sgi™

Hardware Quick-reference Booklet

(Origin™ and Onyx2™ Series)

HMQ-380-C

Any shipment to a country outside of the United States requires a letter of assurance from Silicon Graphics, Inc.

Requests for copies of SGI service publications should be directed to:

Web site: *http://servinfo.csd.sgi.com*

SGI Customer Service - Logistics
1100 Lowater Road
P. O. Box 4000
Chippewa Falls, WI 54729-0078
USA

Comments about this publication should be directed to:

E-mail: *spt@sgi.com*

SGI Customer Service
Service Publications and Training
890 Industrial Blvd.
P.O. Box 4000
Chippewa Falls, WI 54729-0078
USA

Hardware Quick-reference Booklet

HMQ-380-C
Origin™ 2000 and Onyx2™ Series Systems
Last Modified: November, 1999

Procedures 197

Appendix 259

Index 289

Figures

Tables

Record of Revision

November 1997

Original printing.

April 1998

This revision includes new and revised command interpreter information, maintenance procedures, and an index. It also provides quick-reference information about the MetaRouter and IP31 node board compatibility.

December 1998

This revision adds MetaRouter cabling information, the procedure for flashing the IP31 node board, and information about configuring multiple pipes and X servers for IRIX 6.5. It updates the power-on boot process, the flashing MMSC firmware procedure for IRIX 6.5, and the testing memory with POD procedure. This revision also makes various corrections and enhancements throughout the booklet.

November 1999

This revision eliminates O2 information, updates processor and redundant-power-supply material, provides references to configuration information for 256-processor Origin systems, and makes various corrections and enhancements throughout the booklet.

Introduction

This booklet provides quick-reference information for Origin and Onyx2 systems.

Notational Conventions

This booklet uses the following notational conventions:

- Command variables, references to document titles, and terms are in *italics*.
- References to commands and screen output are in `courier`.
- Commands that you enter on the keyboard are in **`courier bold`**.
- [] Indicates an optional item.
- < > Indicates a required variable within an optional item.

Maintenance Cabling and Connectors

This section contains maintenance cabling information and connector pin assignments for the following systems:

Several cables and cable converters are shipped with an Origin computer system. You can use these cables to connect an ASCII terminal, workstation, or personal computer (PC) to the MMSC and to other maintenance connections.

Some connections require a straight cable connection, and other connections require a null modem connection. Basically, the transmit pin on one end of the straight cable is wired to the transmit pin on the other end. The transmit and receive wires of the null modem cable are crossed.

Table 1 through Table 7 list the maintenance cable connections, the type of connections required, and the SGI cable part numbers that are used to make these connections. Table 8 describes the cables.

Note: The revision number, which comprises the last three numbers of the part number, may indicate a different type of cable connector. For example, the 018-0672-001 is a straight connector and 018-0672-002 is a null modem connector.

Table 1. Indy Connections to an Origin 2000 System

Connection Description	Connection Type	Physical Connection
Indy serial port to MMSC COM 1 and COM 5	Null modem	018-0650-001 -> 018-0671-001 -> 018-0672-002 -> 018-0650-001
Indy serial port to MSC front port	Null modem	018-0650-001 -> 018-0671-001 -> 018-0672-002 -> 018-0650-001
Indy serial port to MSC rear port	Straight	018-0650-001 -> 018-0671-001 -> 018-0672-002
Indy serial port to BaseIO serial port	Straight	018-0650-001 -> 018-0671-001 -> 018-0672-002

Table 2. Indy Connections to an Origin 200 System

Connection Description	Connection Type	Physical Connection
Indy serial port to Origin 200 AUX port	Null modem	018-0650-001 -> 018-0671-001 -> 018-0672-002 -> 018-0650-001
Indy serial port to Origin 200 console port	Straight	018-0650-001 -> 018-0671-001 -> 018-0672-002

The PC serial DB-9 connector uses pin 2 for receive data and pin 3 for transmit data; the SGI DB-9 uses pin 3 for receive data and pin 2 for transmit data. Because pins 2 and 3 of the connectors are opposite, the cable configurations that were straight connections for the Indy workstation will be null modem connections for the PC, and the straight connections for the PC will be null modem connections for the Indy workstation.

Table 3. PC Connections to an Origin 2000 System

Connection Description	Connection Type	Physical Connection
PC serial port to MMSC COM 1 and COM 5	Straight	018-0671-001 -> 018-0672-002 -> 018-0650-001
PC serial port to MSC front port	Straight	018-0671-001 -> 018-0672-002 -> 018-0650-001
PC serial port to MSC rear port	Null modem	018-0671-001 -> 018-0672-002
PC serial port to BaseIO serial port	Null modem	018-0671-001 -> 018-0672-002

Table 4. PC Connections to an Origin 200 System

Connection Description	Connection Type	Physical Connection
PC serial port to Origin 200 AUX port	Straight	018-0671-001 -> 018-0672-002 -> 018-0650-001
PC serial port to Origin 200 console port	Null modem	018-0671-001 -> 018-0672-002

Table 5. Wyse Terminal Connections to an Origin 2000 System

Connection Description	Connection Type	Physical Connection
Wyse modem port to MMSC COM 1 and COM 5	Null modem	018-8114-001
Wyse modem port to MSC front port	Null modem	018-8114-001
Wyse modem port to MSC rear port	Straight	018-0671-001
Wyse modem port to BaseIO serial port	Straight	018-0671-001

Table 6. Wyse Terminal Connections to an Origin 200 System

Connection Description	Connection Type	Physical Connection
Wyse modem port to Origin 200 AUX port	Null modem	018-0671-001 -> 018-0650-001
Wyse modem port to Origin 200 console port	Straight	018-0671-001

Table 7. Internal Connections

Connection Description	Connection Type	Physical Connection
MMSC COM 2 and COM 3 to rear MSC port	Straight	018-0644-001
MMSC COM 4 to BaseIO serial port	Straight	018-0644-001

Table 8. Cable Descriptions

Cable Part Number	Cable Description	Connection Type
018-0644-001	DIN 8 male to DB-9 female cable	Straight
018-0650-001	DB-9 male to DIN 8 male converter	Null modem
018-0671-001	DB-25 male to DB-9 female cable	Straight
018-0672-002	DB-25 female to DB-9 female converter **Note:** 018-0672-001 is a similar cable, but it has a straight connection.	Null modem
018-8104-001	DB-9 female to DIN 8 male converter	Straight
018-8114-001	DB-25 male to DIN 8 male cable	Null modem
018-8223-001	DIN 8 male to DIN 8 male cable	Null modem

Origin 200

Note: If you need to configure an SGI workstation as the system console, refer to "Configuring the System Console" on page 204.

The following topics are covered in this section:

Figure 1. Origin 200 Serial Port Connection

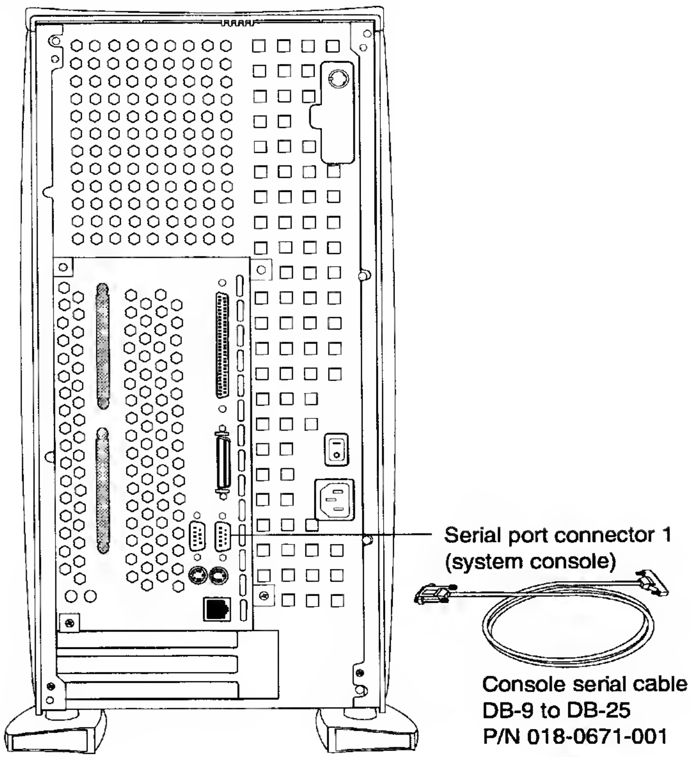


Table 9. Origin 200 Serial Port Pin Assignments, RS-232 Mode

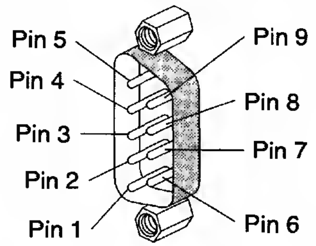
Pin	Assignment	Pin Numbers
1	Data Carrier Detect (DCD)	
2	Receive Data (RD)	
3	Transmit Data (TD)	
4	Data Terminal Ready (DTR)	
5	Signal Ground (SG)	
6	Data Set Ready (DSR)	
7	Request To Send (RTS)	
8	Clear To Send (CTS)	
9	Not Connected	

Table 10. Origin 200 Serial Port Pin Assignments, RS-422 Mode

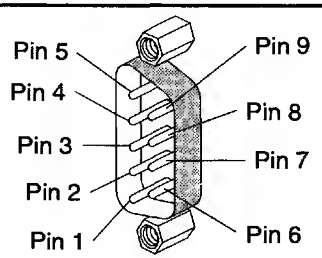
Pin	Assignment	Pin Numbers
1	Reserved	
2	Receive Data (RXD –)	
3	Transmit Data (TXD –)	
4	Transmit Data (TXD +)	
5	Signal Ground (GND)	
6	Receive Data (RXD +)	
7	Output Handshake (HSKo)	
8	Input Handshake (HSKi)	
9	(Reserved)	

Table 11. Origin 200 Parallel Port Pin Assignments

Pin	Assignment	Pin Numbers
1	STB	
2	DATA1	
3	DATA2	
4	DATA3	
5	DATA4	
6	DATA5	
7	DATA6	
8	DATA7	
9	DATA8	
10	ACK	
11	BUSY	
12	PE	
13	ONLINE	
14	PR/SC	
15	NOPAPER	
16	Not Connected	
17	NOINK	
18	Not Connected	
19–30	Signal Ground (GND)	
31	RESET	
32–36	Not Connected	

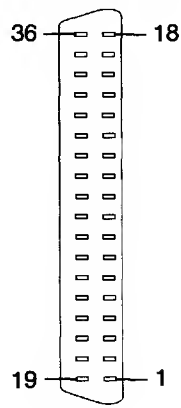
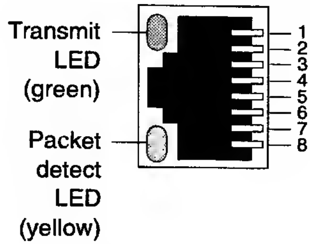


Table 12. Origin 200 Ethernet 10BaseT/100BaseT Port Pin Assignments

Pin	Assignment	Pin Numbers
1	Transmit+	
2	Transmit –	
3	Receive+	
4	(Reserved)	
5	(Reserved)	
6	Receive –	
7	(Reserved)	
8	(Reserved)	

External Interrupt Port

The external interrupt port allows separate machines to send and receive interrupts over a dedicated wire for purposes of intermachine synchronization. The external interrupt interface uses two 1/8-in. (3.5-mm) stereo jacks: one to generate interrupts and one to receive interrupts. The jacks are located on the back of the Origin 200 server (refer to Table 13). Refer to the `ei` man page for more information.

Table 13. Origin 200 External Interrupt Ports

Conductor	Function
Tip	Interrupt (active low)
Ring	+5 V
Sleeve	Chassis ground and cable shield

AUX Port

The AUX port (refer to Figure 2) is a DIN 8 connector that is located at the rear of the Origin 200 system. The AUX port enables you to connect a serial (ASCII) terminal or a modem to the port for remote access to the module system controller (MSC).

The AUX port uses a standard RS-232 interface, and the pin assignments (listed in Table 14) are identical to those used on Silicon Graphics Indy and Indigo2 workstations and the CHALLENGE S server.

Note: You cannot use the Origin 200 AUX port to log into the IRIX operating system.

Figure 2. Origin 200 AUX Port Connection

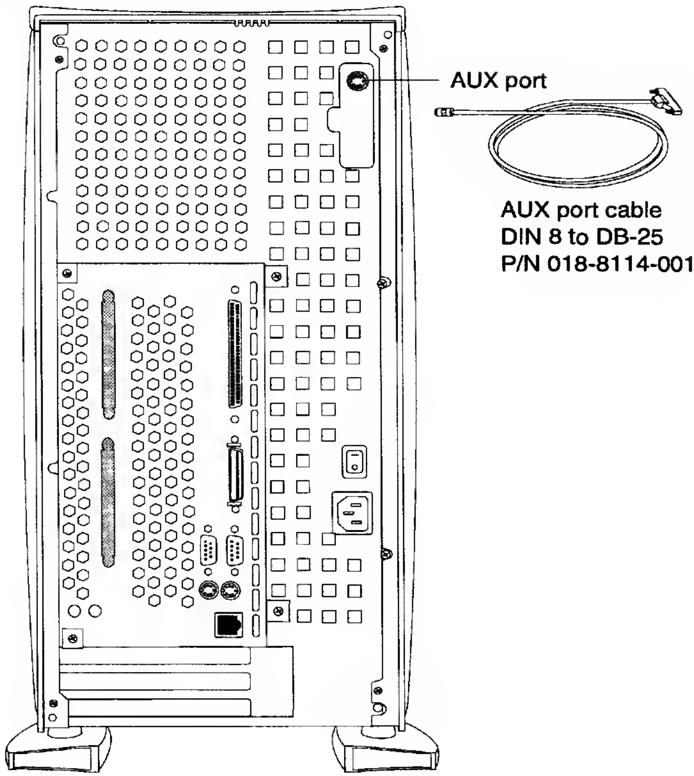
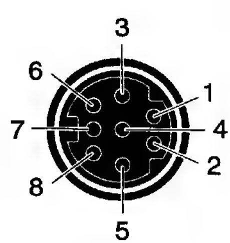


Table 14. Origin 200 AUX Port Pin Assignments

Pin	Assignment	Description	Pin Numbers
1	DTR	Data Terminal Ready	
2	CTS	Clear to Send	
3	TXD	Transmit Data	
4	GND	Ground	
5	RXD	Receive Data	
6	RTS	Request to Send	
7	DCD	Data Carrier Detect	
8	GND	Ground	

Connecting an ASCII Terminal as the System Console

Use the settings that Table 15 and Table 16 list when you connect a character-based (ASCII) terminal to an Origin 200 system.

Table 15. Terminal Settings for Null Modem Cable and AUX Port

Func Key	Parameter Setting		
F2	Personality = VT100	Enhance3 = ON	Font load = ON
	Comm mode = Full Duplex	End of line wrap = ON	Send ACK = ON
	Data/Printer = **AUX/MODEM**	Auto SCRL = ON	Init Tabs = OFF
	RCVD CR = CR	Monitor = OFF	Width change clear = OFF
F3			Corner Key = **HOLD**
F4	BAUD Rate = 9600	Data/stop bits = 8/1	Parity = NONE
	RCV Handshake = NONE	XMT Handshake = NONE	XPC Handshake = OFF
	AUX BAUD rate = **9600**	AUX Data/Stop Bits = **8/1**	AUX Parity = **NONE**
	AUX RCVD Handshake = **DSR**	AUX XMT Handshake = **NONE**	

Table 16. Terminal Settings for Modem Cable and Modem Port

Func Key	Parameter Setting		
F2	Personality = VT100	Enhance3 = ON	Font load = ON
	Comm mode = Full Duplex	End of line wrap = ON	Send ACK = ON
	Data/Printer = **MODEM/AUX**	Auto SCRL = ON	Init Tabs = OFF
	RCVD CR = CR	Monitor = OFF	Width change clear = OFF
F3			Corner Key = **HOLD**
F4	BAUD Rate = **9600**	Data/stop bits = **8/1**	Parity = **NONE**
	RCV Handshake = **XON/XOFF**	XMT Handshake = **XON/XOFF**	XPC Handshake = **OFF**
	AUX BAUD rate = 9600	AUX Data/Stop Bits = 8/1	AUX Parity = NONE
	AUX RCV Handshake = DSR	AUX XMT Handshake = NONE	

Connecting a PC as the System Console

As an alternative to a character-based (ASCII) terminal, you can use another computer such as a laptop IBM compatible computer or Macintosh computer.

To use an IBM compatible or Macintosh computer as the system console, you need a terminal emulation program (such as Term, HyperTerm, terminal.exe, BitCom, Procomm, ZTerm, SITcomm™, or Macterm).

You will also need a serial adapter cable (refer to Figure 3). Use the adapter cables listed in Table 17 and the serial cable provided to connect your computer to the Origin 200 server. Table 17 lists the adapter cables and part numbers.

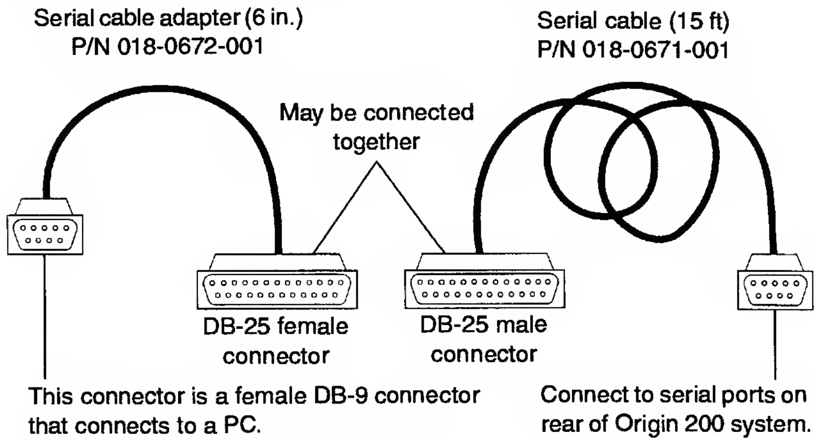
Table 17. Serial Adapter Cables for IBM and Macintosh Computers

Computer Type	Description	Length	P/N
IBM compatible	DB-25 female to DB-9 female	6 feet (2 meters)	9290134
Macintosh	DB-25 female to DIN 8 male	6 feet (2 meters)	9290135

Once you have obtained the correct adapter cable, perform the following steps to connect the computer to the Origin 200 server:

1. Make sure the Origin 200 server is turned off.
2. Locate the serial cable that was shipped with your Origin 200 server.
3. Attach the DB-9 connector to tty port 1 (refer to Figure 3 on page 28).
4. Attach the DB-25 connector of the serial cable to the DB-25 connector of the adapter cable.
5. Attach the other end of the adapter cable to one of the serial ports on your computer. You can use any available serial port on your IBM compatible or Macintosh computer. Make a note of which serial port you use. You may need this information later when you set up your terminal emulator.
6. Start your terminal emulator software and log in to the Origin 200 server using the terminal settings that Table 15 and Table 16 list.

Figure 3. Connecting an Origin 200 Server to a Laptop or PC



Note: This cable can be connected to the other serial cable. You then have a cable with a DB-9 female connector on each end. This enables you to connect an Origin 200 system to a laptop computer DB-9 serial port as shown below.

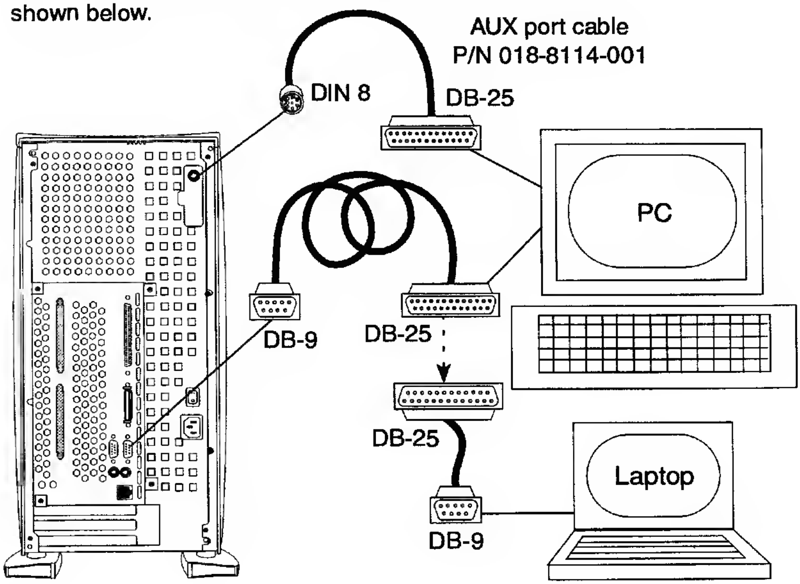


Table 18. Printer/Character (ASCII) Terminal Serial Cable Pin Assignments

Pin Number (DB-9) Female	Pin Number (DB-25)	Signal
1	20, N/C	N/C
2	2	Receive Data (RXD)
3	3	Transmit Data (TXD)
4	6,8, N/C	N/C
5	7	Ground (GND)
6	20, N/C	N/C
7	5, N/C	N/C
8	4, N/C	N/C

Modem (Full Handshake) Cable

This cable provides request-to-send (RTS) and clear-to-send (CTS) connections, and it is appropriate for a modem with RTS/CTS flow control. This cable will also work with other serial devices (except printers and terminals) that do not need RTS/CTS flow control. You can use it with a null modem adapter to attach printers and character terminals to an Origin 200 system (refer to Table 19).

