINIMUM**; any greater amount is fine. ) If not, adjust *HORIZONTAL PHASE* adjustment so there is. If *HORIZONTAL PHASE* is adjusted, the area within the dots will have to be recentered using the *HORIZONTAL CENTERING* control. The case will have to be removed to gain access to the *HORIZONTAL SIZE* and/or *HORIZONTAL CENTERING* adjustments. Normally, there should be no need to make these adjustments.

wolfe

LAB 8.1: Adjustments

**STEP 7: Color tracking check.**

- A. Set User BRIGHTNESS fully **CLOCKWISE**.
- B. Set User CONTRAST fully **CLOCKWISE**.
- C. Display the 100% White pattern.
- D. Set the MINOLTA to **CHROMA MODE**.
- E. Record the top two readings. **Allow time for values to settle.**
- F. Adjust User CONTRAST so that *bottom* reading on the MINOLTA is equal to 10.0.
- G. Now, compare the top two readings with those taken in step 7E. The two *current* readings should not deviate from the two taken in step 7E by more than 10. For example, if the readings in step 7E are  $x = 313$  and  $y = 329$ , then the readings from step 7F should be in the range  $x = 303-323$  and  $y = 319-339$ . If these readings are out of specification, perform **STEP 3** again. (After performing STEP 3, return to the beginning of STEP 7)
- H. Set User CONTRAST fully **CLOCKWISE**.
- I. Set User BRIGHTNESS for bottom reading equal to 10.0, or fully **COUNTER-CLOCKWISE**, whichever is higher.
- J. Now, compare the top two readings with those taken in step 7E. The two *current* readings should not deviate from the two taken in step 7E by more than 10. For example, if the readings in step 7E are  $x = 313$  and  $y = 329$ , then the readings from step 7I should be in the range  $x = 303-323$  and  $y = 319-339$ . If these readings are out of specification, perform **STEP 4** again. (After performing STEP 4, return to the beginning of STEP 7)

315  
330  
250

K. Set User BRIGHTNESS fully *CLOCKWISE*.

L. Display the *RASTER REGULATION* pattern.

M. Now, compare the top two readings with those taken in step 7E. The two *current* readings should not deviate from the two taken in step 7E by more than 10. For example, if the readings in step 7E are  $x = 313$  and  $y = 329$ , then the readings from step 7L should be in the range  $x = 303-323$  and  $y = 319-339$ . If these readings are out of specification, perform **STEP 5** again. (After performing **STEP 5**, return to the beginning of **STEP 7**)

N. If the readings taken in steps 7F, 7I and 7L are within the prescribed range, the monitor is in specification and you are finished. If not, the monitor is out of specification. In this case, either try again, or replace the monitor and return it to manufacturing.

## SECTION 5: Power Supply Adjustments

Adjustments made to the power supply are rather simple. They are set to the values listed below by manufacturing. When manufacturing sets the values, a special load board is installed that simulates a normal system load.

You should check the voltages from time-to-time to insure that they are at the listed levels. Make your measurements at the power supply.

Power Supply Measurements		
Channel	Wire Code	Setting*
1	Orange/White	+5.00vdc
2	Gray	+12.10vdc
3	Yellow	-12.10vdc
4	Purple	-5.24vdc

\* As measured at the power supply terminal block.

- If you upgrade a 2400 to a 2400T, you may have to add the second power supply. You must balance the second supply with the first supply using the load board and the upgrade procedure.
- If, after adding the second power supply, you experience tape errors, you may have to set the voltages per TECH AID GEN-36.

## SECTION 6: Diagrams

DIAGRAM 1: Minolta at Power On

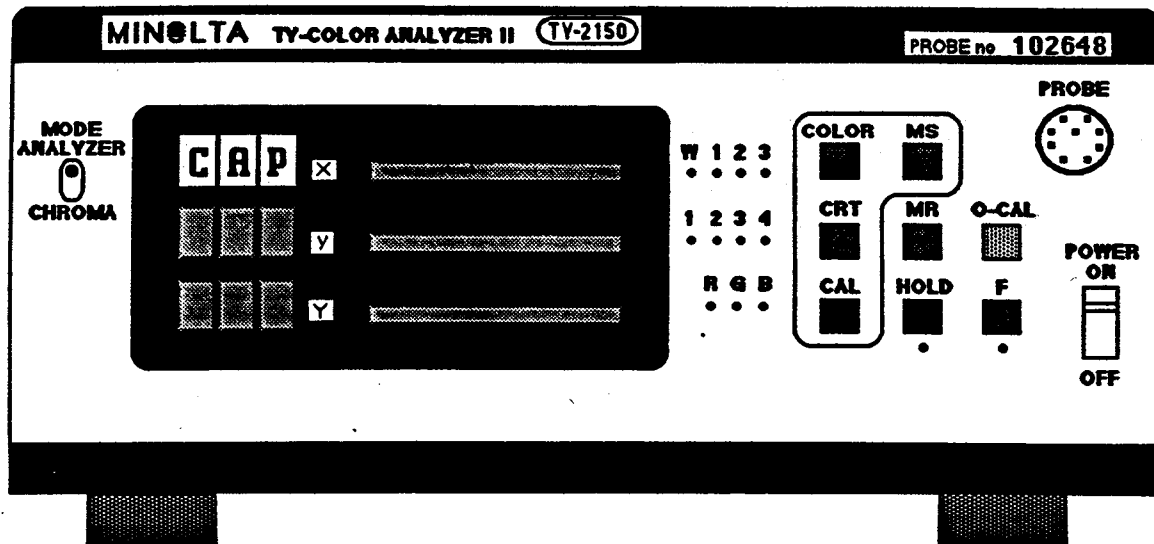
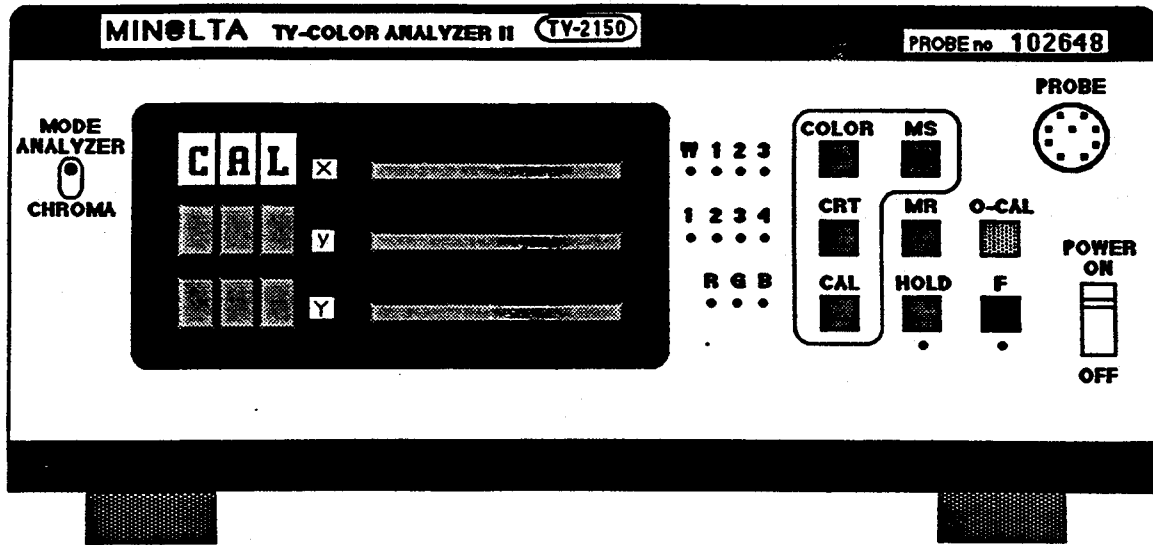




DIAGRAM 2: Minolta - Depressing O-CAL



**DIAGRAM 3: Minolta Channel Data**

**TABLE 1**

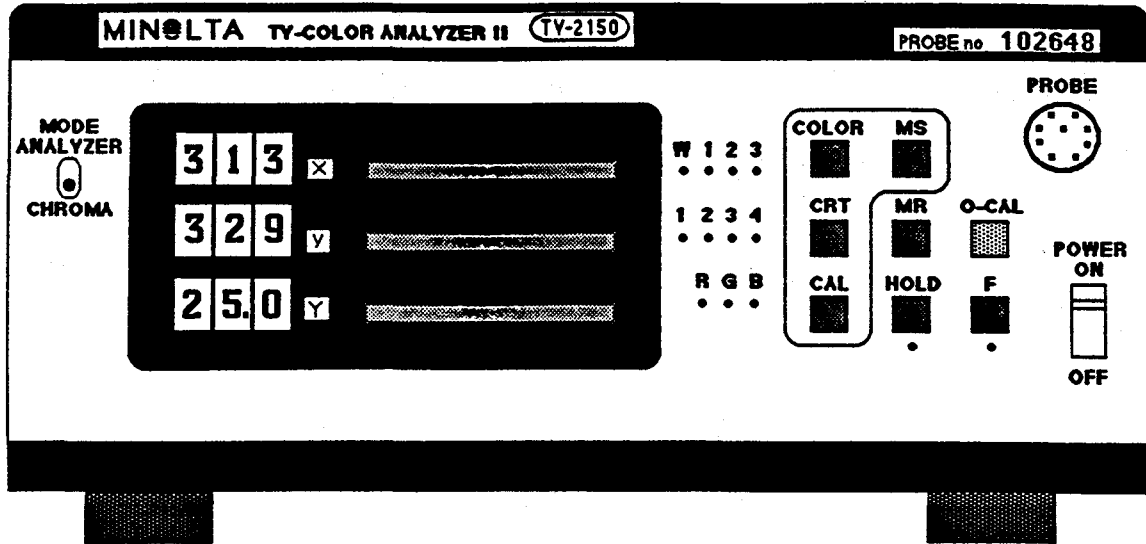
Color Channel	Reference Data		
	x	y	Y
W	313	329	25.0
1	313	329	25.0
2*	0	0	0
3	281	311	10.0

\* Channel 2 is not used.

**TABLE 2**

CRT Channel	Monitor Phosphor Data
1	Both 19" monitors
2	15" monitor
3	Not used
4	19", 30 Hz. monitor

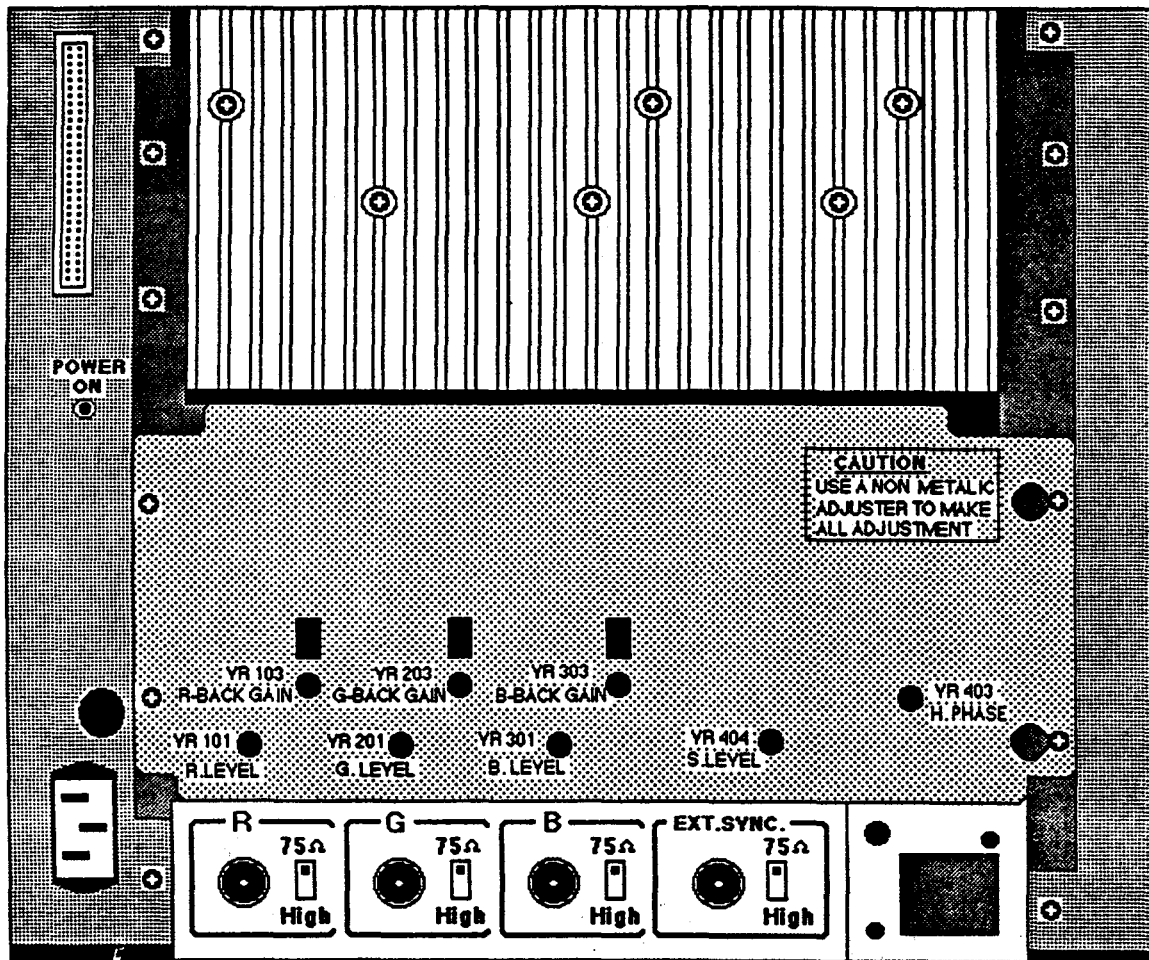
DIAGRAM 4: Minolta - Channel Pair W-1 with MR depressed