Table 19. Modem Cable Pin Assignments

Pin Number (DB-9 Female)	Pin Number (DB-25)	Signal
1	8	Data Carrier Detect (DCD)
2	3	Receive Data (RXD)
3	2	Transmit Data (TXD)
4	20	Data Terminal Ready (DTR)
5	7	Ground (GND)
6	6	Data Set Ready (DSR)
7	4	Request To Send (RTS)
8	5	Clear To Send (CTS)
9	22	N/C

Origin 200 to Indy or Indigo2 Cable

This cable connects Silicon Graphics Indy and Indigo2 workstations to an Origin 200 system (refer to Table 20).

Table 20. Origin 200 to Indy and Indigo2 Serial Cable Pin Assignments

Pin Number (DB-9 Female)	Signal	Pin Number (DIN 8 Male)	Signal
1	Data Carrier Detect (DCD)	1	DTR
2	Receive Data (RXD)	3	TXD
3	Transmit Data (TXD)	5	RXD
4	Data Terminal Ready (DTR)	7	DCD
5	Ground (GND)	4	GND
6	Data Set Ready (DSR)	N/C	N/C
7	Request To Send (RTS)	2	CTS
8	Clear To Send (CTS)	6	RTS
9	N/C	N/C	N/C

Origin 200 to Onyx2 and IBM Compatible Cable

This cable connects an Origin 200 system to Onyx2 systems and IBM compatible PCs with a DB-9 serial port (refer to Table 21).

Table 21. Origin 200 to Onyx2 and IBM Serial Cable Pin Assignments

Pin Number (DB-9 Female)	Signal	Pin Number (DB-9 Female)	Signal
1	Data Carrier Detect (DCD)	9	DTR
2	Receive Data (RXD)	2	TXD
3	Transmit Data (TXD)	3	RXD
4	Data Terminal Ready (DTR)	8	DCD
5	Ground (GND)	7	GND
6, N/C	Data Set Ready (DSR)	N/C	N/C
7	Request To Send (RTS)	5	RTS
8	Clear To Send (CTS)	4	CTS
9	N/C	N/C	N/C

Origin 200 to Onyx (DB-9 to DB-9) Cable

Note that the Onyx DB-9 pin assignments are different from the Origin 200 DB-9 pin assignments (refer to Table 22).

Table 22. Origin 200 to Onyx Serial Cable Pin Assignments

Pin Number (DB-9 Female)	Signal	Pin Number (DB-9 Female)	Signal
1	Data Carrier Detect (DCD)	9	DTR
2	Receive Data (RXD)	2	TXD
3	Transmit Data (TXD)	3	RXD
4	Data Terminal Ready (DTR)	8	DCD
5	Ground (GND)	7	GND
6	Data Set Ready (DSR)	N/C	N/C
7	Request To Send (RTS)	5	CTS
8	Clear To Send (CTS)	4	RTS
9	Not connected (N/C)	N/C	N/C

Origin 200 to Macintosh (DB-9 to DIN 8) Cable

This cable connects any Macintosh computer to the Origin 200 system (refer to Table 23).

Table 23. Origin 200 to Macintosh Serial Cable Pin Assignments

Pin Number (DB-9 Female)	Signal	Pin Number (DIN 8 Male)	Signal
1	Data Carrier Detect (DCD)	8	RXD+
2	Receive Data (RXD)	3	TXD–
3	Transmit Data (TXD)	5	RXD–
		7	GPi
4	Data Terminal Ready (DTR)	6	TXD+
5	Ground (GND)	4	GND
6	Data Set Ready (DSR)	N/C	N/C
7	Request To Send (RTS)	2	HSKi
8	Clear To Send (CTS)	1	HSKo
9	Not connected (N/C)	N/C	

AUX Port Cable Pinout Assignments

The AUX port terminal cable is P/N 018-8114-001. Table 24 shows the cable pin assignments.

Table 24. AUX Port Terminal Cable Pin Assignments

Pin Number DIN-8 Connector (Male)	Pin Number DB-25 Connector (Male)	Signal
1	N/C	N/C
2	N/C	N/C
3	3	Transmit Data (TXD)
4	7	Ground (GND)
5	2	Receive Data (RXD)
6	N/C	N/C
7	N/C	N/C
8	7	Ground (GND)

The AUX port modem cable is P/N 018-8109-001. Table 25 shows the cable pin assignments.

Table 25. AUX Port Modem Cable Pin Assignments

Pin Number DIN 8 Connector (Male)	Pin Number DB-25 Connector (Male)	Signal
1	20	Data Terminal Ready (DTR)
2	5	Clear To Send (CTS)
3	2	Transmit Data (TXD)
4	7	Ground (GND)
5	3	Receive Data (RXD)
6	4	Request To Send (RTS)
7	8	Data Carrier Detect (DCD)
8	7	Ground (GND)
N/C	1	
N/C	6	

Origin 200 Vault

Refer to the "Origin 200" section on page 20 for information about the common Origin 200 ports.

This section describes the following topics:

Table 26. Origin Vault Single-ended SCSI Port Pin Assignments

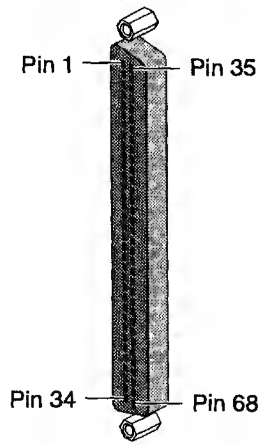
Pin	Assignment	Pin Numbers
1	Ground	
2	Ground	
3	Ground	
4	Ground	
5	Ground	
6	Ground	
7	Ground	
8	Ground	
9	Ground	
10	Ground	
11	Ground	
12	Ground	
13	Ground	
14	Ground	
15	Ground	
16	Ground	
17	TERMPWR	
18	TERMPWR	
19	SPACING	

Table 26. Origin Vault Single-ended SCSI Port Pin Assignments (continued)

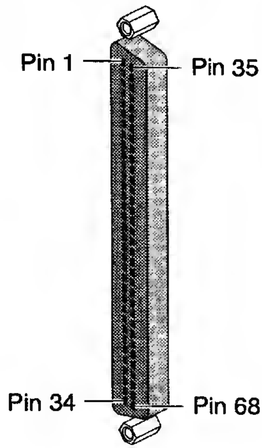
Pin	Assignment	Pin Numbers
20	Ground	
21	Ground	
22	Ground	
23	Ground	
24	Ground	
25	Ground	
26	Ground	
27	Ground	
28	Ground	
29	Ground	
30	Ground	
31	Ground	
32	Ground	
33	Ground	
34	Ground	
35	–DB(12)	
36	–DB(13)	
37	–DB(14)	
38	–DB(15)	
39	–DPARH	
40	–D0	
41	–D1	
42	–D2	
43	–D3	
44	–D4	
45	–D5	
46	–D6	
47	–D7	
48	–DPAR	
49	Ground	

Table 26. Origin Vault Single-ended SCSI Port Pin Assignments (continued)

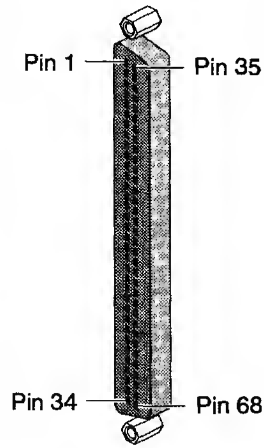
Pin	Assignment	Pin Numbers
50	Ground	
51	TERMPWR	
52	TERMPWR	
53	Reserved	
54	Ground	
55	–ATN	
56	Ground	
57	–BSY	
58	–ACK	
59	–RST	
60	–MSG	
61	–SEL	
62	–C/D	
63	–REQ	
64	–I/O	
65	–DB(8)	
66	–DB(9)	
67	–DB(10)	
68	–DB(11)	

Table 27. Origin Vault 68-Pin Differential High-Density SCSI Port Pin Assignments

Pin	Assignment	Pin Numbers
1	+DB(12)	
2	+DB(13)	
3	+DB(14)	
4	+DB(15)	
5	+DPARH	
6	+Ground	
7	+D0	
8	+D1	
9	+D2	
10	+D3	
11	+D4	
12	+D5	
13	+D6	
14	+D7	
15	+DPAR	
16	DIFFSENS	
17	TERMPWR	
18	TERMPWR	
19	Reserved	
20	+ATN	
21	Ground	
22	+BSY	
23	+ACK	
24	+RST	
25	+MSG	
26	+SEL	
27	+C/D	
28	+REQ	
29	+I/O	
30	Ground	

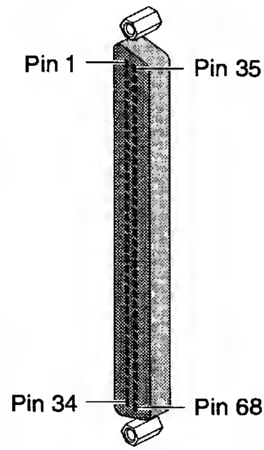


Table 27. Origin Vault 68-Pin Differential High-Density SCSI Port Pin Assignments (continued)

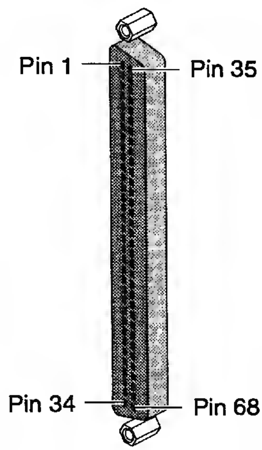
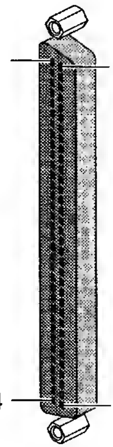
Pin	Assignment	Pin Numbers
31	+DB(8)	
32	+DB(9)	
33	+DB(10)	
34	+DB(11)	
35	–DB(12)	
36	–DB(13)	
37	–DB(14)	
38	–DB(15)	
39	–DPARH	
40	–Ground	
41	–D0	
42	–D1	
43	–D2	
44	–D3	
45	–D4	
46	–D5	
47	–D6	
48	–D7	
49	–DPAR	
50	Ground	
51	TERMPWR	
52	TERMPWR	
53	Reserved	
54	–ATN	
55	Ground	
56	–BSY	
57	–ACK	
58	–RST	
59	–MSG	
60	–SEL	

Table 27. Origin Vault 68-Pin Differential High-Density SCSI Port Pin Assignments (continued)

Pin	Assignment	Pin Numbers
61	–C/D	
62	–REQ	
63	–I/O	
64	Ground	
65	–DB(8)	
66	–DB(9)	
67	–DB(10)	
68	–DB(11)	

Origin 2000 Deskside

This section describes the following topics:

Serial Ports

Figure 4 shows the Origin 2000 deskside system serial port (Console tty_1). Table 9 (on page 22) lists the serial port connector pin assignments in RS-232 mode. Table 10 (on page 22) lists the serial port connector pin assignments in RS-422 mode.

Figure 4. Origin 2000 Deskside Serial Port Connections

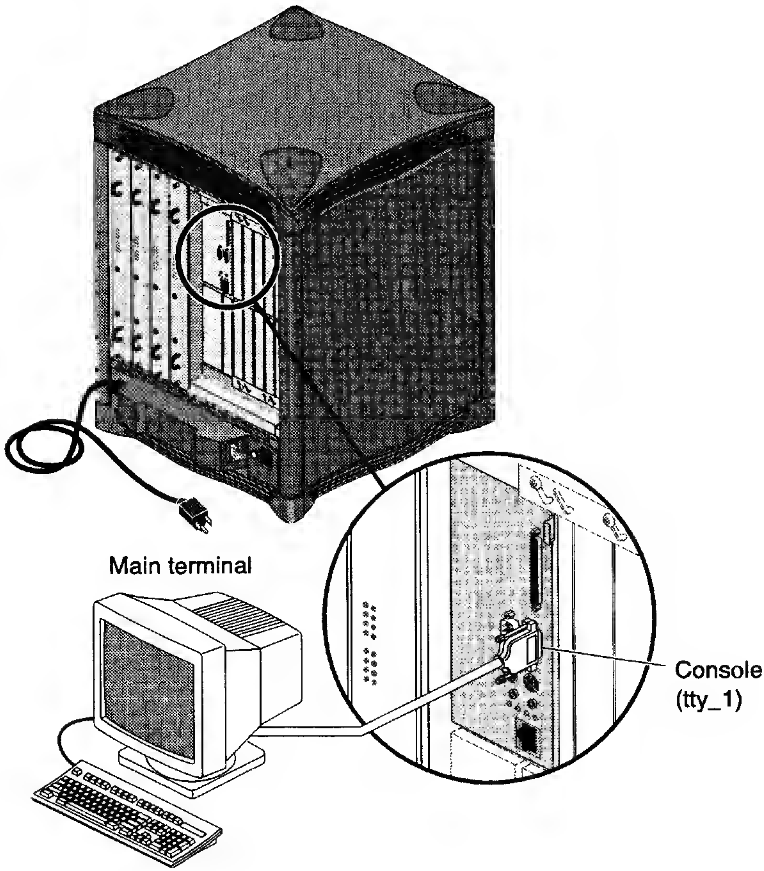


Table 28. Origin 2000 Deskside Serial Port Pin Assignments, RS-232 Mode

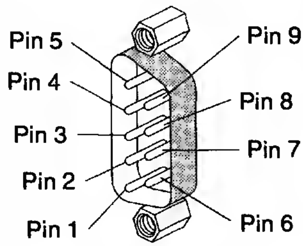
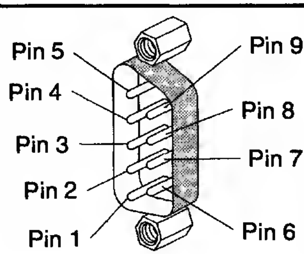
Pin	Assignment	Pin Numbers
1	Data Carrier Detect (DCD)	
2	Receive Data (RD)	
3	Transmit Data (TD)	
4	Data Terminal Ready (DTR)	
5	Signal Ground (SG)	
6	Data Set Ready (DSR)	
7	Request To Send (RTS)	
8	Clear To Send (CTS)	
9	Not Connected	

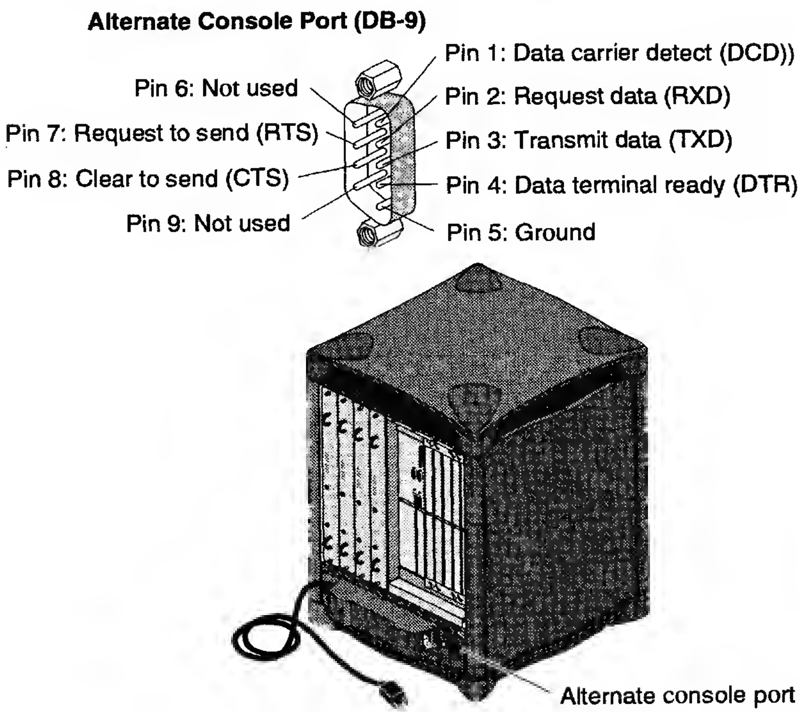
Table 29. Origin 2000 Deskside Serial Port Pin Assignments, RS-422 Mode

Pin	Assignment	Pin Numbers
1	Reserved	
2	Receive Data (RXD –)	
3	Transmit Data (TXD –)	
4	Transmit Data (TXD +)	
5	Signal Ground (GND)	
6	Receive Data (RXD +)	
7	Output Handshake (HSKo)	
8	Input Handshake (HSKi)	
9	(Reserved)	

Alternate Console Port

The alternate console port (ACP) is located at the lower-right of the rear of the module and is a DB-9 male connector. This is a parallel connection to the module system controller (MSC) Diagnostic port, which is a DIN 8 female connector that is located on the front of the module.

Figure 5. Origin 2000 Deskside Alternate Console Port

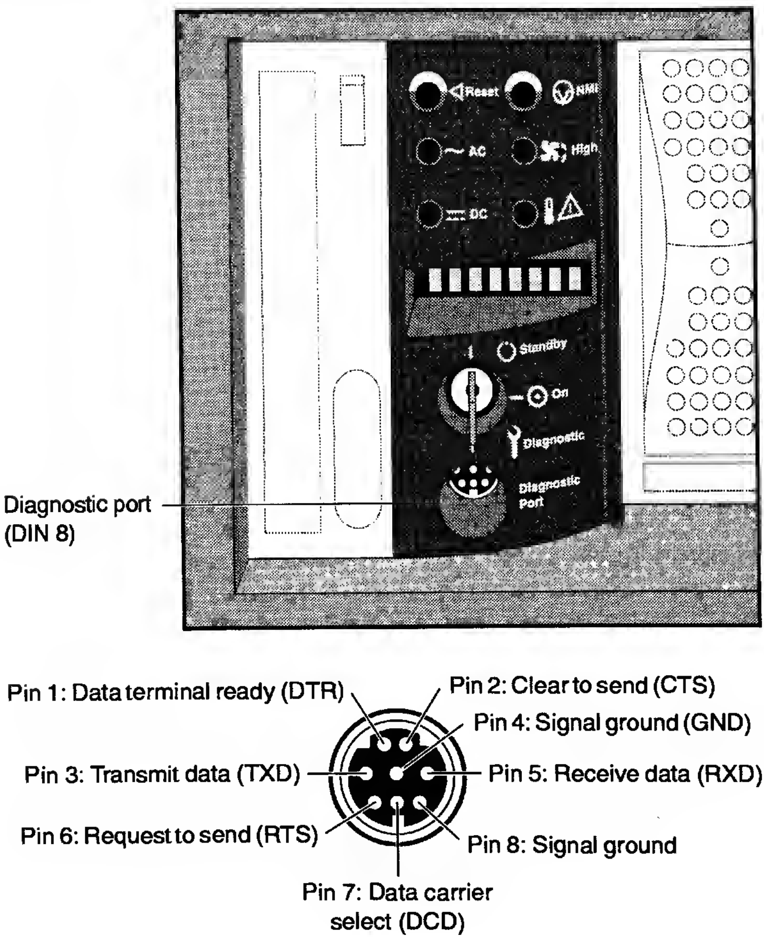


Diagnostic Port

The Diagnostic port on the MSC is a DIN 8 female connector. This connector is electrically the same as the ACP connector.

Note: The ACP and Diagnostic port inputs are exclusively ORed. Connecting a cable to the ACP *and* the Diagnostic port results in no output.

Figure 6. Origin 2000 Deskside Diagnostic Port



Connecting a Wyse Terminal to an Origin 2000 Deskside as the System Console

There are two cables available for the Wyse terminal to BaseIO tty port 1 connection (Console port), which is a DB-25 to DB-9 connection.