**DIAGRAM 5: Patterns Help Menu**

<b>100% white field</b>
<b>90% white field</b>
<b>Focus 1</b>
<b>Focus 2</b>
<b>Focus 3</b>
<b>Raster Regulation</b>
<b>Gray scale</b>
<b>Convergence</b>
<b>Black level 1</b>
<b>Black level 2</b>
<b>Quit</b>

**DIAGRAM 6: Monitor Rear View without cover**

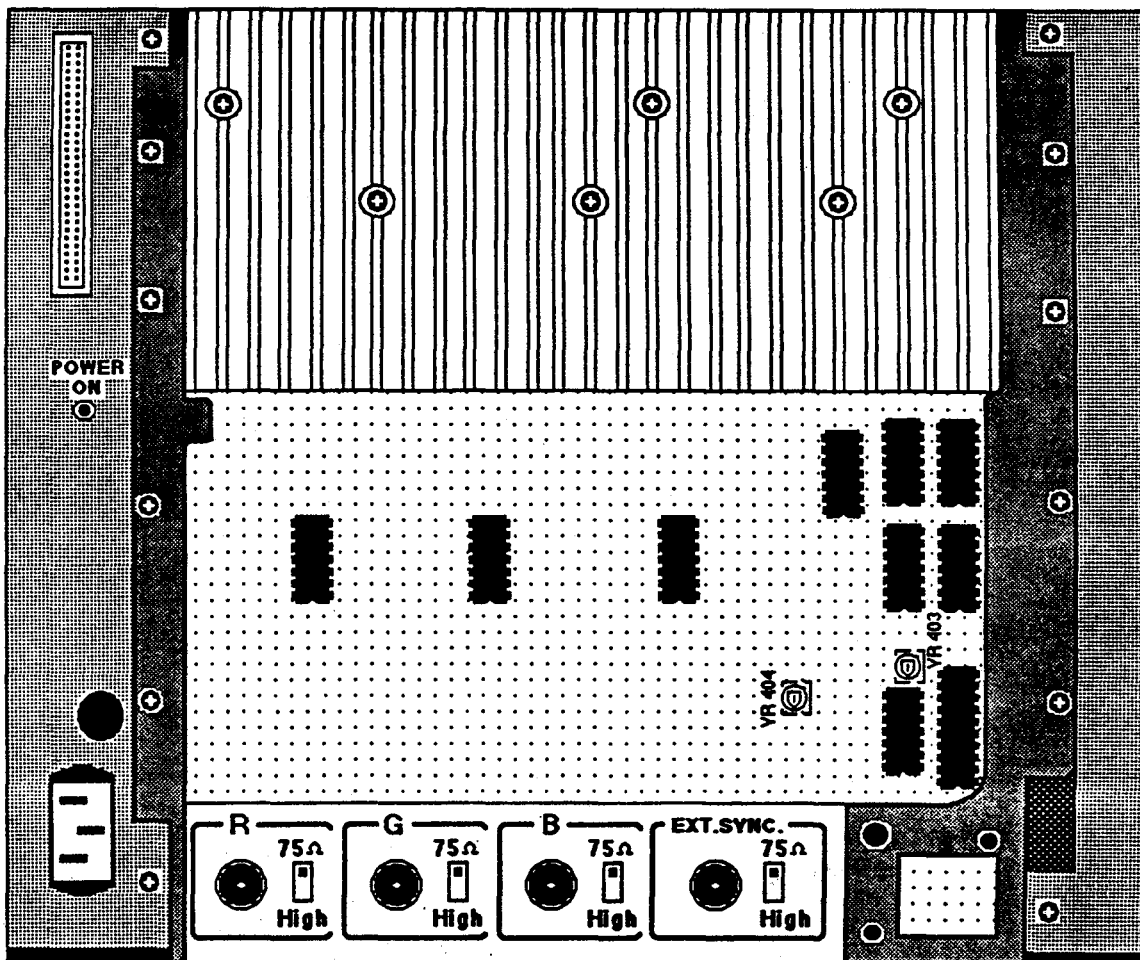


**NOTE:**  
The plate the covers the logic board with all the adjustment pots has several other holes that allow access to other pots.

Only the holes allowing access to the adjustments that you are going to make are shown by the drawing.

**DO NOT** adjust any other pots of the monitor.

**DIAGRAM 7: Screen Position and Sync Adjustment Pots**



**NOTE:**

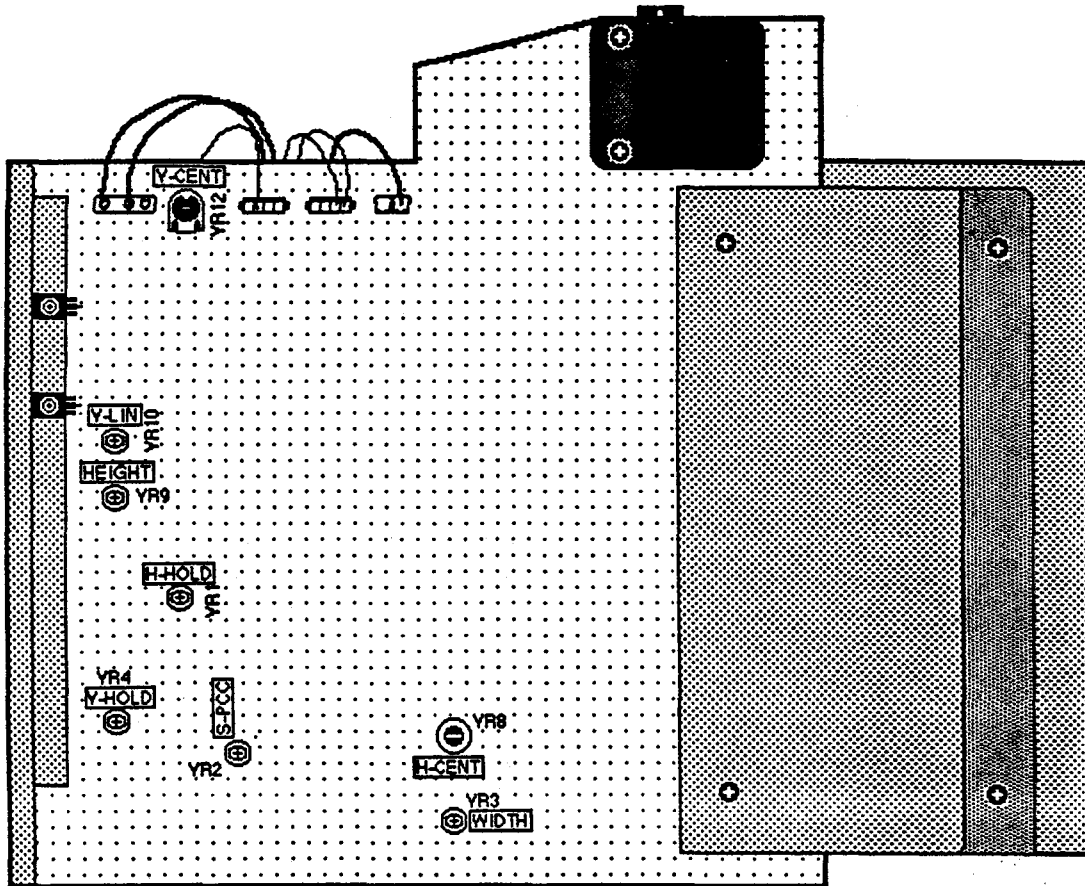
This drawing only shows the position of the H.Phase (YR 403) and S.Level (YR 404) adjustment pots. These pots are used:

H.Phase used during Horizontal Screen Adjustment.

S.Level used during Synchronization Adjustment.

**DO NOT** change the position of any other adjustment pots.

**DIAGRAM 8: Monitor Side View - Adjustment Pots**



**NOTE:**

This drawing only shows the position of those adjustment pots that the F.E. may have to locate. DO NOT move any other pots located on the board.

The following pots are used for the listed adjustment:

Synchronization Adjustment - YR1, YR4

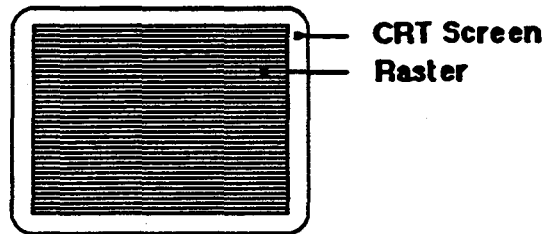
Horizontal Screen Location - YR8, YR1

Vertical Screen Location - YR12

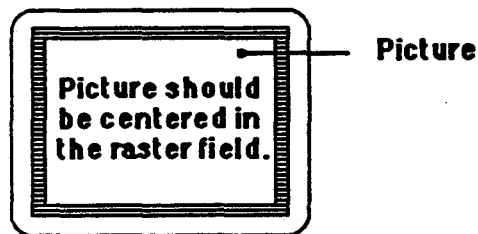
Picture Size - YR3, YR9, YR2, and YR10

**DIAGRAM 9: Display Centering**

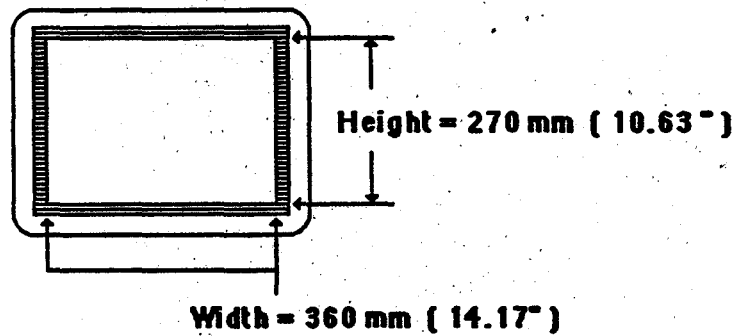
**Example 1**



**Example 2**

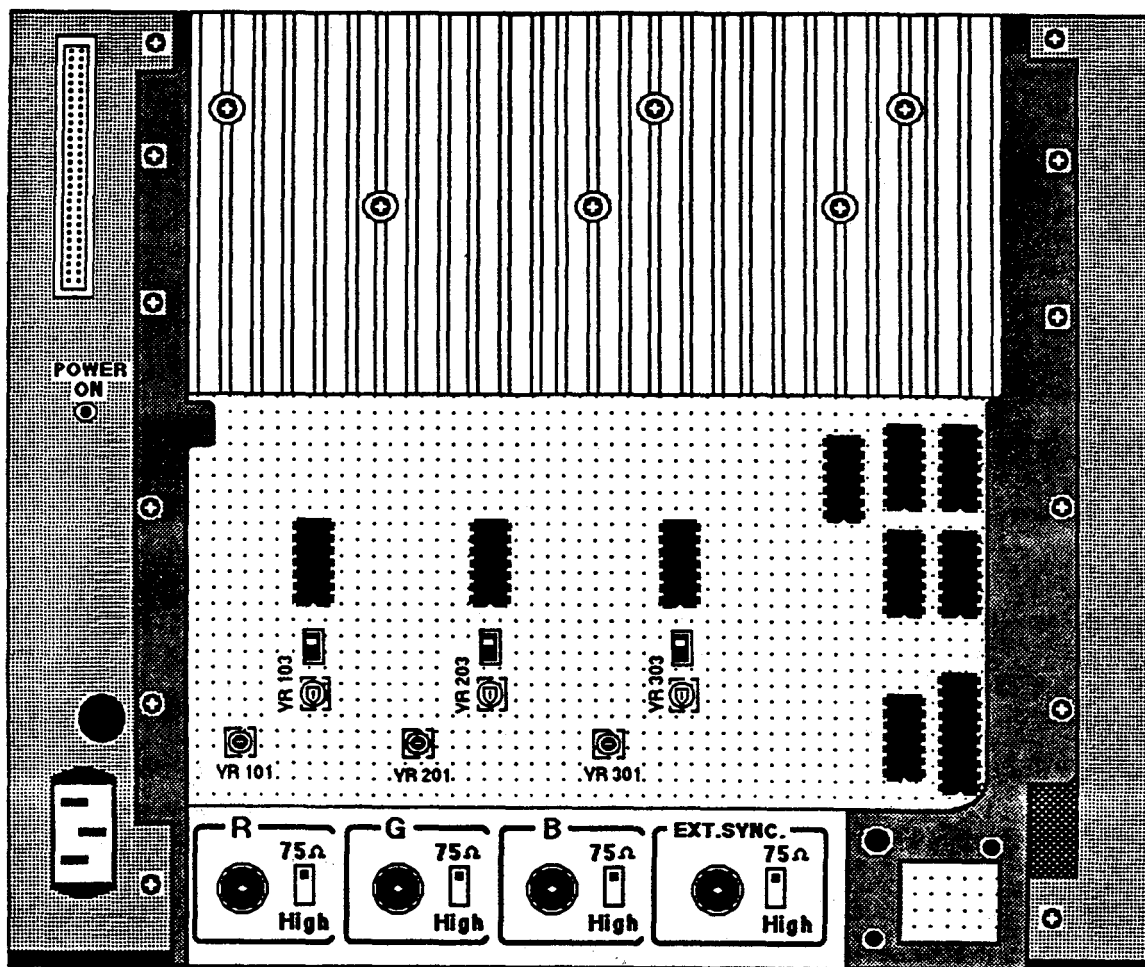


**Example 3**





**DIAGRAM 10: Monitor Rear View - Color Adjustment Pots**



**NOTE:**  
 You **DO NOT** have to completely remove the screws that hold the logic board cover plate.

This drawing only shows the six adjustment pots that you will move when making the color adjustments. **DO NOT** move the position of any other pots.