One cable is a null modem cable with a male DB-9 connector and the other cable is a straight serial cable with a female DB-9 connector. The male connector on the null modem cable requires a female adapter. You can use either cable. Be sure that you have the correct ASCII terminal port connected and the correct settings selected as listed in Table 15 and Table 16.

The null modem cable (P/N 018-0230-002) goes from the AUX port on the Wyse terminal (using the settings in Table 15) to the BaseIO serial tty port 1 connector. This cable requires a female adapter, because both the cable connector and the BaseIO connector are male. The straight cable (P/N 018-0671-001) goes from the MODEM port on the Wyse terminal (using the settings listed in Table 16) to the BaseIO serial tty port 1 connector.

Connecting any ASCII terminal to the alternate console port (ACP) is the same as connecting the ASCI terminal to the MSC. Use the same two cables and terminal settings as listed in Table 15 and Table 16.

Connecting an ASCII terminal to the Diagnostic port (DIN 8 female connector) on the front of the MSC uses the same two cables but opposite terminal settings and connections. The null modem cable uses the MODEM terminal port and the settings listed in Table 16, and the straight cable uses the AUX terminal port and the settings listed in Table 15.

Connecting an Indy Workstation to an Origin 2000 Deskside as the System Console

You can also connect an Indy workstation to any of these connectors on the Origin 2000 deskside system. The Indy workstation serial port provides a DIN 8 female connector. You can use the male DIN 8 cable that comes with the Origin series system, which is a null modem cable. You can connect this cable directly to the Diagnostic Port, and you connect it to the Console Port (BaseIO DB-9 connector) with a female DIN 8 to female DB-9 converter.

Alternatively, you can use another supplied male DIN 8 to female DB-9 cable to connect to the BaseIO serial port. These are both null modem cables.

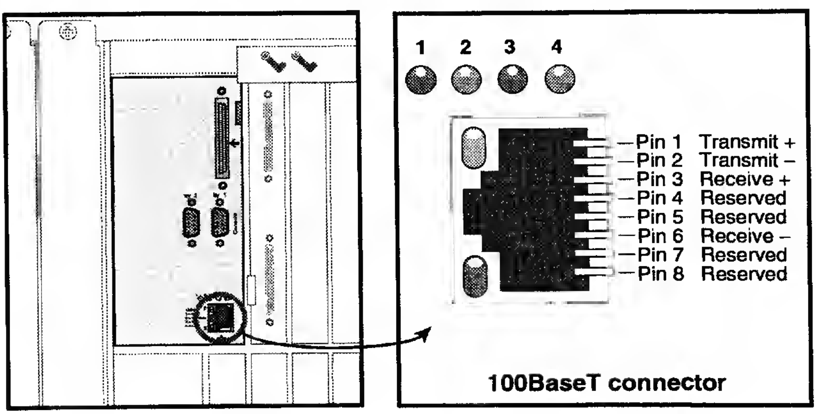
Standard 100BaseT Ethernet Port LEDs

There are two LEDs on the RJ-45 Ethernet port. The top (green) LED illuminates only when the system is transmitting. The bottom (yellow) LED illuminates whenever it senses any packet on the wire, including packets not destined for your system.

The four LEDs above the RJ-45 Ethernet connector have the following functions:

- The yellow LED on the far left (LED 1) illuminates to indicate SCSI activity on the BaseIO single-ended SCSI connector.
- The green LED (LED 2) illuminates to indicate 100 MB-per-second packet activity.
- The yellow LED on the right (LED 3) illuminates to indicate when the Ethernet is operating at full duplex rates of transfer or receive.
- The green LED on the far right (LED 4) shows the Ethernet link test. It illuminates when the link state is valid.

Figure 7. Origin 2000 Deskside 100BaseT Ethernet Port Pin Assignments



Standard SCSI Connector

A single, external 68-pin SCSI connector is provided on the BaseIO panel (refer to Figure 8 and Table 30). This connector supports both Ultra SCSI and SCSI-2 devices. The connector sends single-ended SCSI signals only. Optional additional SCSI ports can be implemented using XIO option boards. The dash (–) preceding a signal name indicates that the signal is low.

Note: 8-bit devices that connect to the P cable leave the following signals open: –DB(8), –DB(9), –DB(10), –DB(11), –DB(12), –DB(13), –DB(14), –DB(15), –DB(P1). All other signals are connected as shown in Table 30.

Figure 8. Origin 2000 Deskside SCSI Connector Pin Assignments

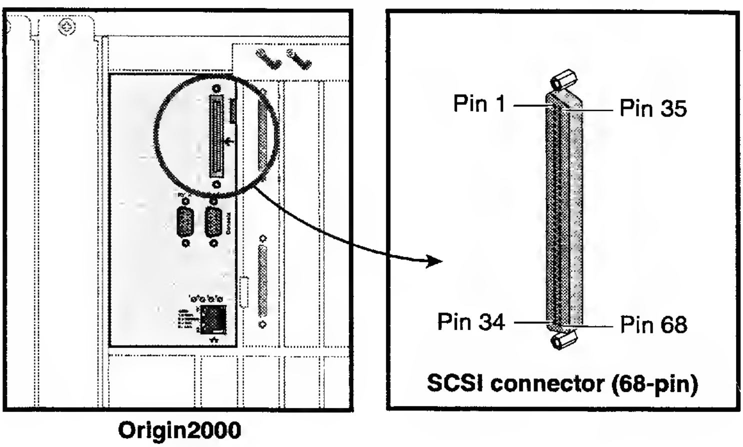


Table 30. Origin 2000 Deskside SCSI Port Pin Assignments

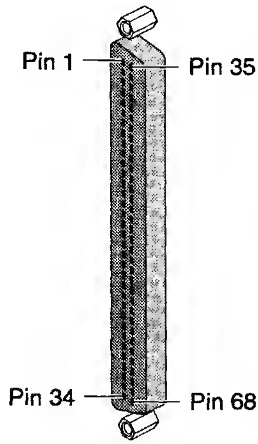
Pin	Assignment	Pin Numbers
1	Ground	
2	Ground	
3	Ground	
4	Ground	
5	Ground	
6	Ground	
7	Ground	
8	Ground	
9	Ground	
10	Ground	
11	Ground	
12	Ground	
13	Ground	
14	Ground	
15	Ground	
16	Ground	
17	TERMPWR	
18	TERMPWR	
19	Spacing	
20	Ground	
21	Ground	
22	Ground	
23	Ground	
24	Ground	
25	Ground	
26	Ground	
27	Ground	
28	Ground	
29	Ground	
30	Ground	
31	Ground	

Table 30. Origin 2000 Deskside SCSI Port Pin Assignments (continued)

Pin	Assignment	Pin Numbers
32	Ground	
33	Ground	
34	Ground	
35	–DB(12)	
36	–DB(13)	
37	–DB(14)	
38	–DB(15)	
39	–DPARH	
40	–D0	
41	–D1	
42	–D2	
43	–D3	
44	–D4	
45	–D5	
46	–D6	
47	–D7	
48	–DPAR	
49	Ground	
50	Ground	
51	TERMPWR	
52	TERMPWR	
53	Reserved	
54	Ground	
55	–ATN	
56	Ground	
57	–BSY	
58	–ACK	
59	–RST	
60	–MSG	
61	–SEL	
62	–C/D	

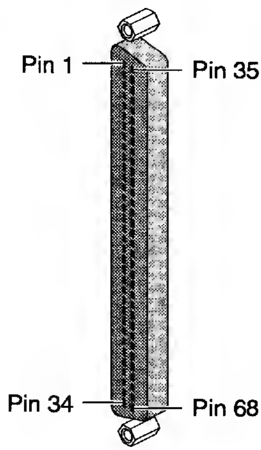
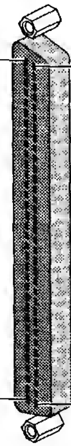


Table 30. Origin 2000 Deskside SCSI Port Pin Assignments (continued)

Pin	Assignment	Pin Numbers
63	–REQ	
64	–I/O	
65	–DB(8)	
66	–DB(9)	
67	–DB(10)	
68	–DB(11)	

Origin 2000 and Onyx2 Rack

Note: The Origin 2000 rack maintenance ports are identical to the Origin 2000 deskside ports that are described starting on page 39.

This section describes the following maintenance cabling topics:

Each rack has a multimodule system controller (MMSC) with 6 ports; all are female DIN 8 connectors. The labeling may differ with the different revisions of the MMSC. One revision labels the ports COM 1 through 6; the other revision labels them as CONSOLE, UPPER BAY, LOWER BAY, BASE IO, and ALTERNATE CONSOLE PORT, respectively. The sixth port is not used.

Refer to Table 31 on page 56 for a description of each port.

The Indy workstations and ASCII terminals are normally connected to thc MMSC, which, in turn, is cabled to the modules in the rack. For isolation purposes, you can cable directly to the module using the same connections as for the Origin 2000 deskside system.

All the cables supplied with the rack system are null modem and are either DIN 8 to DIN 8 (P/N 018-8223-001) or DIN 8 to DB-9 (P/N 018-0644-001). Of course, other options are available such as DIN 8 to DIN 8 with a DIN 8 to DB-9 converter.

Connecting a Terminal to a Rack as the System Console

To connect the ASCII terminal to the MMSC, you need a DB-25 to DIN 8 connection, and you must configure the connection with either a null modem or straight cable to DB-9, male or female, respectively, using a DB-9 to DIN 8 converter.

To connect the terminal to the rack system, perform the following steps:

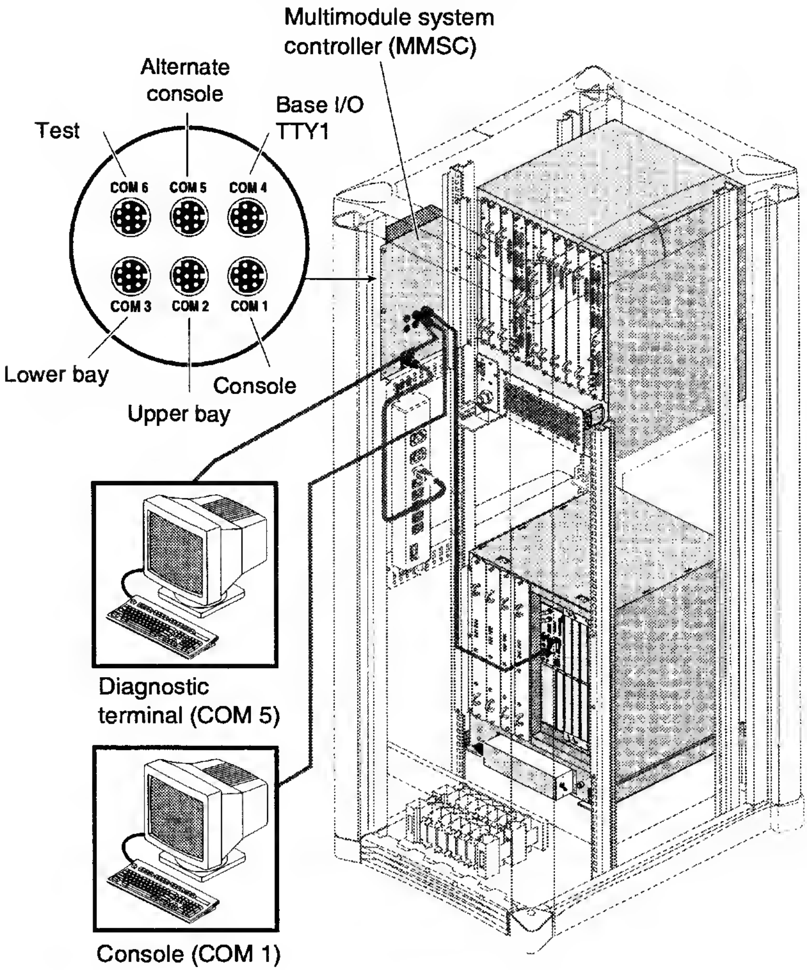
1. Connect the tty_1 (CONSOLE) port on the BaseIO to the BASE IO (COM 4) port on the multimodule system controller (MMSC) using a DIN 8-to-DB-9 cable (refer to Figure 9). This enables remote access to the BaseIO console port through the MMSC.

   Note: You can connect a terminal directly to the tty_1 (Console) port on the BaseIO board using a DB-9 to DB-25 PC cable; however, you will not be able to access this port remotely as a result.

2. Connect the COM 1 (CONSOLE) connector on the MMSC to the console or terminal device using a DB-9 to DB-25-pin PC cable with the DIN 8-to-DB-9 converter; refer to Figure 9.

3. Connect a diagnostic terminal to the COM 5 (ALTERNATE CONSOLE) port using a DB-9 to DB-25 PC cable with the DIN 8-to-DB-9 converter; refer to Figure 9.

Figure 9. Connecting a Terminal to a Rack System



Connecting the Rack MSCs to the MMSC

The rack system requires that the single module system controller (MSC) serial cable must have a DB-9 (female) connector on one end and a DIN 8 (male) connector on the other end (P/N 018-0644-001).

To connect the MSC(s) to the MMSC and verify that the multimodule display is connected to the MSC(s), perform the following steps:

1. Go to the rear of the chassis and plug the serial cable into the DB-9 connector near the module power switch (refer to Figure 10).
2. Plug the other end of the serial cable into the MMSC (LOWER BAY or UPPER BAY) connector as applicable.
3. Plug the second serial cable from the second MSC into the MMSC, as required.
4. After you connect the terminals and MSCs to the MMSC, your cabling scheme should resemble the one that Figure 11 depicts.

   Refer to Table 31 for a description of each MMSC serial port.

Figure 10. Connecting the Rack MSCs to the MMSC

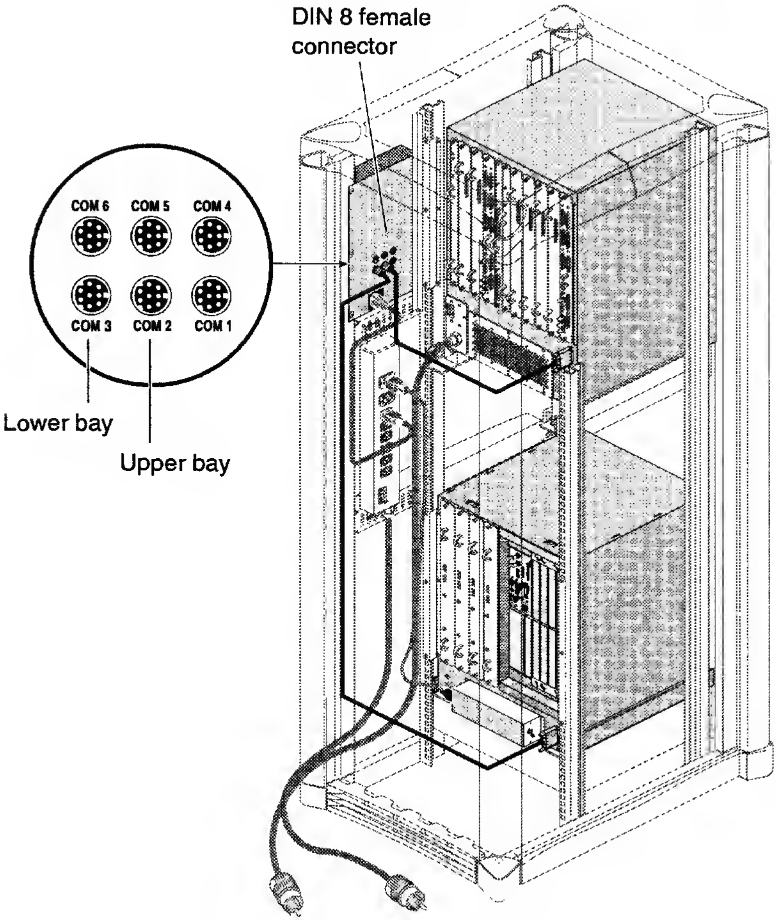


Figure 11. All Connections to the Rack MMSC

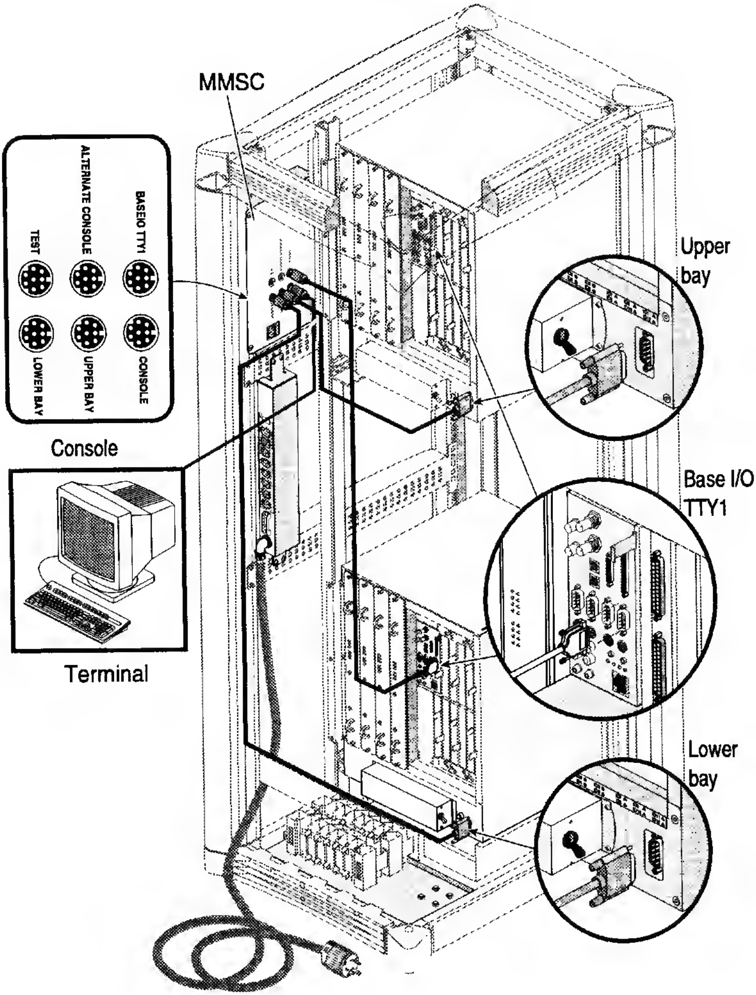


Table 31. Origin 2000 Rack MMSC Port Descriptions

Port	Function	Cabling
COM1 CONSOLE	Terminal device or console connects to this port. In a multirack system, this port is typically used only with the system that has the master BaseIO.	Normally requires a DIN 8 to DB-9 converter, along with a DB-9 to DB-25 PC-style serial cable. The part numbers of the two cables that are required to connect a serial console with a DB-25 port to COM 1 are 018-8104-001 (DIN 8 to DB-9) and 018-0230-002 (DB-9 to DB-25).
COM2 UPPER BAY	Connection to the MSC in the upper bay of the Origin 2000 rack. In a multirack system, this port is used on every MMSC.	Cable provided with system, or DB-9 to DIN 8 null modem cable. The part number of the connecting cable is 018-0644-001.
COM3 LOWER BAY	Connection to the MSC in the lower bay of the Origin 2000 rack. In a multirack system, this port is used on every MMSC.	Cable provided with system, or DB-9 to DIN 8 null modem cable. The part number of the cable is 018-0644-001.
COM4 BASE I/O TTY	Connection to the master BaseIO board tty_1 (console port). In a rack configuration with two BaseIO boards (an upper and a lower board) the master BaseIO is typically the one in the lower rack. In a multirack system, this port is typically used only with the module that has the master BaseIO board.	Cable provided with system, or standard DB-9 to DIN 8 null modem cable. The part number of the cable between the COM 4 port and the ttyd1 port on the master BaseIO board is 018-0644-001.
COM5 ALTERNATIVE CONSOLE	This is the remote service port; it is not a general-purpose modem port. Also used for direct firmware downloads in emergencies.	Normal DIN 8 to DIN 8 null modem cable. RAT uses this port to communicate with the MMSC. The cable part number depends on the type of connection that is used.

Table 31. Origin 2000 Rack MMSC Port Descriptions (continued)

Port	Function	Cabling
COM 6 TEST	This port is the MMSC debugging port; not used in customer systems.	N/A
DISPLAY	The display port is cabled to the MMSC display unit. Note that only one display is used per system configuration, not one display per rack.	Refer to Figure 12 on page 58.
MULTI FFSC (for Ethernet)	This port is a private Ethernet connection that is used only for communication between MMSCs. Do not connect it to a customer's local network for any reason.	Refer to "Connecting Multiple MMSCs" on page 59.

Table 32 lists the COM port attributes.

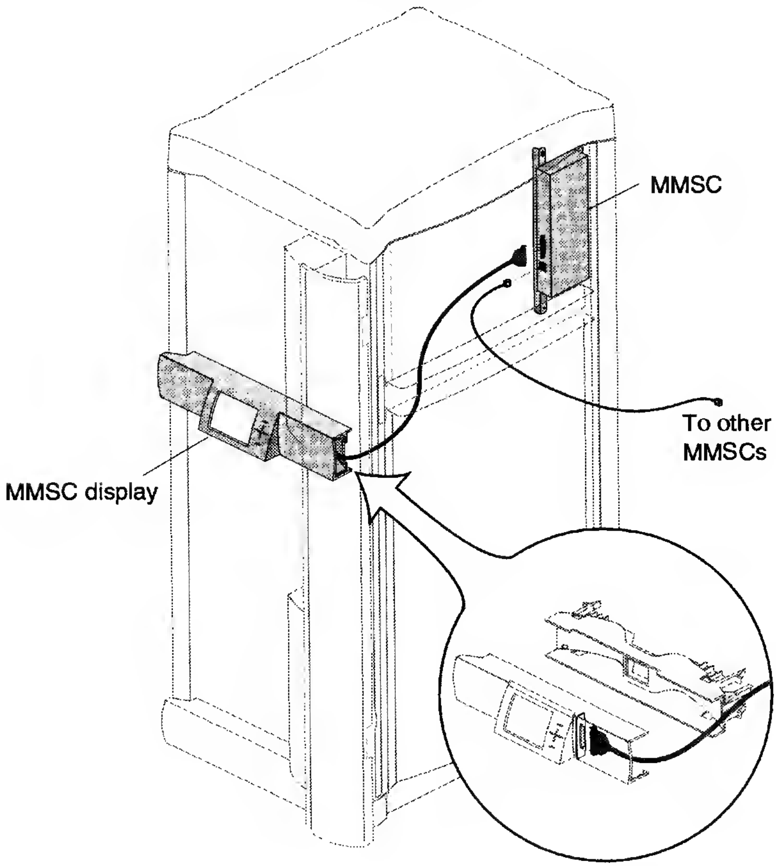
Table 32. COM Port Attributes

Port	speed	func	cmd	oob	rxbuf	txbuf	hwflow
COM 1	9600	TERMINAL	Y	N	4096	65536	N
COM 2	9600	UPPER	N	N	16384	4096	N
COM 3	9600	LOWER	N	N	16384	4096	N
COM 4	9600	SYSTEM	Y	Y	16384	4096	N
COM 5	9600	ALTCONS	Y	N	4096	4096	N
COM 6	9600	DEBUG	N	N	4096	4096	N

Connecting the MMSC Display

The MMSC display connects to the MMSC through a video cable that routes through the side of the chassis (refer to Figure 12).

Figure 12. Connecting the Rack MMSC Display



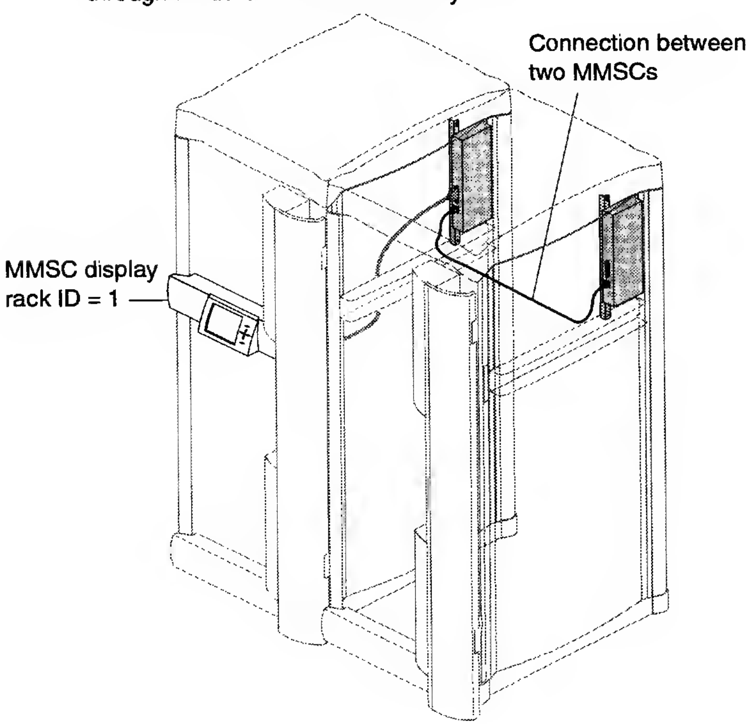
Connecting Multiple MMSCs

- Every Origin 2000 rack must have a unique ID
- Each rack has an MMSC (multimodule system controller)
- The rack number is stored in the MMSC NVRAM
- The MMSC `rackid` command changes the rack number
- You must install the MMSC display in rack 1 (R1) or MetaRouter
- Number the racks from left to right in ascending order as shown in Figure 68 on page 149.

Figure 13. Connecting Multiple MMSCs

MMSCs connect via a private Ethernet network and use special null-crossover cables (P/N 018-0625-001).

For more than two racks, you must connect the MMSCs through an Ethernet HUB assembly.



Onyx2 Deskside

The primary device for interconnection is the IO6G. Refer to the "IO6G Cabling and Pin Assignments" on page 105 for information about the Onyx2 deskside maintenance cabling and connector pin assignments.

Onyx2 InfiniteReality

The primary device for interconnection is the IO6G. Refer to "IO6G Cabling and Pin Assignments" on page 105 and Figure 11 on page 55 for information about the Onyx2 InfiniteReality rack maintenance cabling and connector pin assignments.

MetaRouter

This section describes the following topics:

MMSC-to-Ethernet Hub Connections

Origin 2000 systems with more than 2 racks require an Ethernet Hub to interconnect the MMSCs. For example, a 9-rack Origin 2000 system uses 3 Hubs to interconnect all 9 racks in the system.

Figure 14 illustrates the MetaRouter Hub connections (P/N 9980995). Figure 15 illustrates the MetaRouter Hub connections (P/N 9470235).

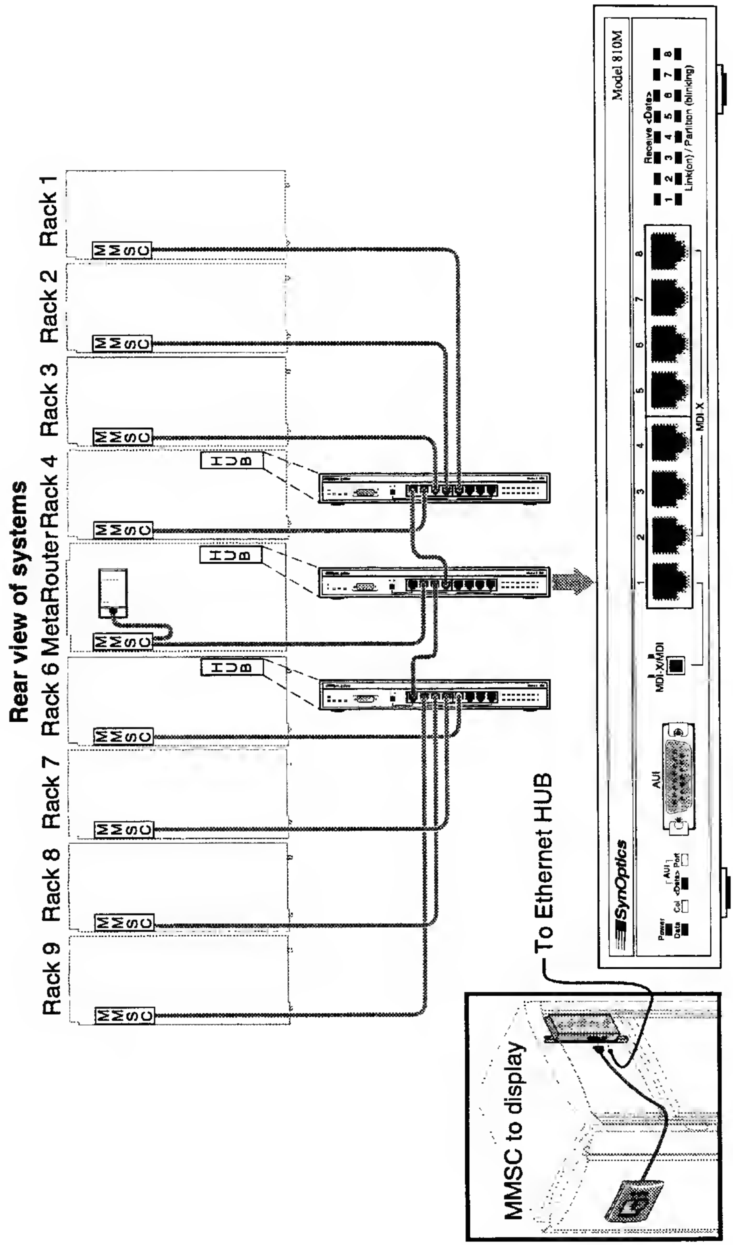


Figure 14. MetaRouter Hub Connections (P/N 9980995)

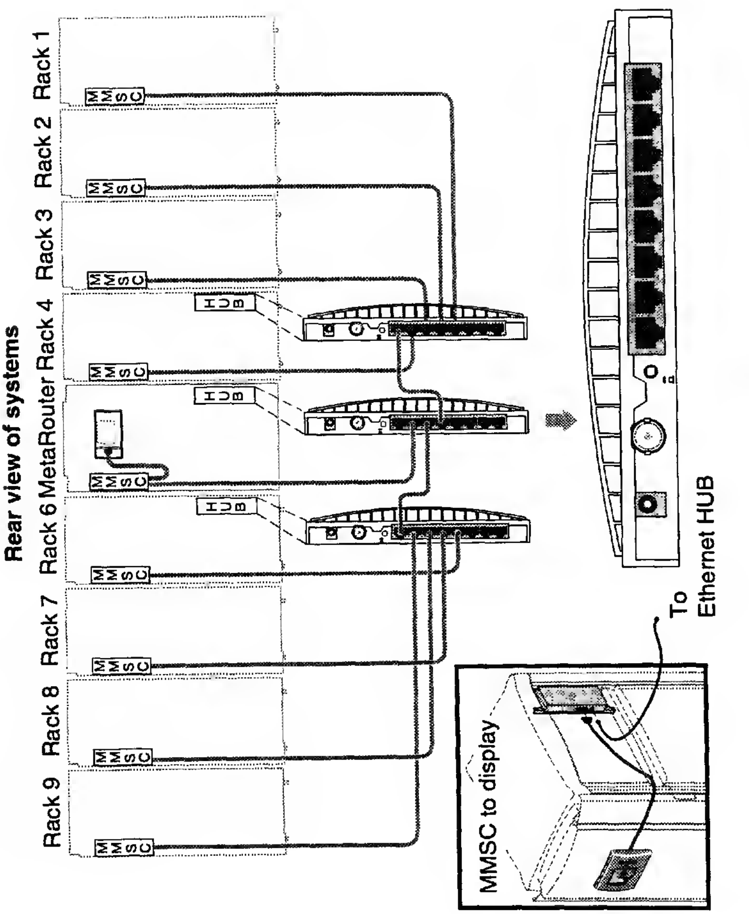
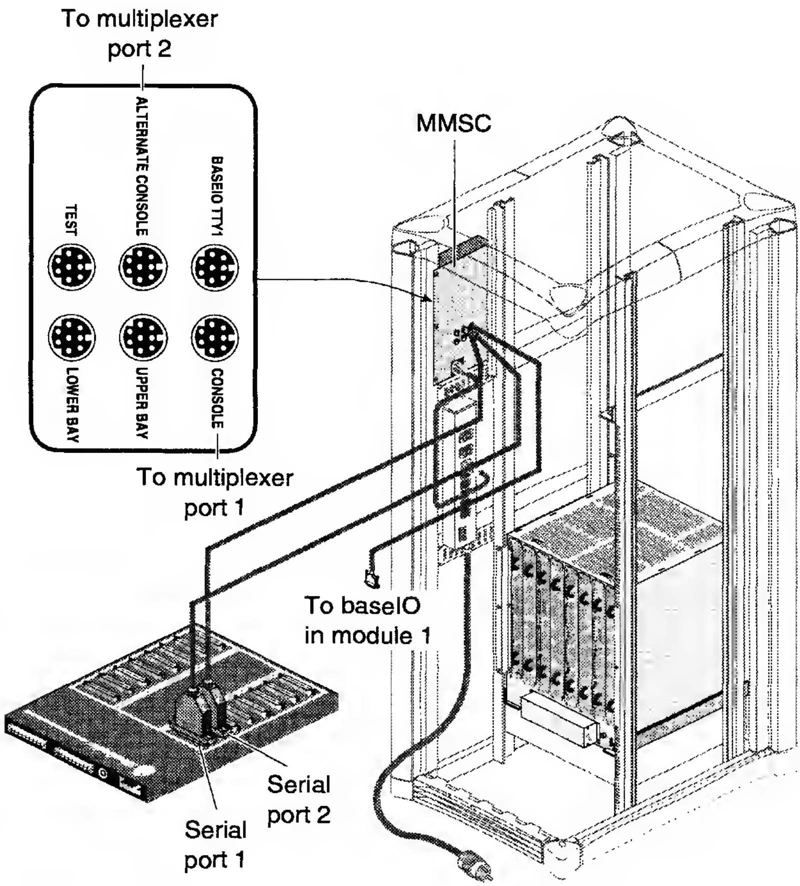


Figure 15. MetaRouter Hub Connections (P/N 9470235)

MetaRouter-to-Serial Multiplexer Connections

As shown in Figure 16, the serial multiplexer connects to the CONSOLE and ALTERNATE CONSOLE ports of the MetaRouter rack MMSC.

Figure 16. MetaRouter Serial Multiplexer Console Connections



System Cabling and Board Configurations

This section contains cabling and board configuration information for Origin and Onyx2 systems. It includes the following topics:

IRIX Cabling Information Commands

Use the following IRIX commands to display information about the system cabling configuration.

- `topology` - Returns the system hardware graph
- `linkstat` - Returns router traffic and status information. Refer to "Using the linkstat Command" on page 248 for an example of how to run this command.
- `xbstat` - Returns XBOW traffic and status information. Refer to "Using the xbstat Command" on page 249 for an example of how to run this command.

Origin 2000 Systems

For information about cabling Origin systems larger than 64 processors, refer to *32- to 128-processor Upgrade Procedure*, publication 108-0188-002 and *128- to -256-processor Upgrade Procedure*, publication number 108-0229-001. This section contains cabling information for the following Origin 2000 system configurations:

Figure 17. 16-processor System Cabling

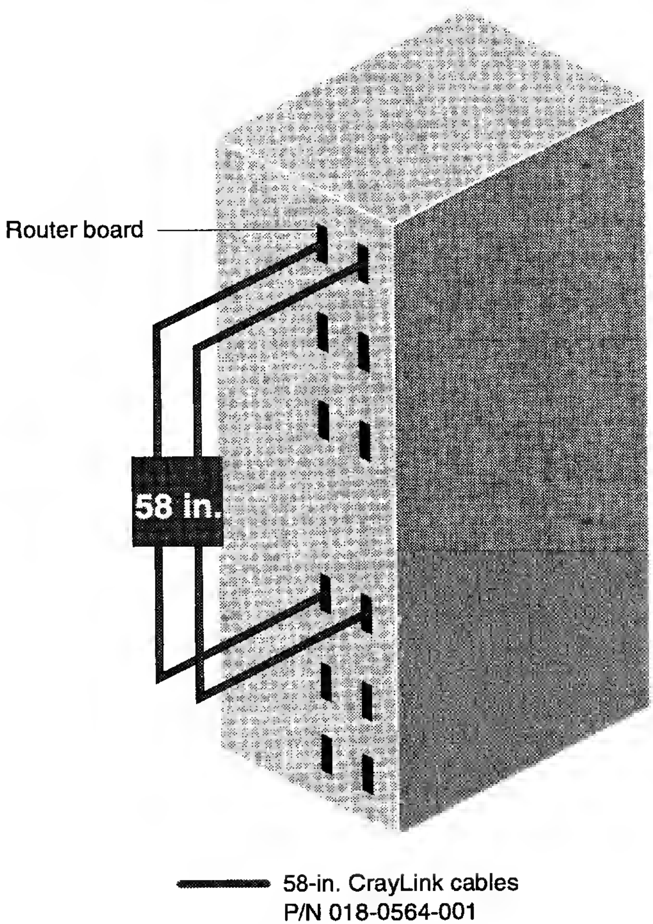


Figure 18. 16-processor System Cable Routing

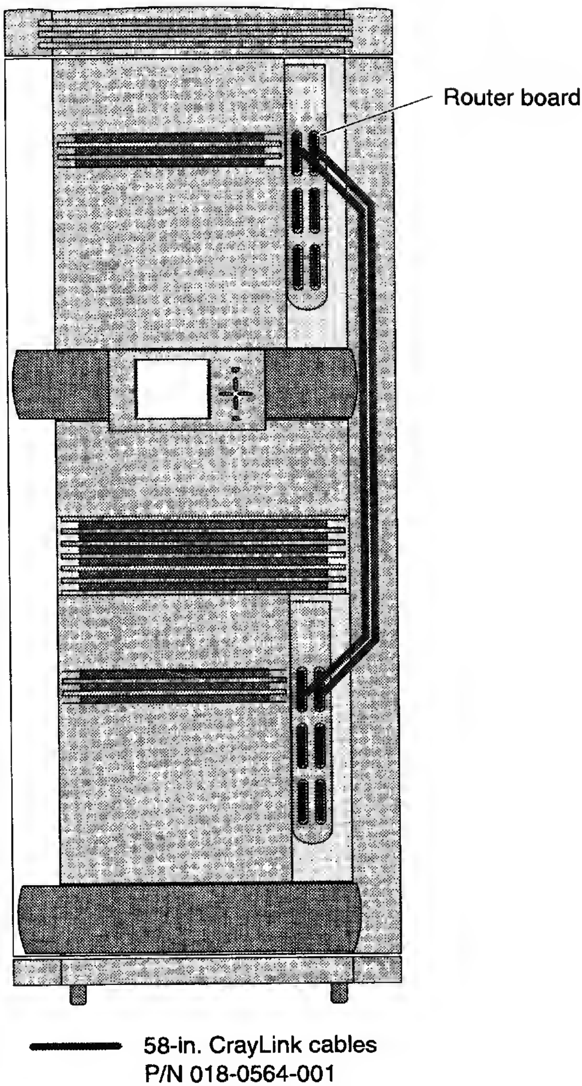
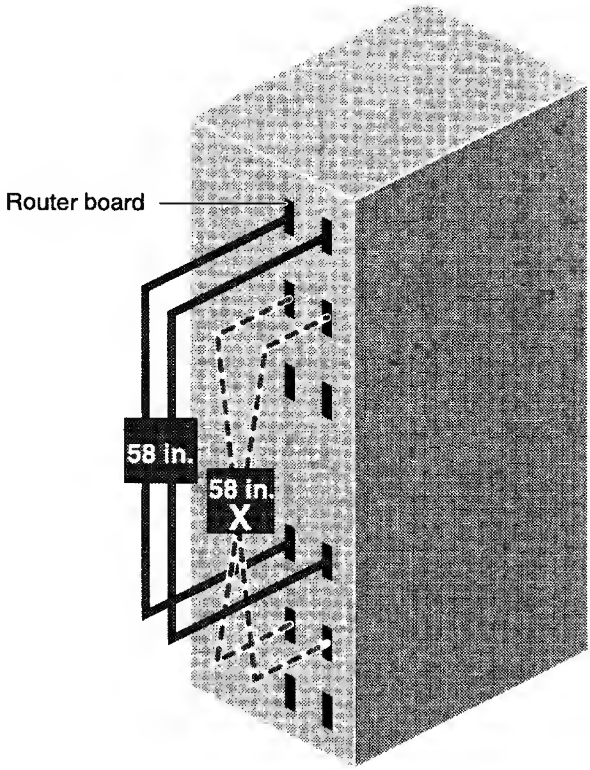