- |                          |                      |
|--------------------------|----------------------|
| VR 103 = Red Back Gain   | VR 101 = Red Level   |
| VR 203 = Green Back Gain | VR 201 = Green Level |
| VR 303 = Blue Back Gain  | VR 301 = Blue Level  |

DIAGRAM 11: Monitor Top View - Color Adjustment Pots

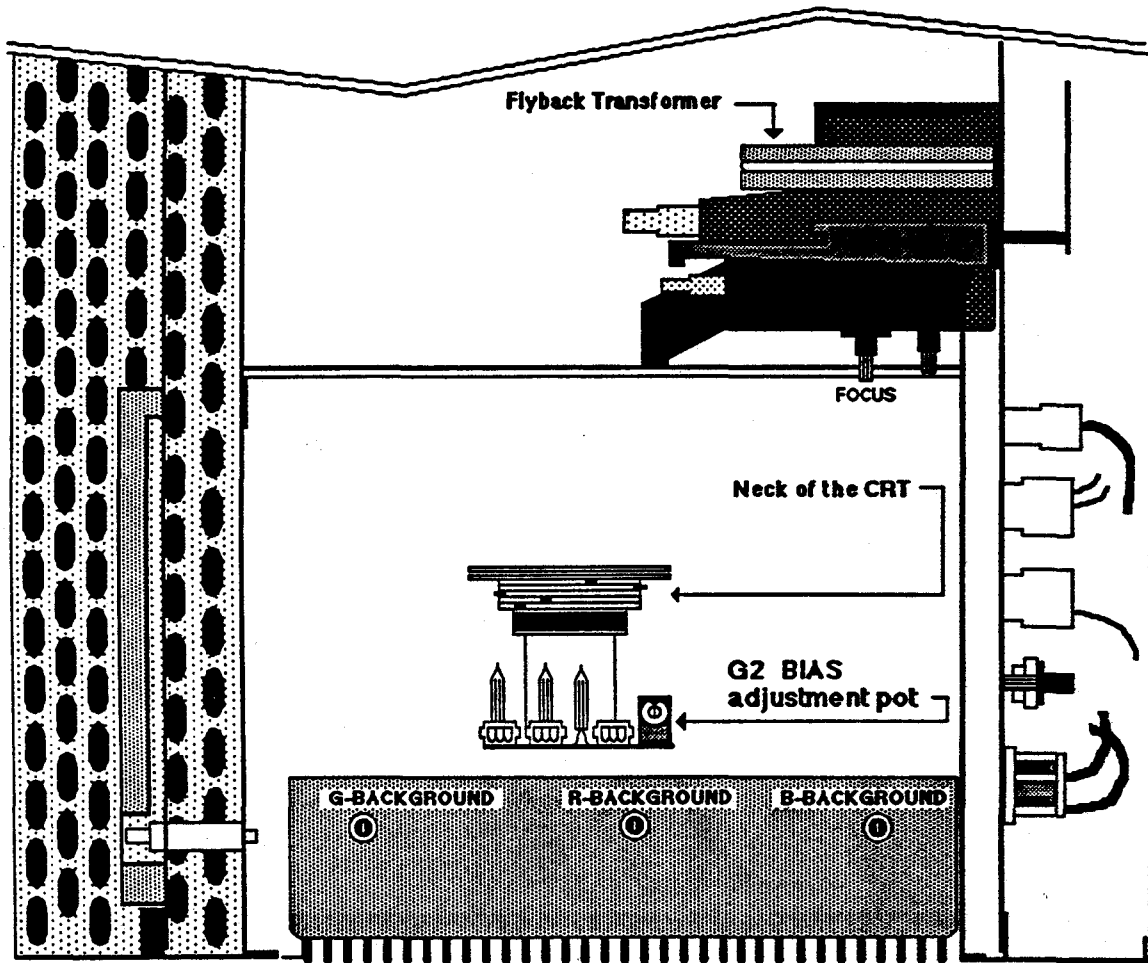
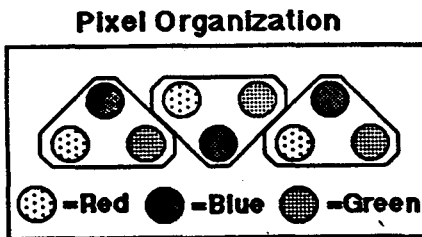
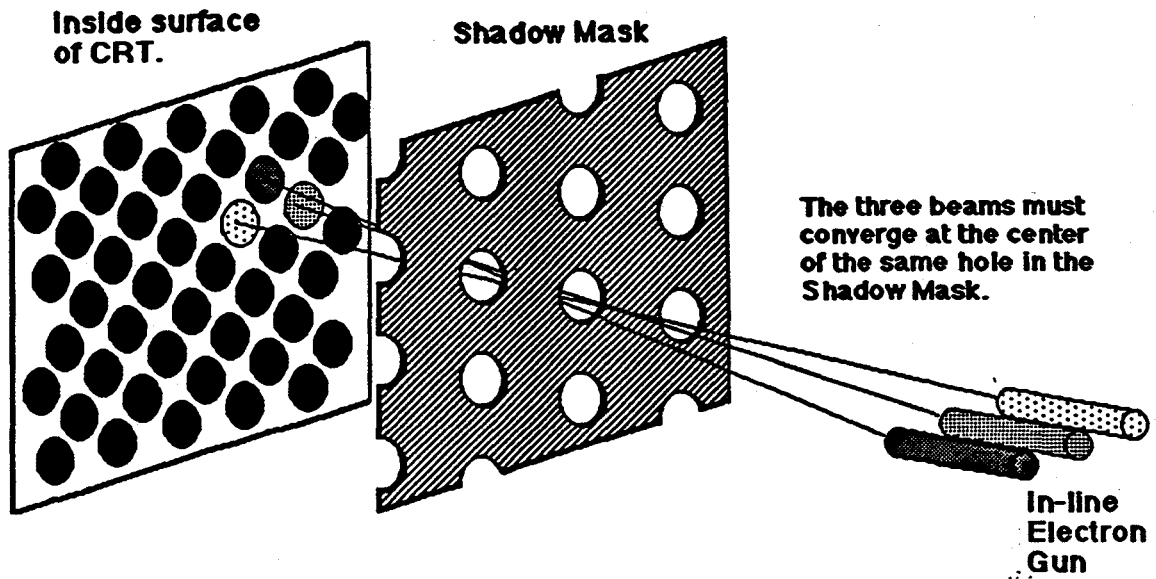
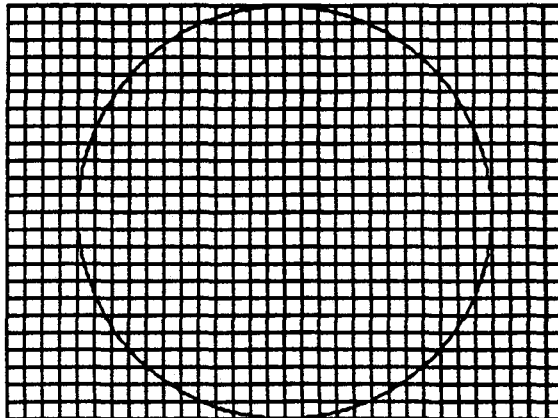