Figure 19. 16-processor System with Xpress Links Cabling



Xpress link cables

58-in. CrayLink cables
P/N 018-0564-001

Figure 20. 16-processor System with Xpress Links Cable Routing

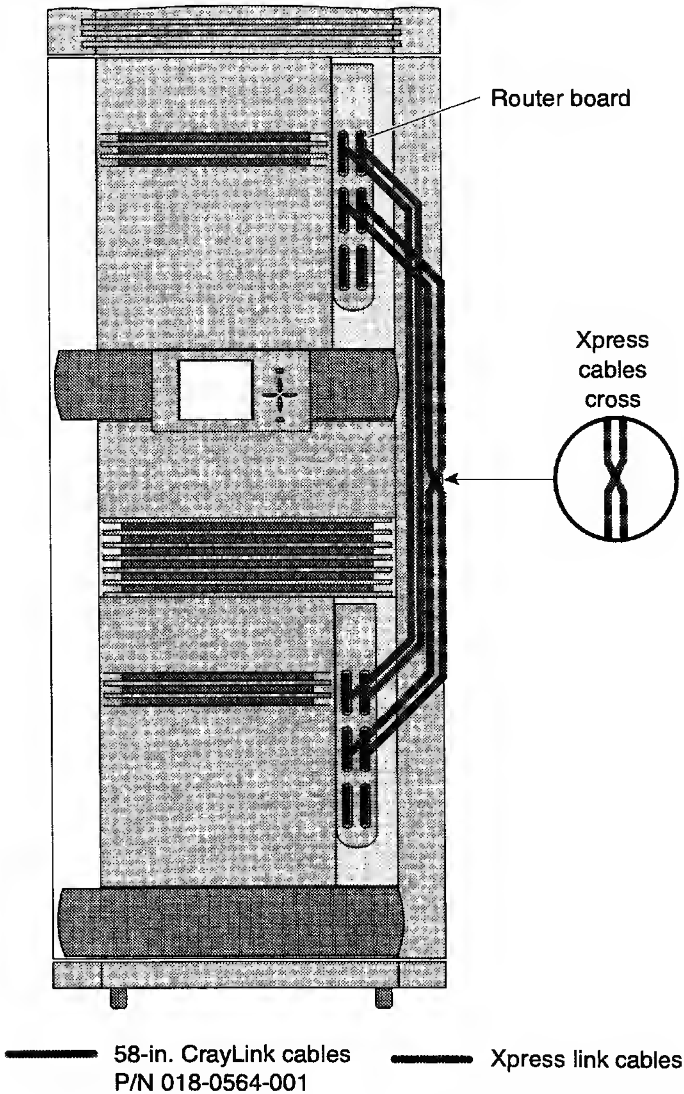


Figure 21. 24-processor System with Xpress Links Cabling

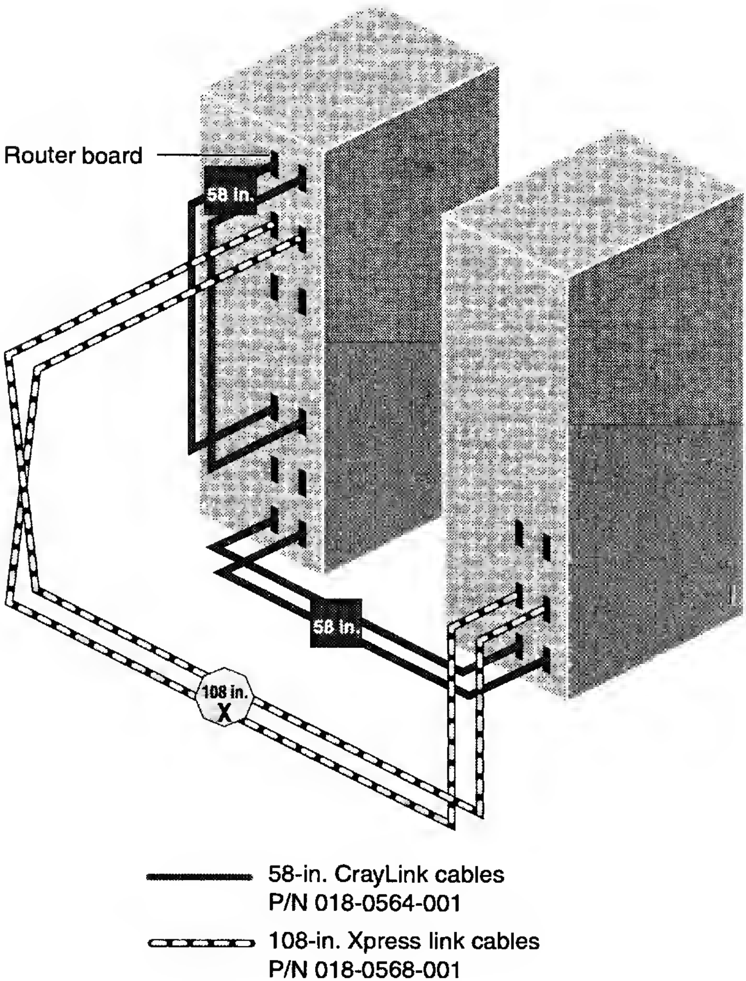


Figure 22. 24-processor System with Xpress Links Cable Routing

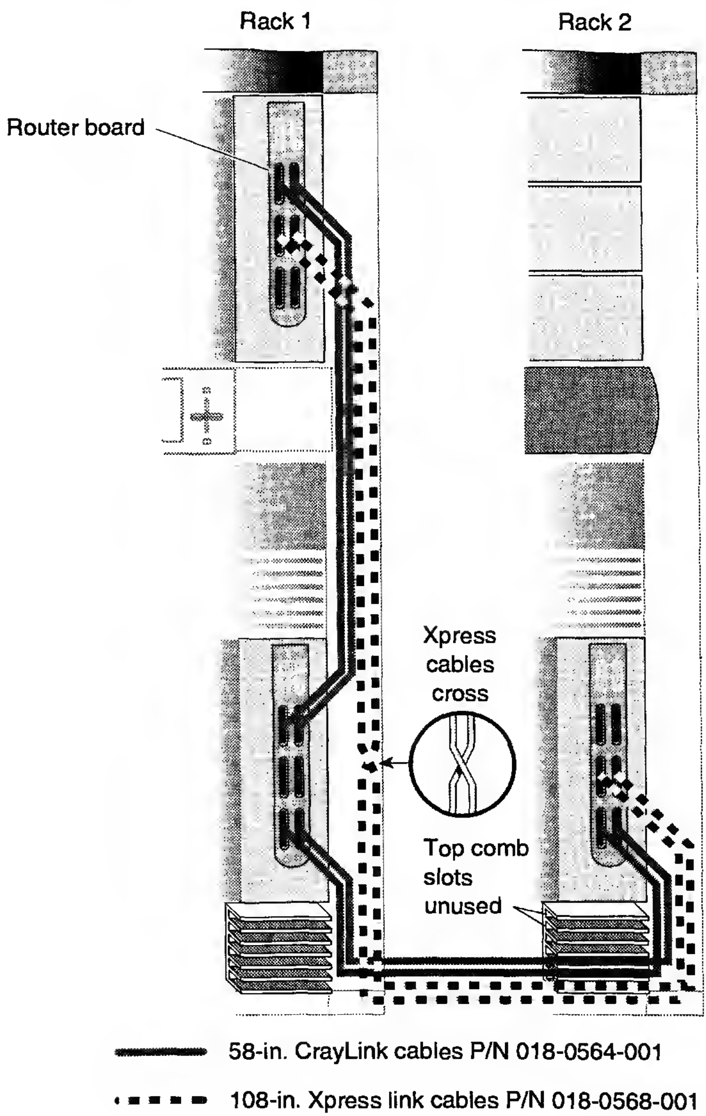


Figure 23. 32-processor System Cabling

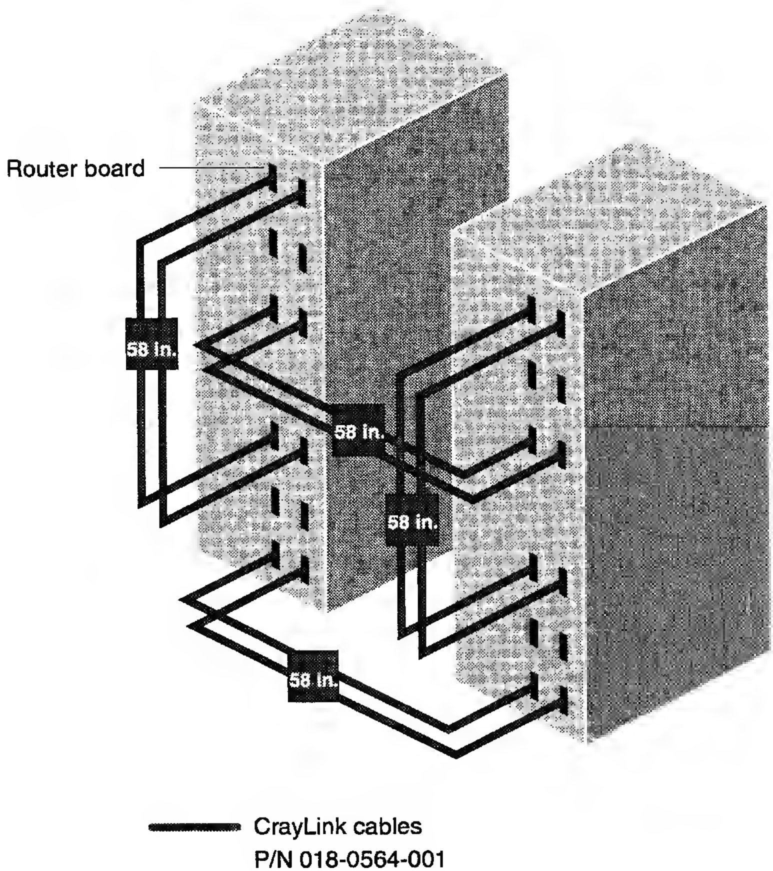


Figure 24. 32-processor System Cable Routing

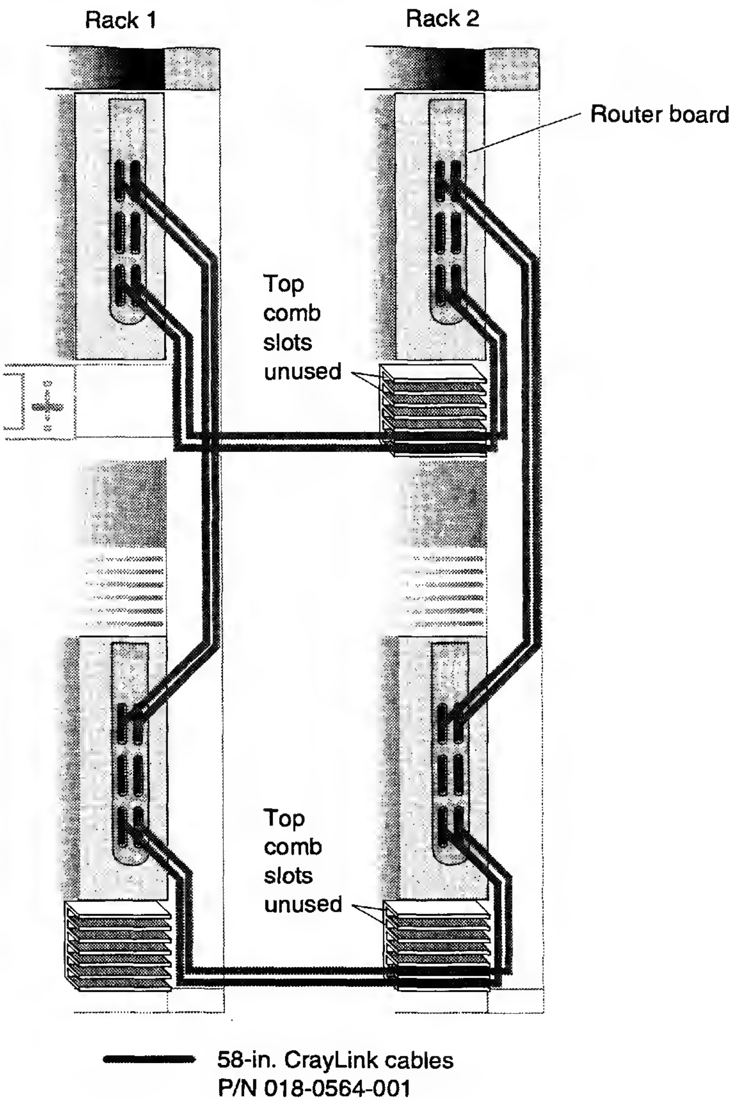


Figure 25. 32-processor System with Xpress Links Cabling

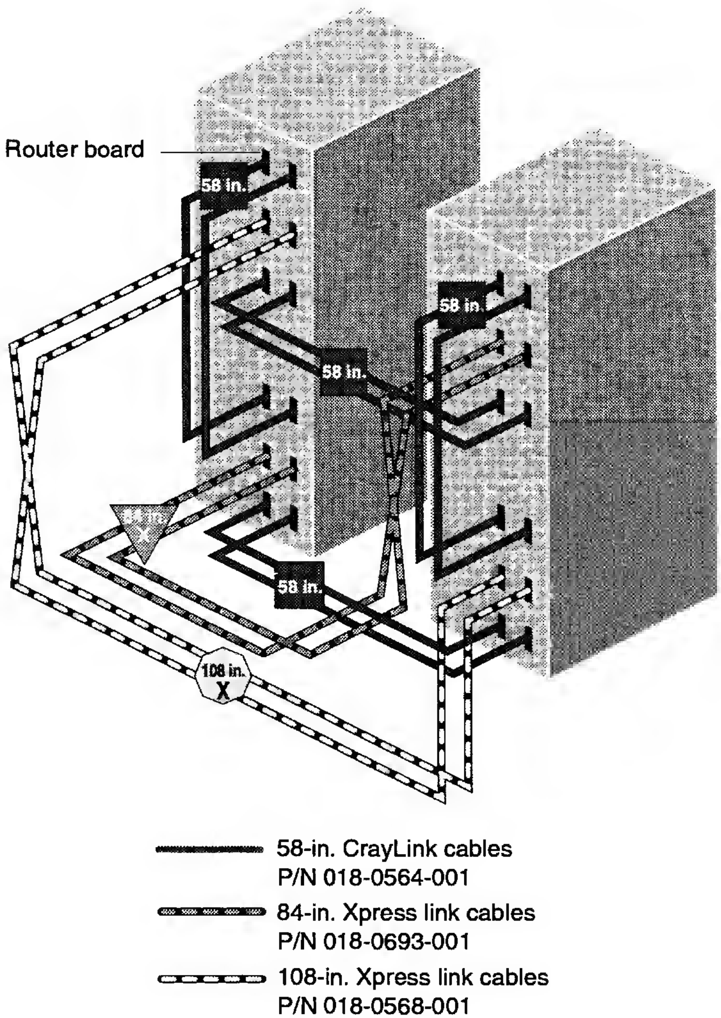


Figure 26. 32-processor System with Xpress Links Cable Routing

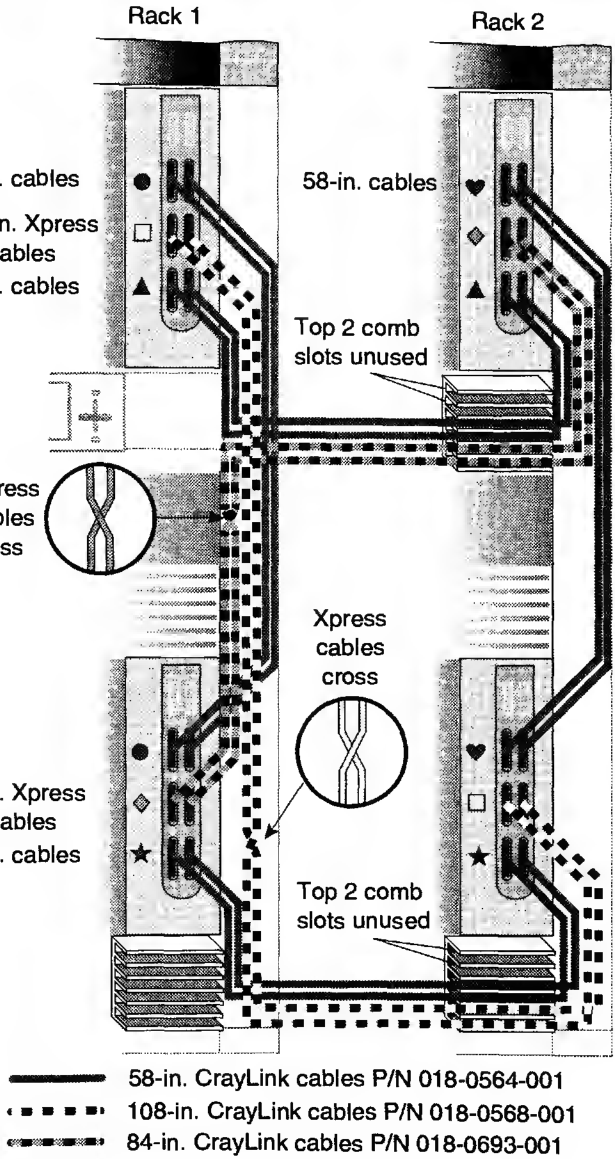


Figure 27. 40-processor System Cabling

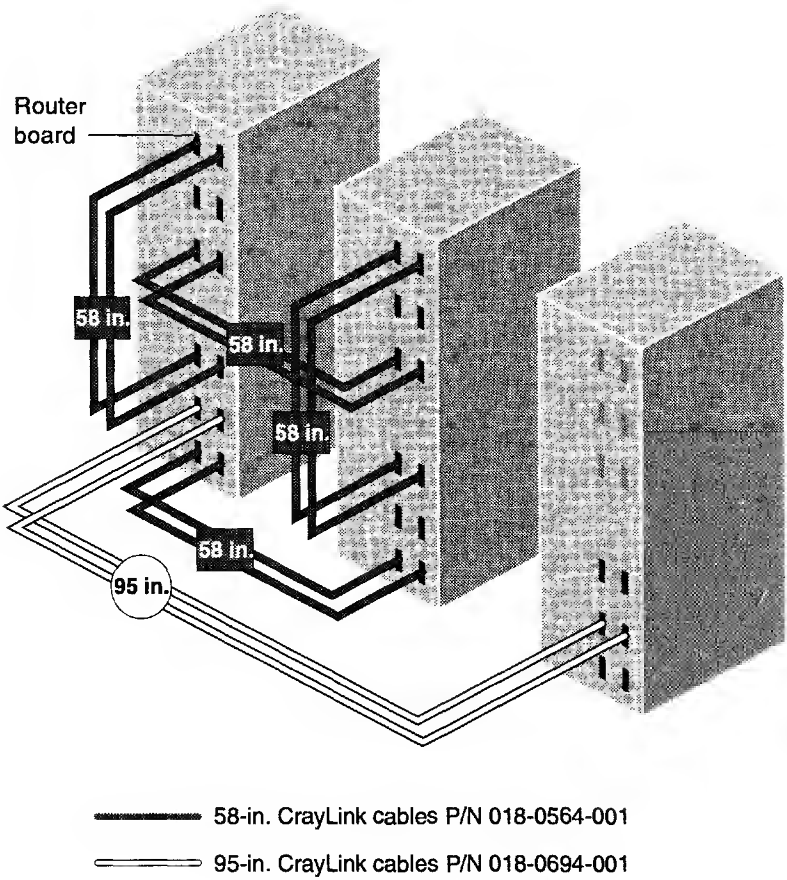


Figure 28. 40-processor System Cable Routing

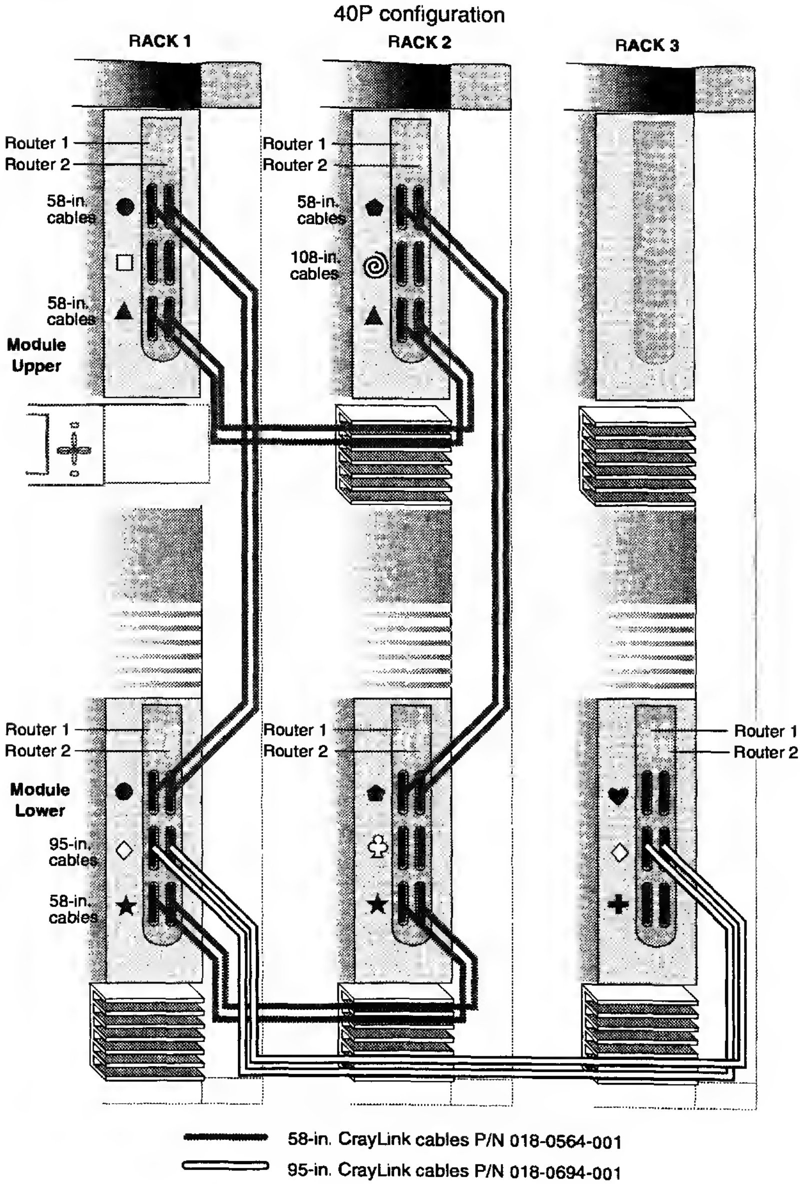


Figure 29. 48-processor System Cabling

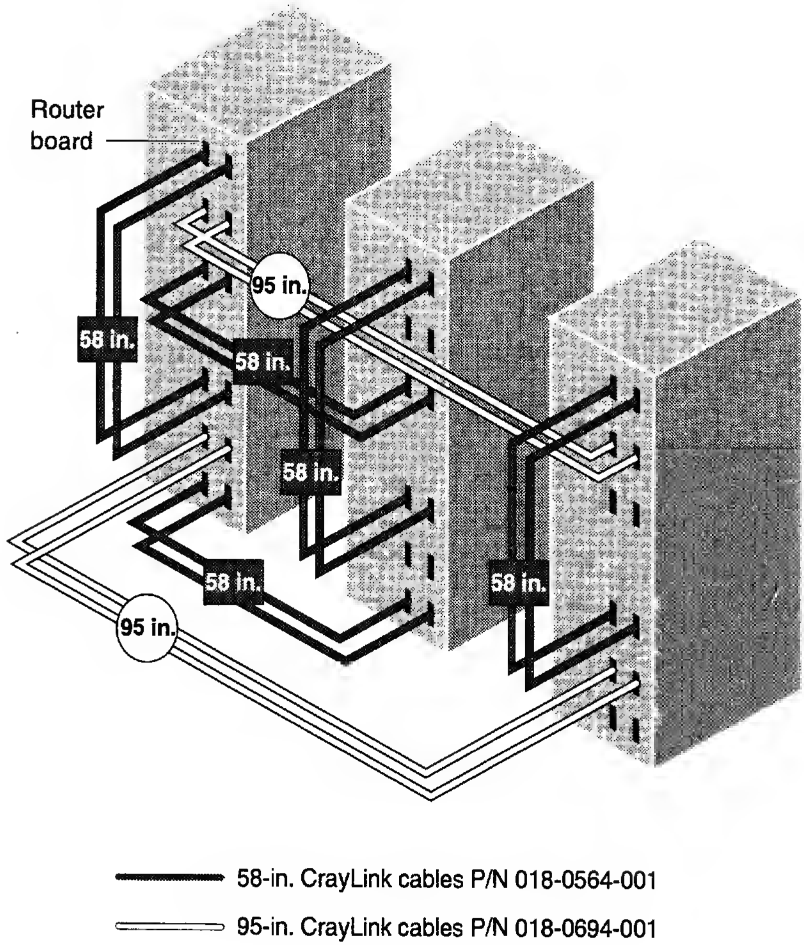


Figure 30. 48-processor System Cable Routing

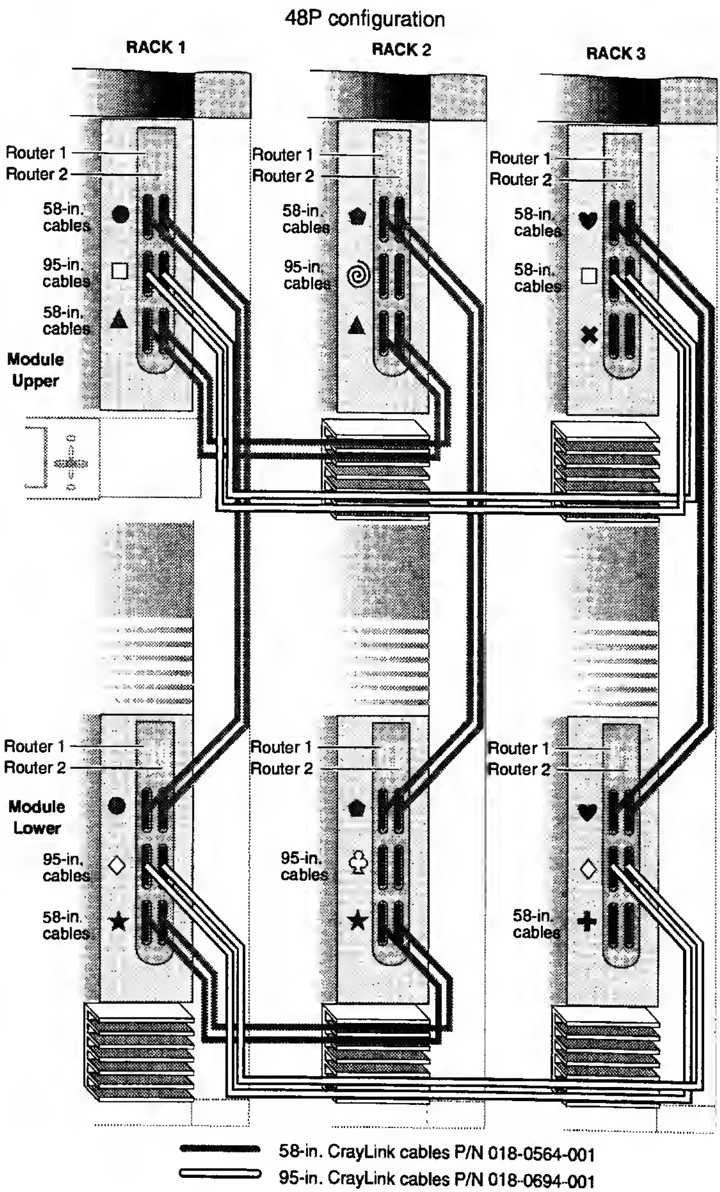


Figure 31. 56-processor System Cabling

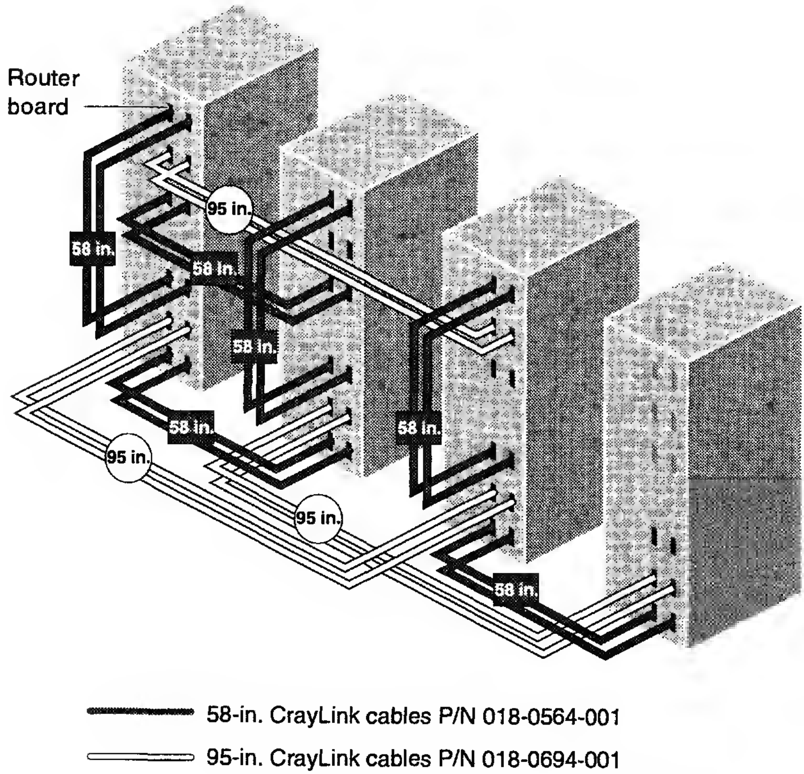


Figure 32. 56-processor System Cable Routing

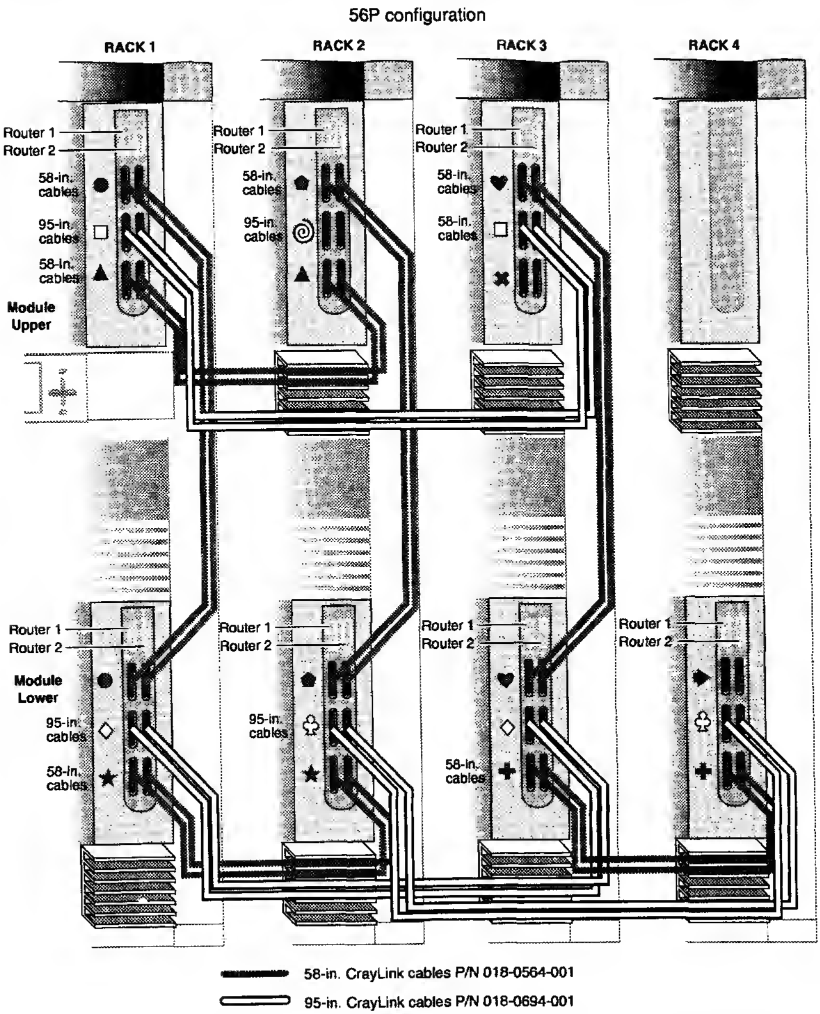


Figure 33. 64-processor System Cabling

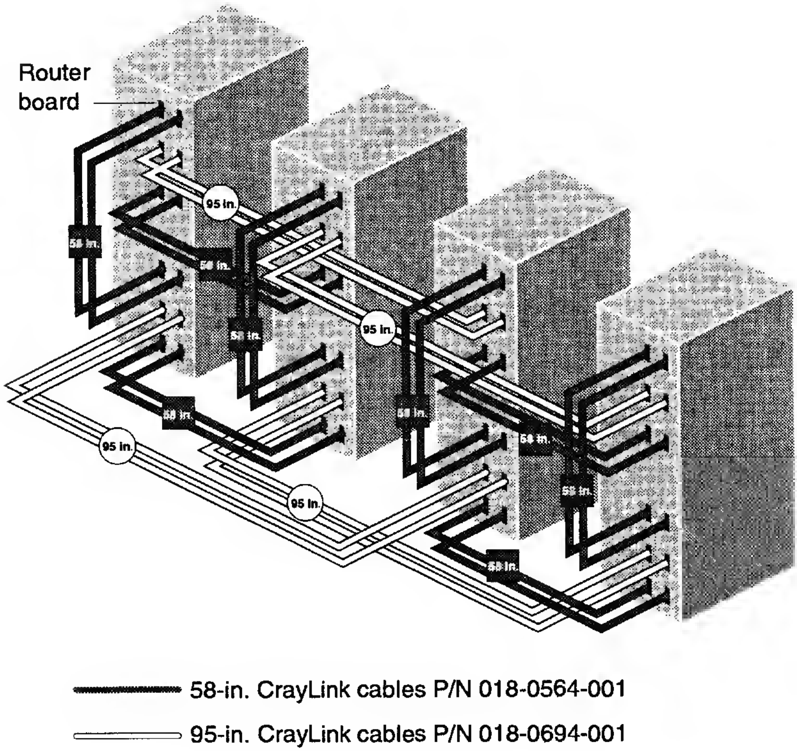


Figure 34. 64-processor System Cable Routing

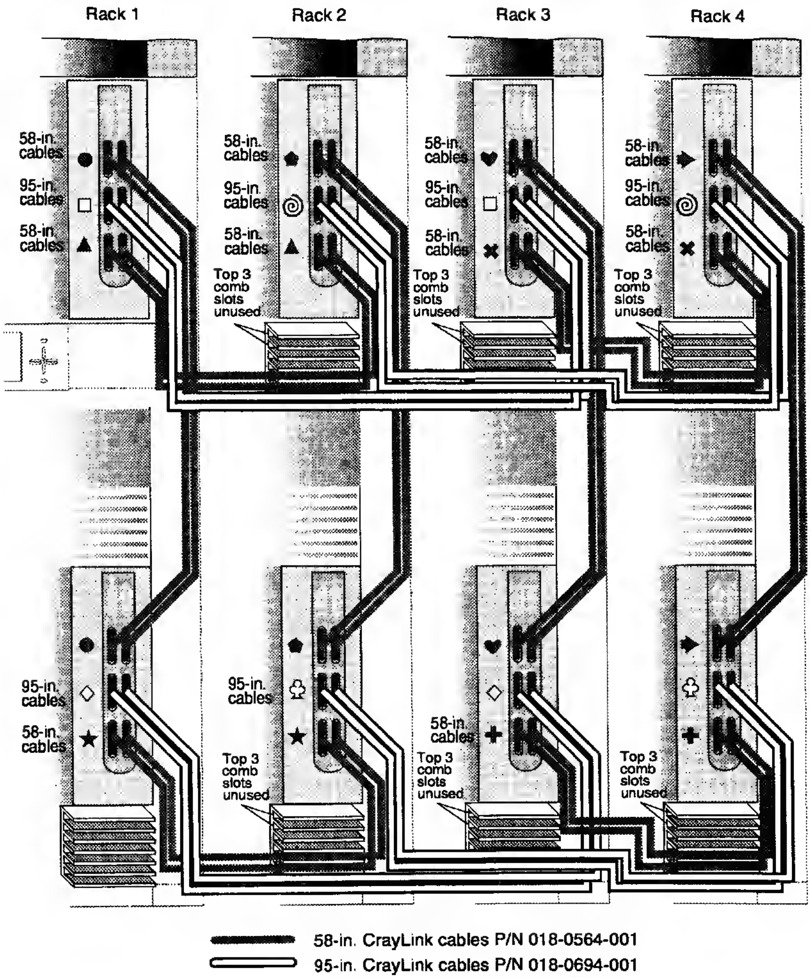


Figure 35. CrayLink Cables and Router Board Ports

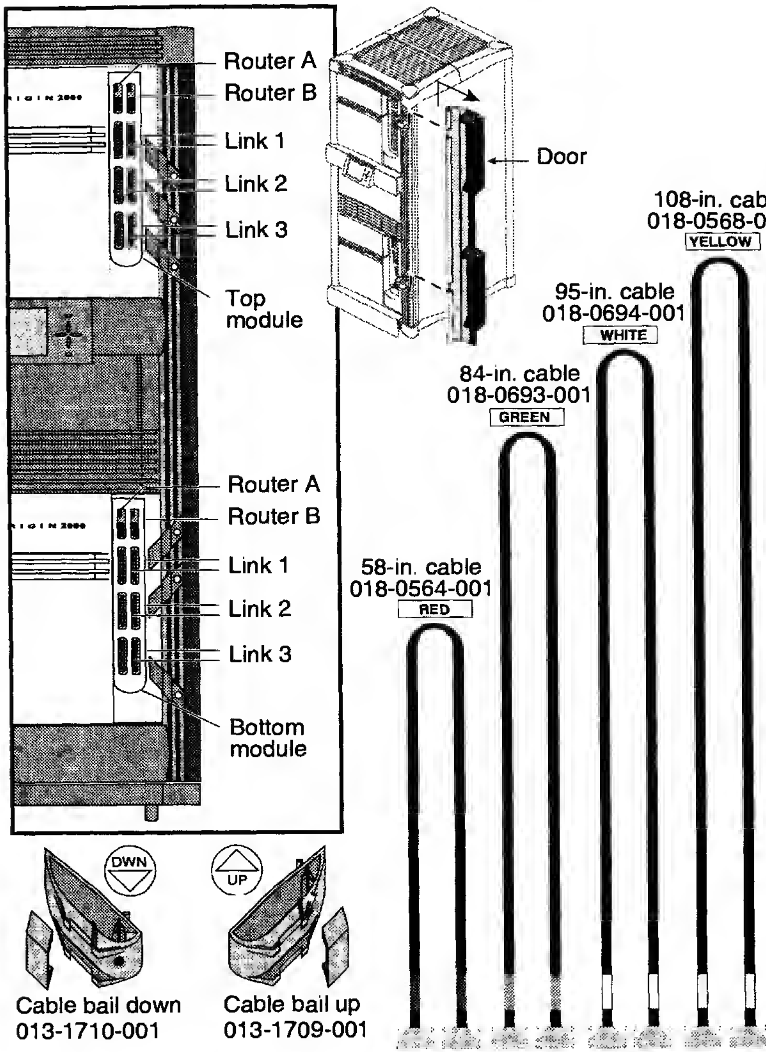


Figure 36. MetaRouter Slot Numbering

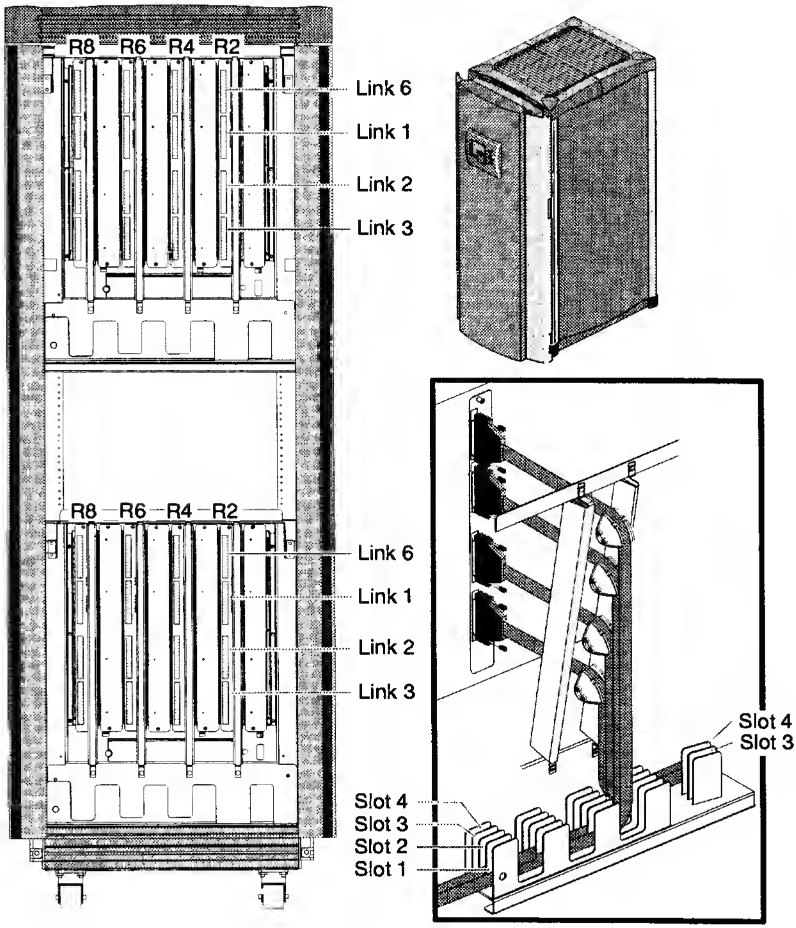
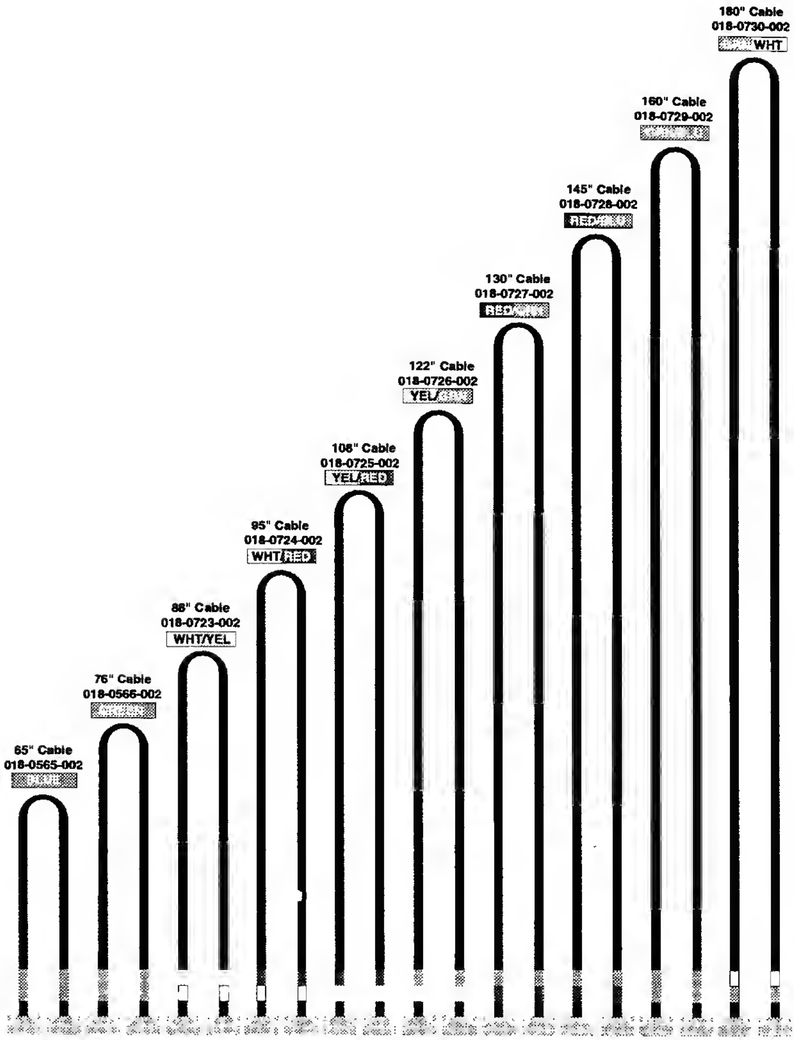


Figure 37. MetaRouter Cable Kit (P/N 026-1130-001)



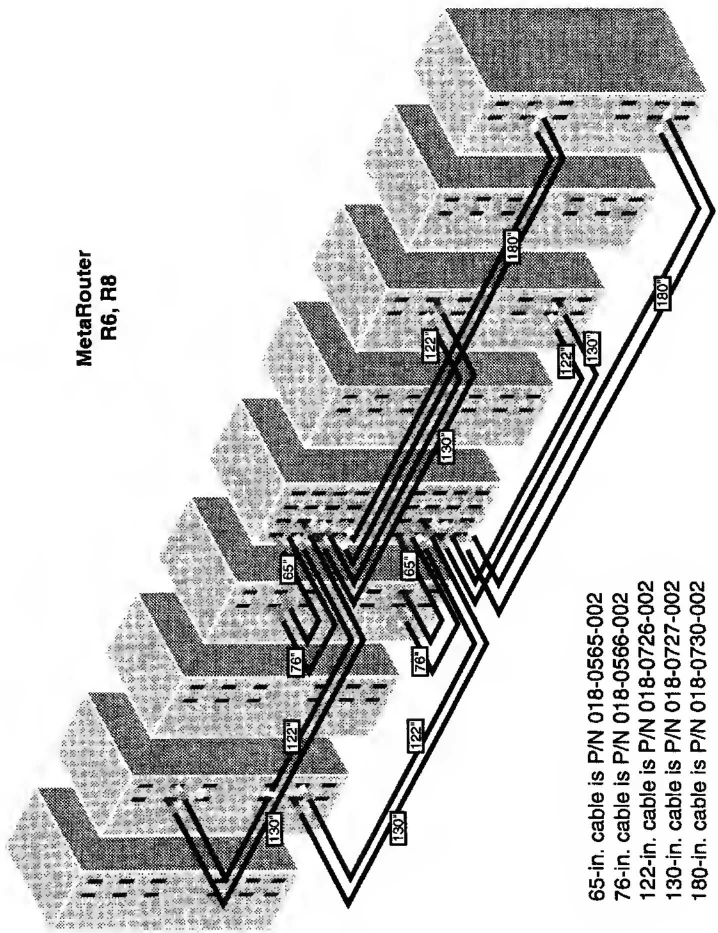


Figure 38. MetaRouter Cabling: Routers R1 and R3 Upper and Lower Modules

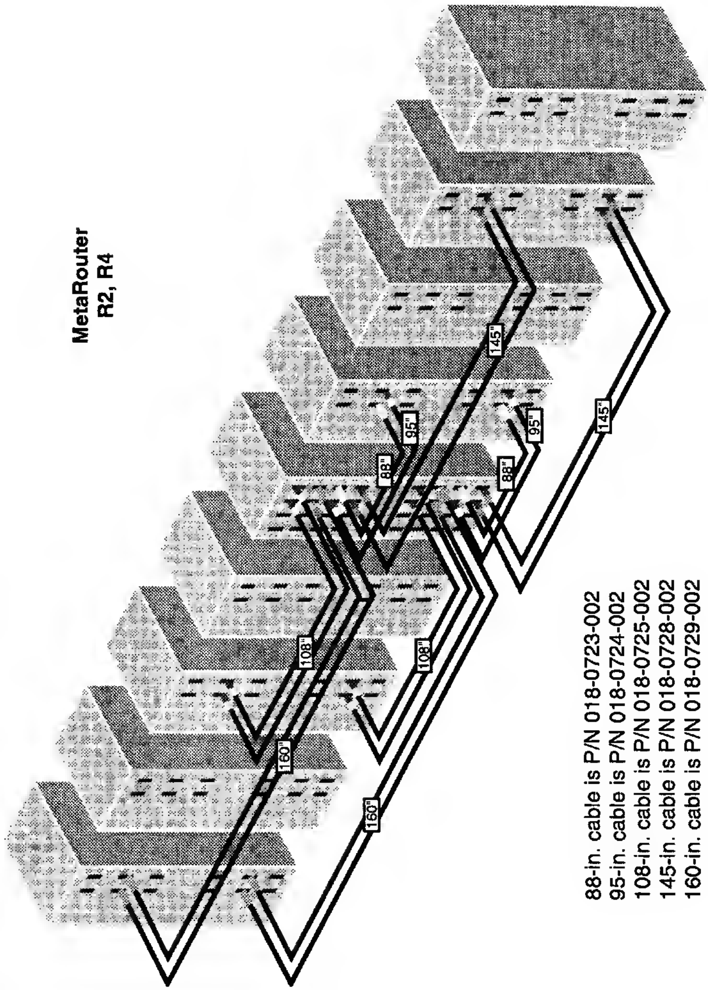


Figure 39. MetaRouter Cabling: Routers R5 and R7 Upper and Lower Modules

Table 33. MetaRouter to Origin 2000 Rack Cable Connections

	Step	Cable		MetaRouter		Origin 2000 Module		
		Color	Length	Router/Link	Cable Comb	Module	Cable Comb Route	Router/Link 2
Upper Modules	1	Green/White	180	R8 /L3	Slot 4	18	Slot 6	R2/L2
	2	Green/White	180	R6 /L3	Slot 4		Slot 5	R1/L2
	3	Red/Blue	145	R4 /L3	Slot 4	16	Slot 4	R2/L2
	4	Red/Blue	145	R2 /L3	Slot 4		Slot 3	R1/L2
	5	Red/Green	130	R8 /L2	Slot 3	14	Slot 6	R2/L2
	6	Yellow/Green	122	R6 /L2	Slot 3		Slot 5	R1/L2