DIAGRAM 12: Convergence



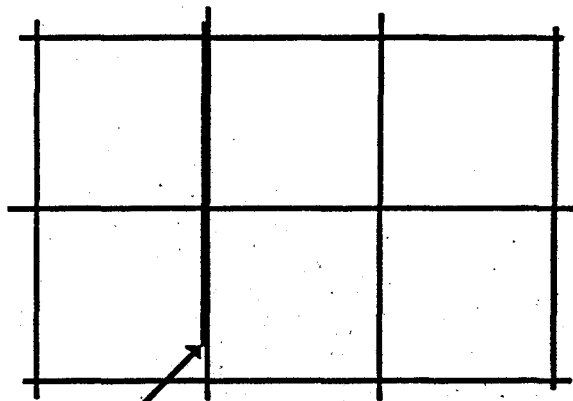
**DIAGRAM 13: Convergence Test Pattern**

**CONVERGENCE PATTERN**



The lines should be all white. If convergence is off in an area of the screen, then you will see Red, Green, or Blue along side portions of a line or lines. The diagram below illustrates this point.

**ENLARGED SECTION OF PATTERN**



A Red, Blue, or Green halo next to pattern line.

This line is White, but it has a "Halo" of Red, Blue, or Green next to it. ( The convergence is out in this section on the screen )

DIAGRAM 14: Connecting Convergence Tester

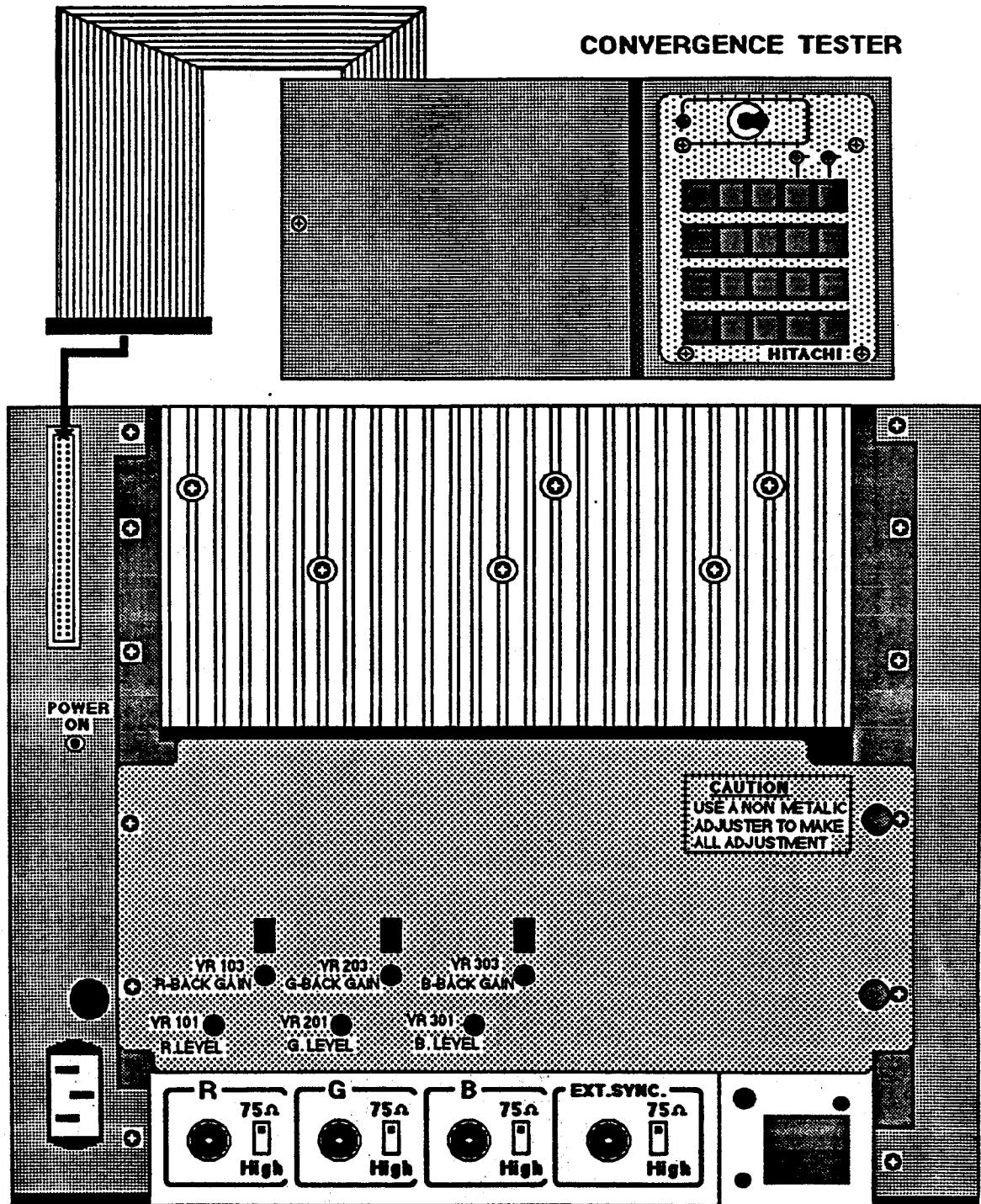
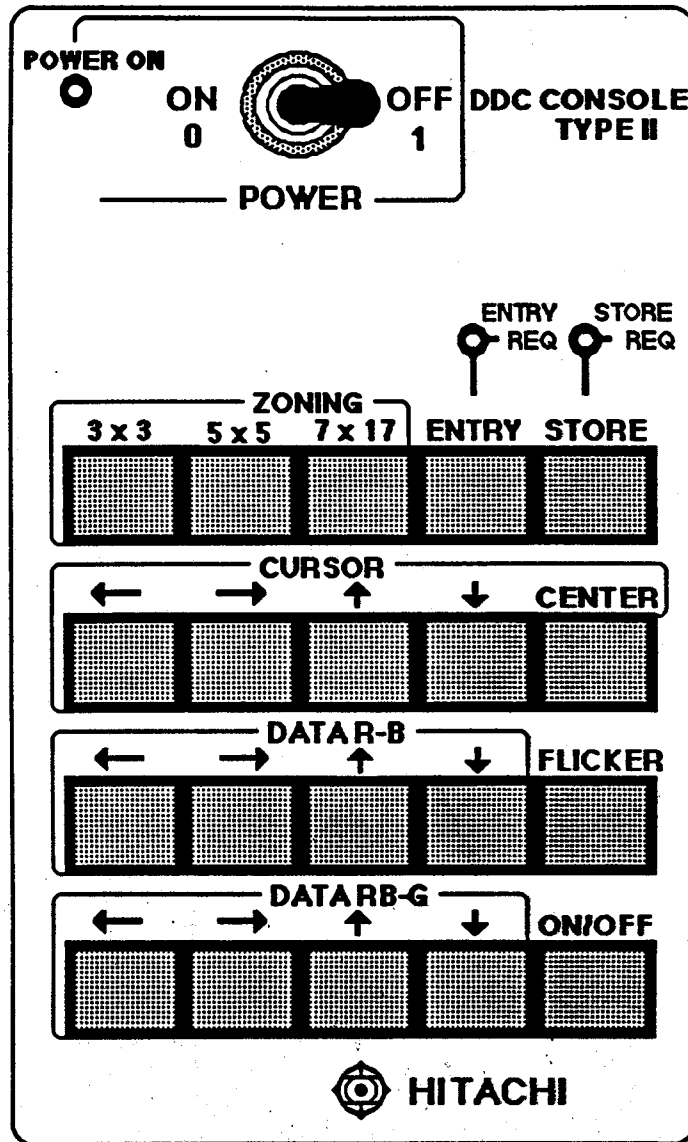
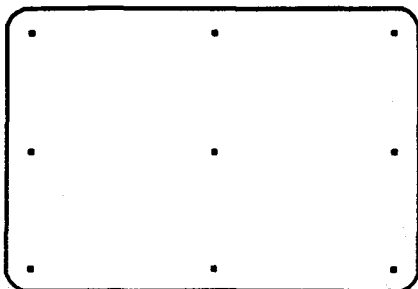
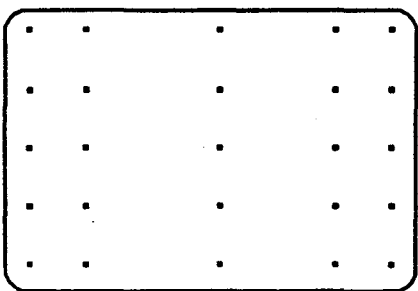
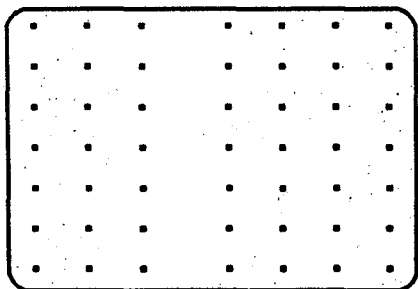


DIAGRAM 15: Convergence Tester Panel

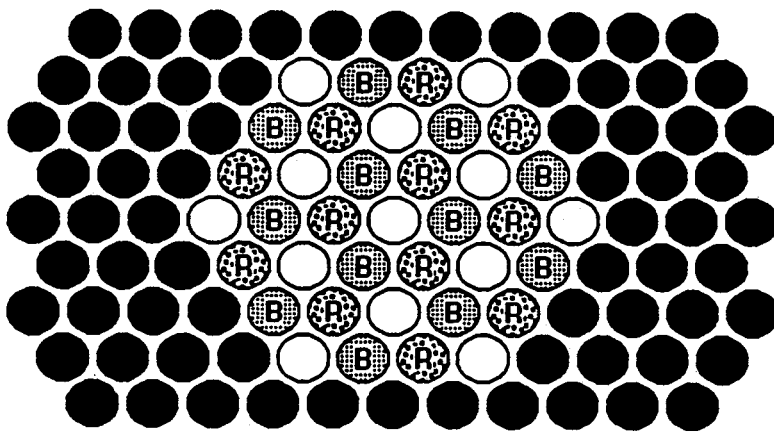


**DIAGRAM 16: Test Patterns****FIGURE 1****( 3 x 3 ) dot display****FIGURE 2****( 5 x 5 ) dot display.****NOTE:**  
The distance between the  
columns is not equal. Do  
not become concerned.**FIGURE 3****( 7 x 17 ) dot display.****NOTE:**  
Again, the distance is not equal  
between dot columns.