	7	White/Red	95	R4 /L2	Slot 3	12	Slot 4	R2/L2
	8	White/Yellow	88	R2 /L2	Slot 3		Slot 3	R1/L2
	9	Green/Blue	160	R2 /L1	Slot 2	2	Slot 6	R1/L2
	10	Green/Blue	160	R4 /L1	Slot 2		Slot 5	R2/L2
	11	Red/Green	130	R6 /L1	Slot 2	4	Slot 4	R1/L2
	12	Yellow/Green	122	R8 /L1	Slot 2		Slot 3	R2/L2
	13	Yellow/Red	108	R2 /L6	Slot 1	6	Slot 6	R1/L2
	14	Yellow/Red	108	R4 /L6	Slot 1		Slot 5	R2/L2
	15	Green	76	R6 /L6	Slot 1	8	None	R1/L2
	16	Blue	65	R8 /L6	Slot 1		None	R2/L2

Table 33. MetaRouter to Origin 2000 Rack Cable Connections (continued)

	Step	Cable		MetaRouter		Origin 2000 Module		
		Color	Length	Router/Link	Cable Comb	Module	Cable Comb Route	Router/Link 2
Lower Modules	17	Green/White	180	R8 /L3	Slot 4	17	Slot 6	R2/L2
	18	Green/White	180	R6 /L3	Slot 4		Slot 5	R1/L2
	19	Red/Blue	145	R4 /L3	Slot 4	15	Slot 4	R2/L2
	20	Red/Blue	145	R2 /L3	Slot 4		Slot 3	R1/L2
	21	Red/Green	130	R8/L2	Slot 3	13	Slot 6	R2/L2
	22	Yellow/Green	122	R6 /L2	Slot 3		Slot 5	R1/L2
	23	White/Red	95	R4 /L2	Slot 3	11	Slot 4	R2/L2
	24	White/Yellow	88	R2 /L2	Slot 3		Slot 3	R1/L2
	25	Green/Blue	160	R2/L1	Slot 2	1	Slot 6	R1/L2
	26	Green/Blue	160	R4 /L1	Slot 2		Slot 5	R2/L2
	27	Red/Green	130	R6 /L1	Slot 2	3	Slot 4	R1/L2
	28	Yellow/Green	122	R8 /L1	Slot 2		Slot 3	R2/L2
	29	Yellow/Red	108	R2 /L6	Slot 1	5	Slot 6	R1/L2
	30	Yellow/Red	108	R4 /L6	Slot 1		Slot 5	R2/L2
	31	Green	76	R6 /L6	Slot 1	7	None	R1/L2
	32	Blue	65	R8/L6	Slot 1		None	R2/L2

Midplane Jumper Settings

Set the midplane oscillator jumpers as follows for your system (refer to Figure 40 and Figure 41):

Origin 2000 System

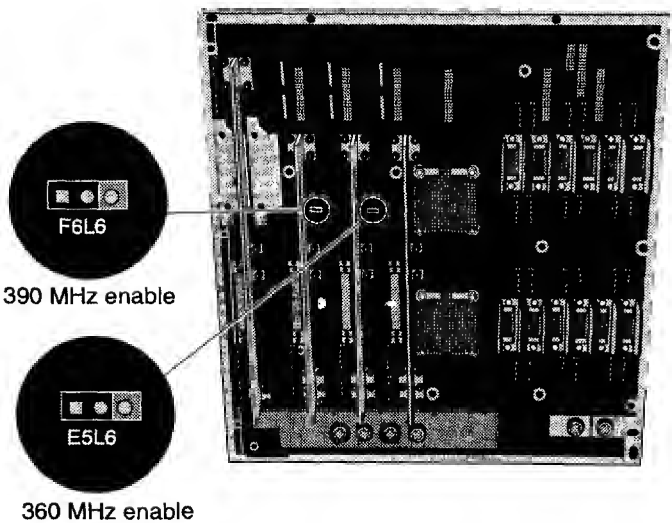
Midplane revision AB and earlier:

- The position of the jumpers for 1-MB HIMM (180 Mhz) and 4-MB HIMM (195 Mhz) node board assemblies is the same.
- The jumpers should be on the headers (platforms) at D3L4B and D3L2B (pins 1 and 2).

Midplane revision C and later:

- For 1-MB HIMM assemblies, install the jumper on header (platform) F5L4B (pins 1 and 2).
- For 4-MB HIMM assemblies, install the jumper on header (platform) E4L4B (pins 1 and 2).

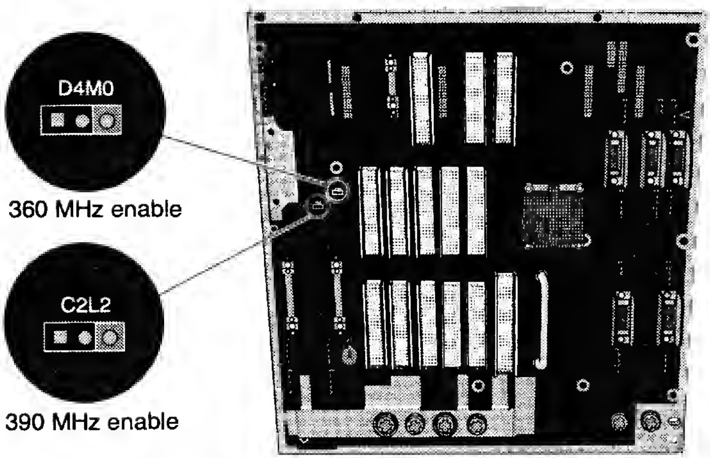
Figure 40. Origin 2000 Midplane Oscillator Jumpers



Onyx2 Deskside System

- For 1-MB HIMM IP27 assemblies, install the jumper on header (platform) D4M0 (pins 1 and 2).
- For 4-MB HIMM IP27 assemblies, install the jumper on header (platform) C2L2 (pins 1 and 2).

Figure 41. Onyx2 Midplane Oscillator Jumpers



IP31 Compatibility with IP27 Boards

Any IP31 installed in an Onyx2 or Origin 2000 system must be serviced by an 8P12 router that is at part number 030-0841-003 revision A or later. For node card slots N1-N2, this is the router card in slot R1 (the leftmost router as viewed from the router card side of the system); and for node card slots N3-N4, this is the router card in slot R2 (the rightmost router as viewed from the router card side of the system). Refer to Figure 42 and Figure 43.

The following rules apply to mixing IP27 and IP31 boards in a system:

1. Only one module in any system may have a mixture of IP27 node boards and IP31 node boards. Any additional modules must contain homogeneous node cards; that is, only IP27 or only IP31 node boards.
2. The PROM must be at revision 5.26 or later.
3. All CPUs that connect to a full router board (2 node boards per router board) and node boards that connect via null router must run at the same speed.
4. A mixture of IP27 and IP31 node boards cannot share the same router.
5. In a module with mixed IP27 and IP31 node boards, the IP31 node boards must occupy the slots immediately to the left of the IP27 node boards. Any IP27 node boards should be placed in the lower-numbered slots (closer to the XIO slots) starting at N1.
6. IP31 node boards are not supported in any mixed configuration with 180-MHz IP27 node boards. Only 195-MHz IP27 node boards may coexist in the system with IP31 node boards.
7. Systems with 180-MHz IP27 node boards can be upgraded to IP31 node-board systems by replacing the 180-MHz IP27 node boards and changing the midplane jumper to enable 390-MHz operation. Origin 2000 and Onyx2 rackmount systems require midplane P/N 030-0762-006 rev C or later, and Onyx2 deskside systems require midplane P/N 013-1839-001 or P/N 030-1066-002 or later.
8. Modules that contain node boards of one clock speed can be mixed with modules that contain a different clock speed, to a maximum of three different clock speeds in a single system: 195 MHz, 250 MHz, and 390 MHz.
9. Mixing is not supported in Origin 2000 and Onyx2 deskside systems that scale from 1 to 4 processors (null router).

10. Origin 2000 and Onyx2 *deskside* systems of 6 to 8 processors may have two processor speeds: the first four processors must have the same speed; the remaining processors may have different speeds.

Figure 42. Correct IP31 Module and Node Board Configurations

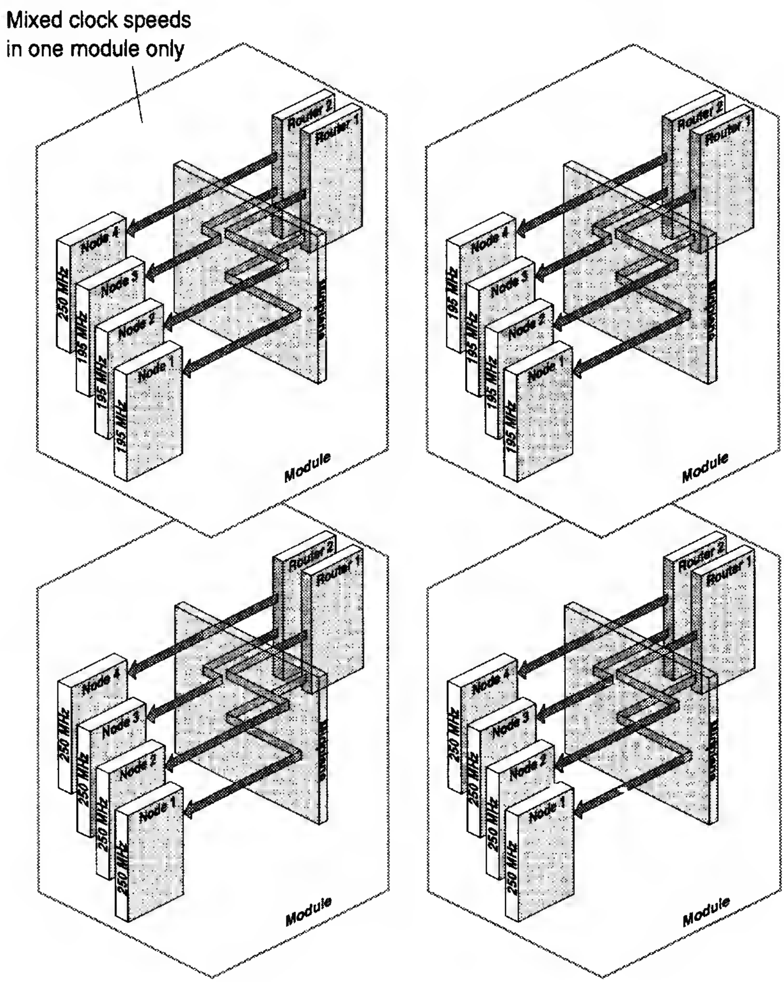
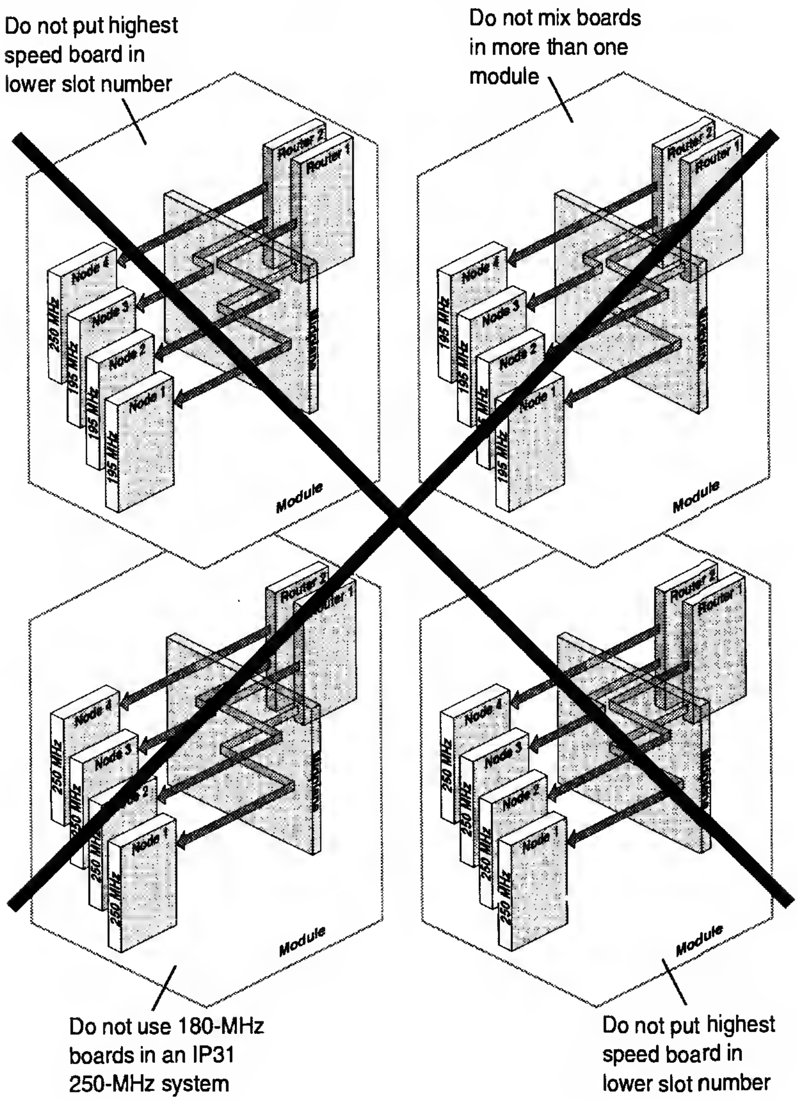