**DIAGRAM 17: Convergence Examples**

**DIAGRAM 1**

**ONLY BLUE AND RED DOTS WITH GOOD CONVERGENCE.**



**DIAGRAM 2**

**BLUE AND RED DOTS OUT OF CONVERGENCE**

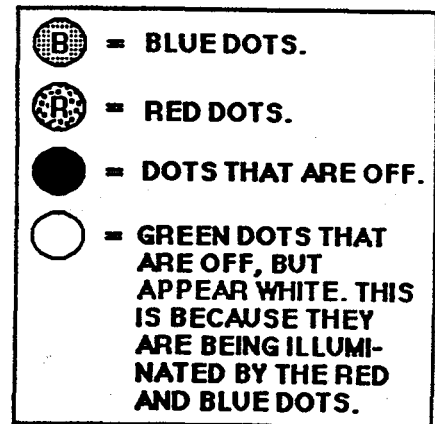
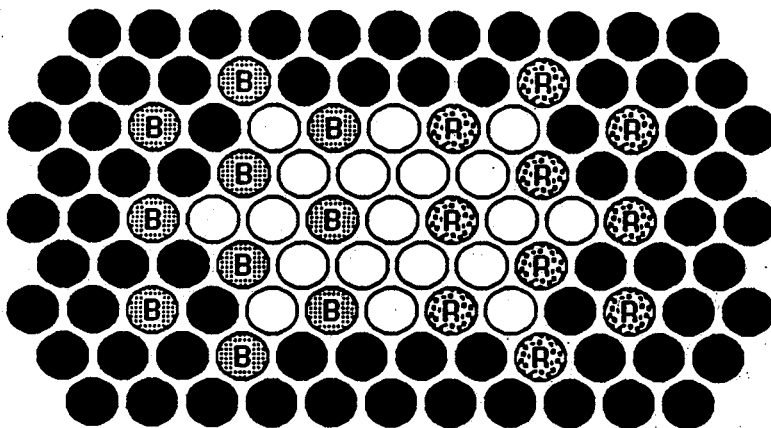
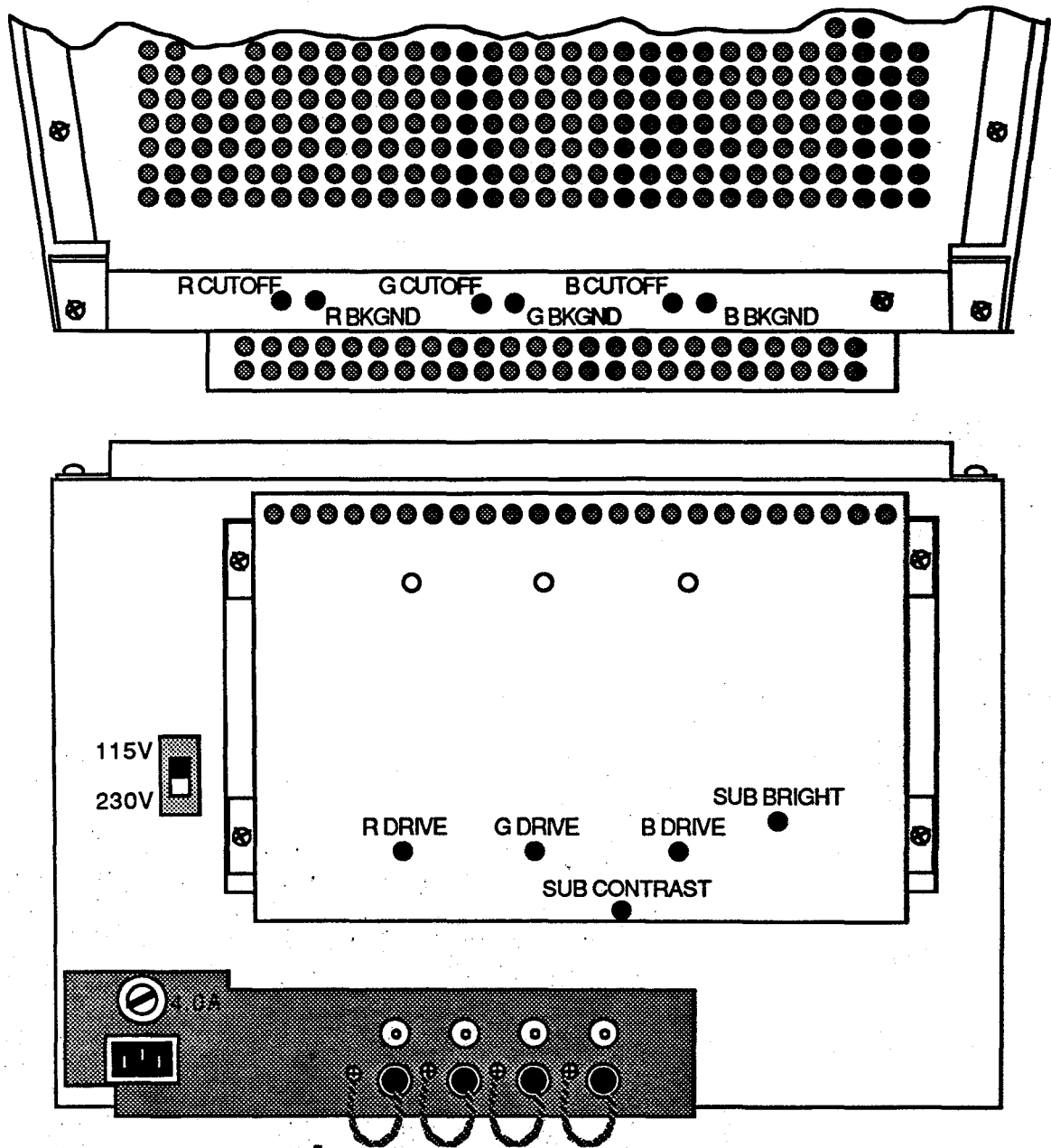
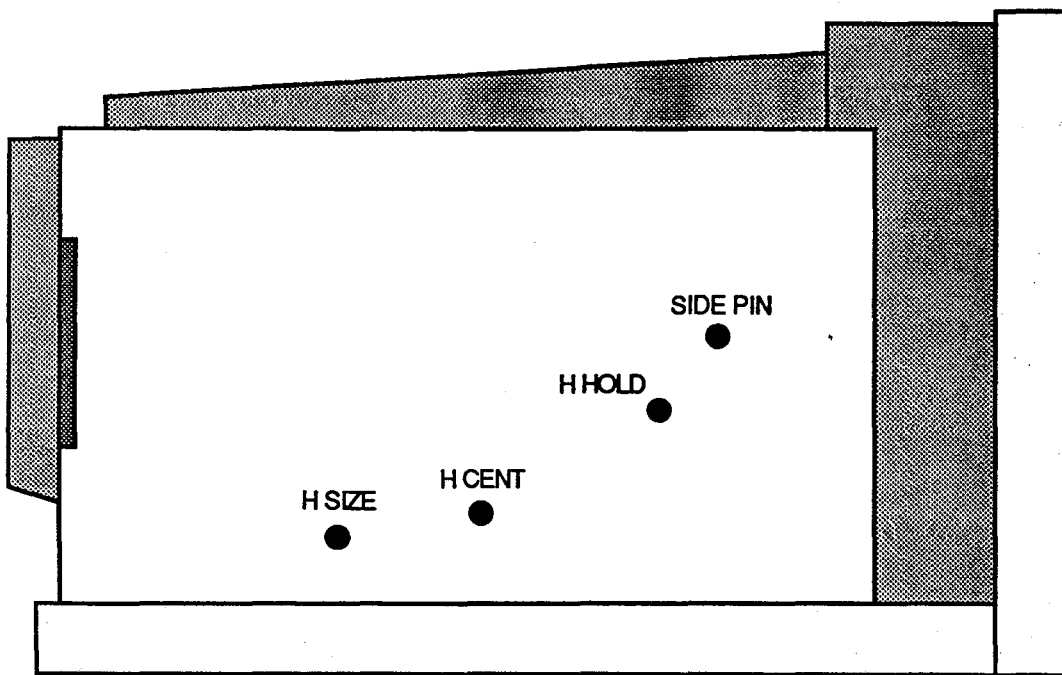




DIAGRAM 18: Hitachi CM2073A Color Adjustment Pots



**DIAGRAM 19: Hitachi CM2073A Background Adjustment Pots**



---

**SECTION 7: Review**

Student Name:

Please complete the review and hand it into your instructor.

1. What channel pair of the Minolta Color Analyzer must contain valid reference data if the other channel pairs are to be programmed with reference data?

---

2. All four Minolta CRT channels will hold a percentage value that will define to what level you will adjust your monitor color to the reference color. You can display these values on only one CRT channel. Which channel?

---

3. What is the maximum *ft. lamberts* SGI recommends that you set your monitor drive to?

---

4. If the Minolta needs programming and you are using your monitor to produce the Minolta reference values, what trick can you use to increase the monitor's drive if the front panel Brightness control is insufficient.

---

5. What must the minimum warm up time be, prior to performing color adjustments on the monitor?

---

6. What must your color signal line termination be?

---

LAB 8.1: Adjustments

---

7. What are the recommended values for the physical size of the monitor picture area?

---

8. If you over drive the monitor, what damage can result?

---

9. When performing the convergence adjustment on the HM-4619-C monitor, what color cable do you remove before you begin the adjustment?

---

10. Convergence can be out of spec. in specific areas of the CRT screen.

TRUE

FALSE