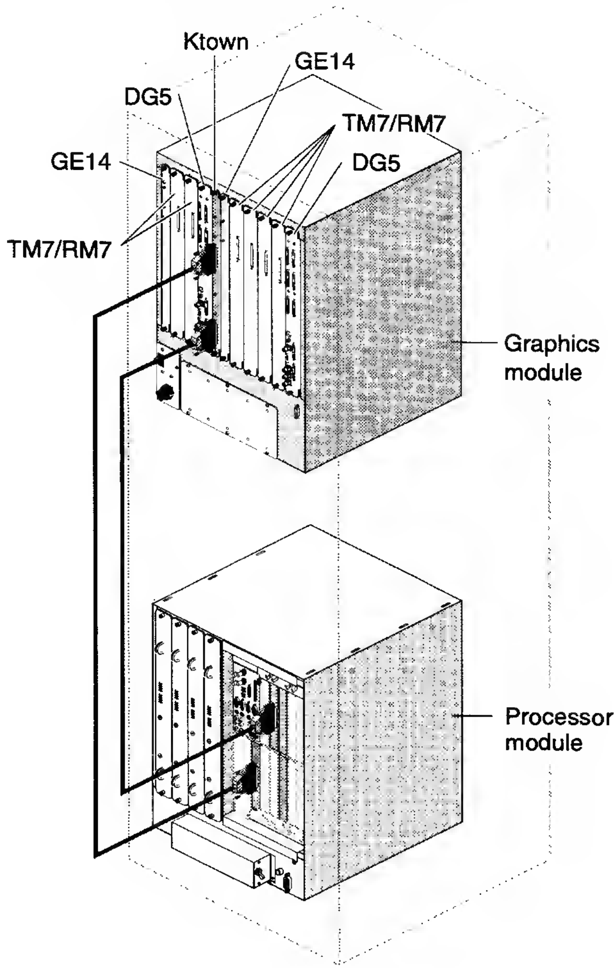
Figure 43. Incorrect IP31 Module Node Board Configurations



Onyx2 Systems

This section contains cabling information for the following Onyx2 configurations and devices:

Figure 44. Onyx2 Single-rack System (8P and 2 Graphics Pipes)



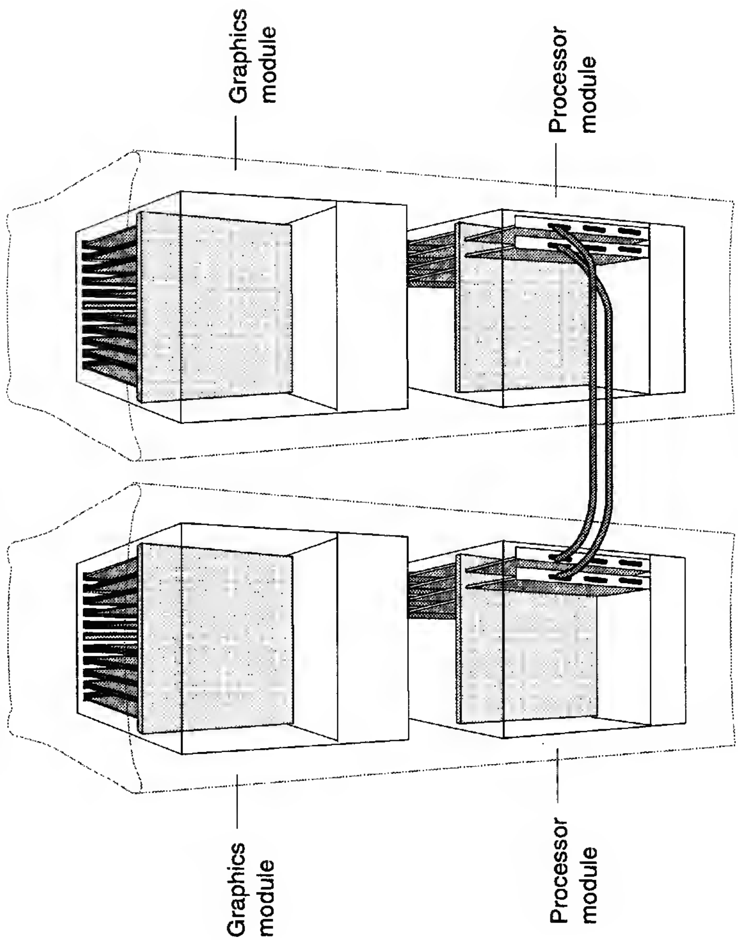


Figure 45. Onyx2 Multirack System (16P and 4 Graphics Pipes)

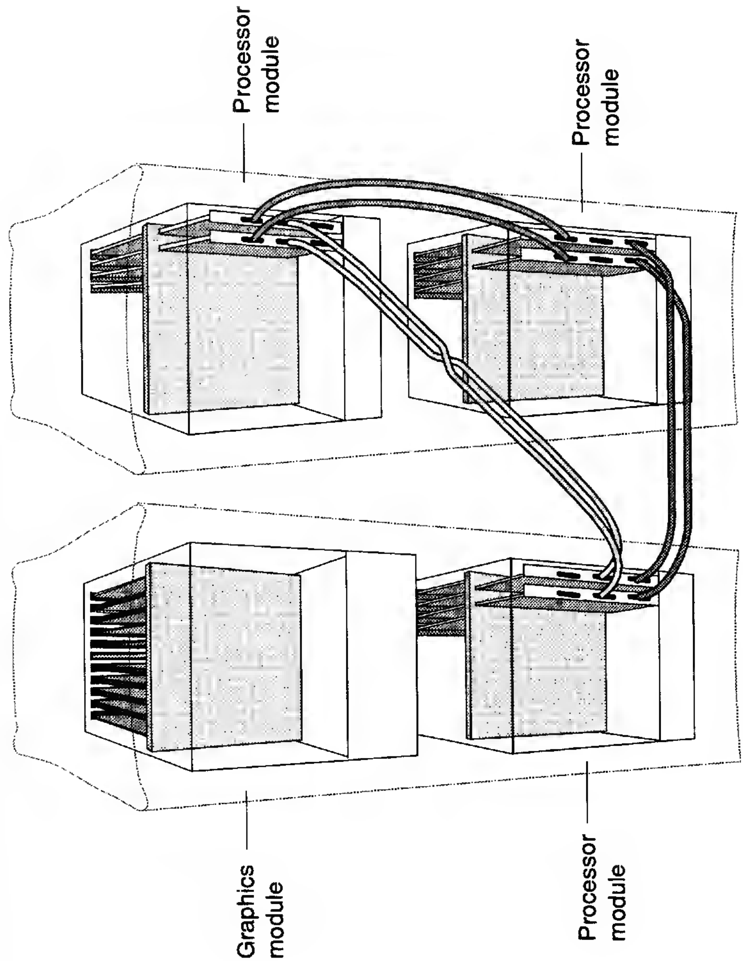


Figure 46. Onyx2 Multirack System (24P and 2 Graphics Pipes)

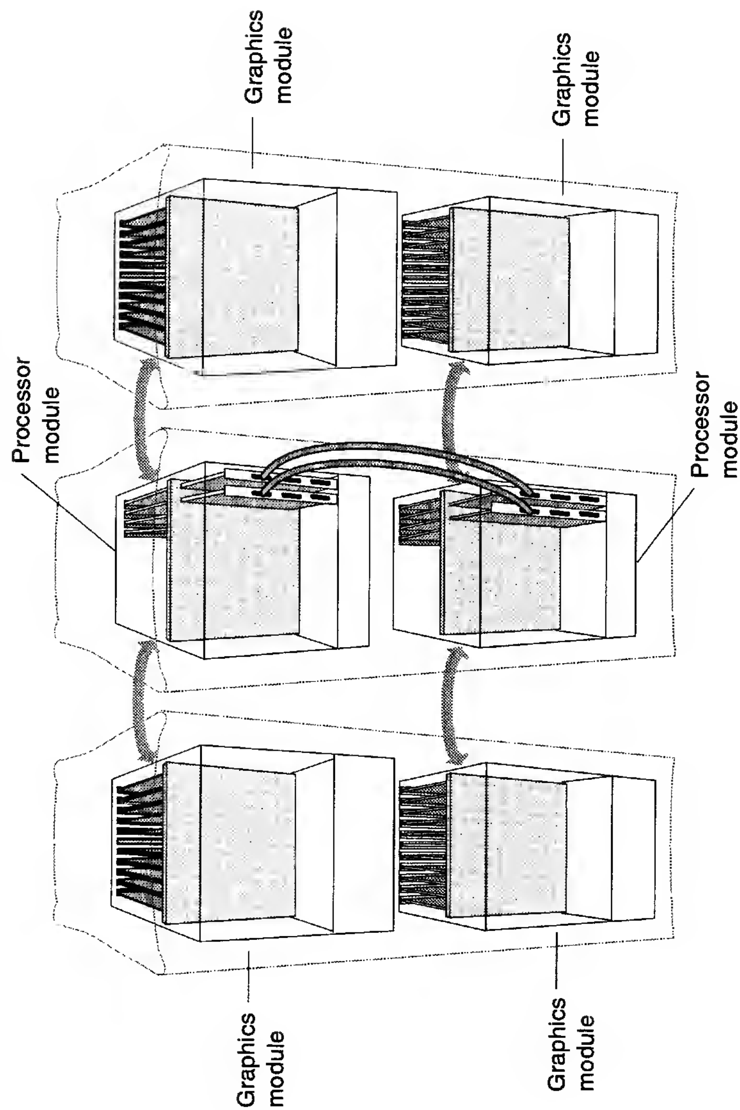


Figure 47. Onyx2 RealityMonster Multirack System (16P and 8 Graphics Pipes)

Figure 48. Onyx2 RealityMonster Ktown Cabling

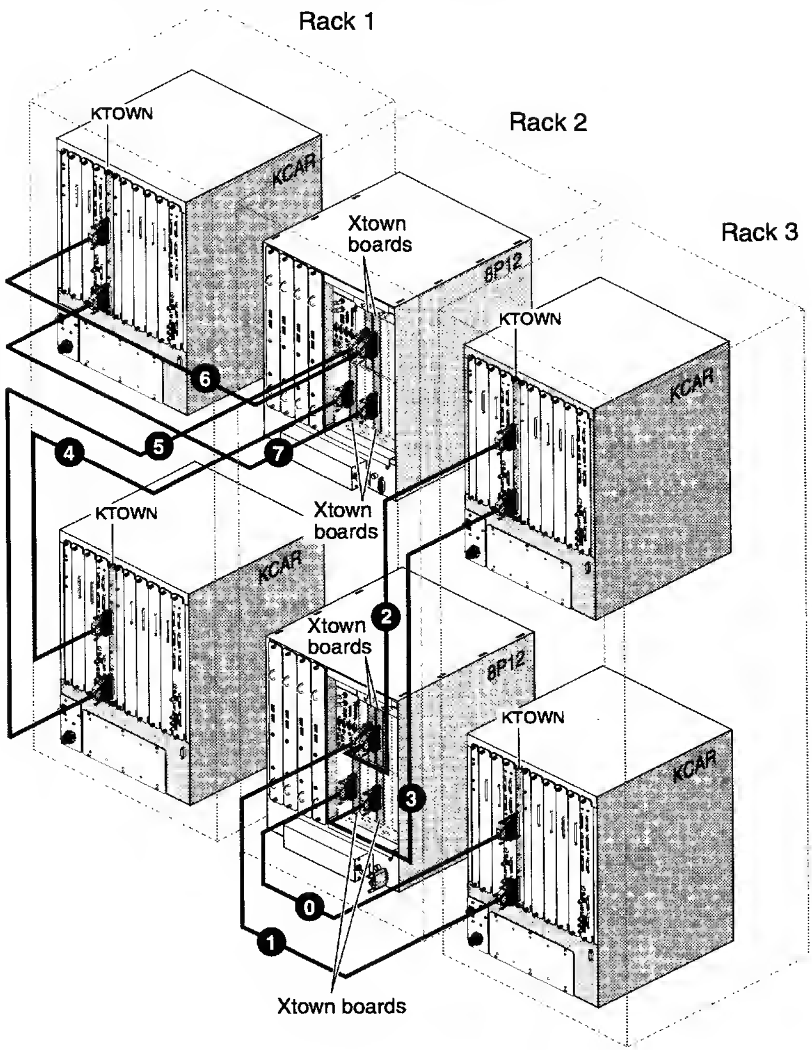
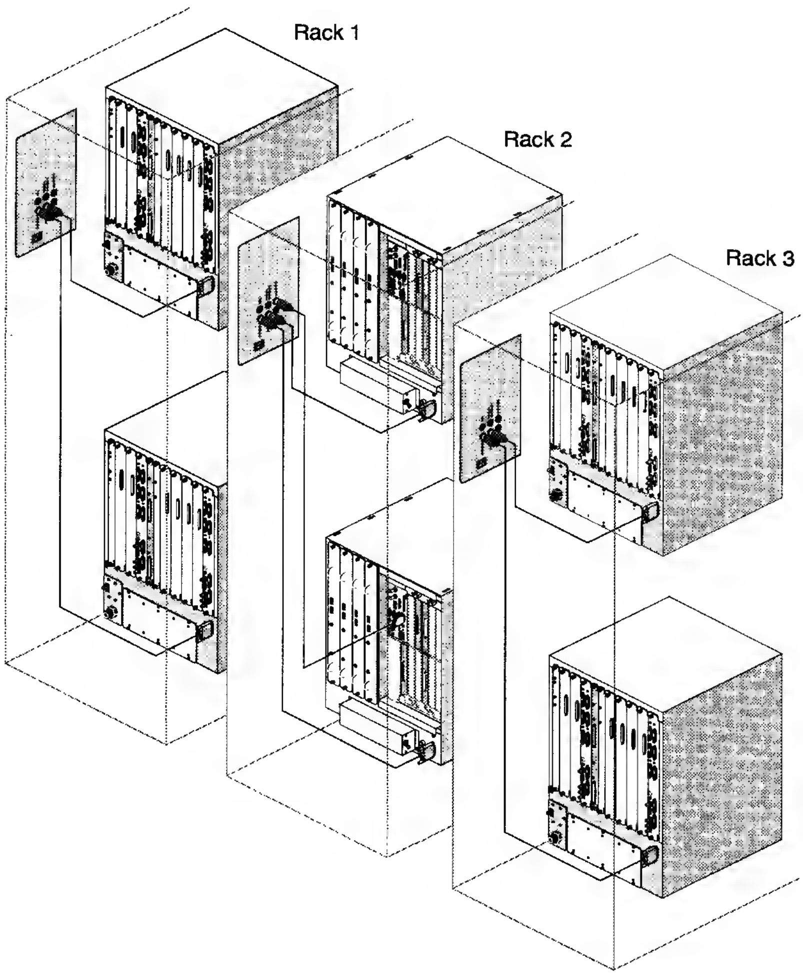


Figure 49. Onyx2 RealityMonster MMSC Cabling

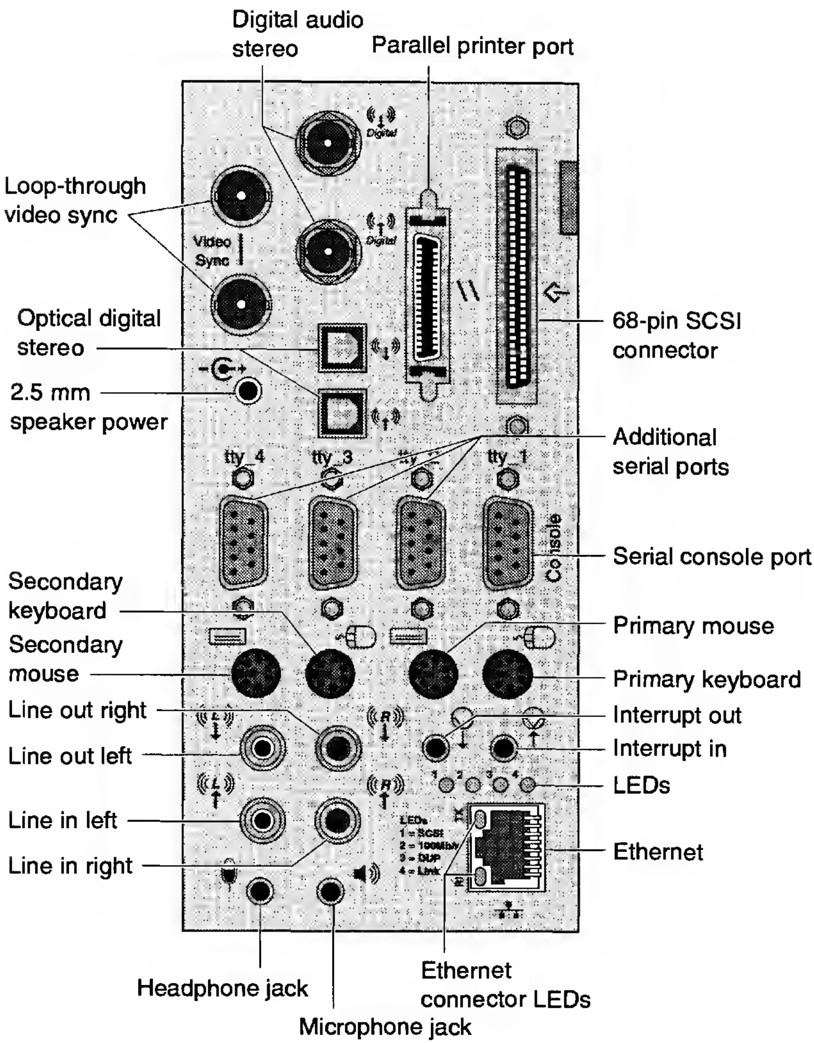


IO6G Cabling and Pin Assignments

The Onyx2 deskside and Onyx2 rack systems use a different BaseIO (IO6G) module than Origin 2000 series systems (refer to Figure 50).

The following IO6G connectors are covered in this section:

Figure 50. BaseIOG (IO6G) for Onyx2 Systems



Onyx2 Serial Ports

Figure 51 shows the Onyx2 deskside system serial port (Console tty_1). Table 34 (on page 108) lists the serial port connector pin assignments in RS-232 mode. Table 35 (on page 108) lists the serial port connector pin assignments in RS-422 mode.

Figure 51. Onyx2 Serial Port Connections

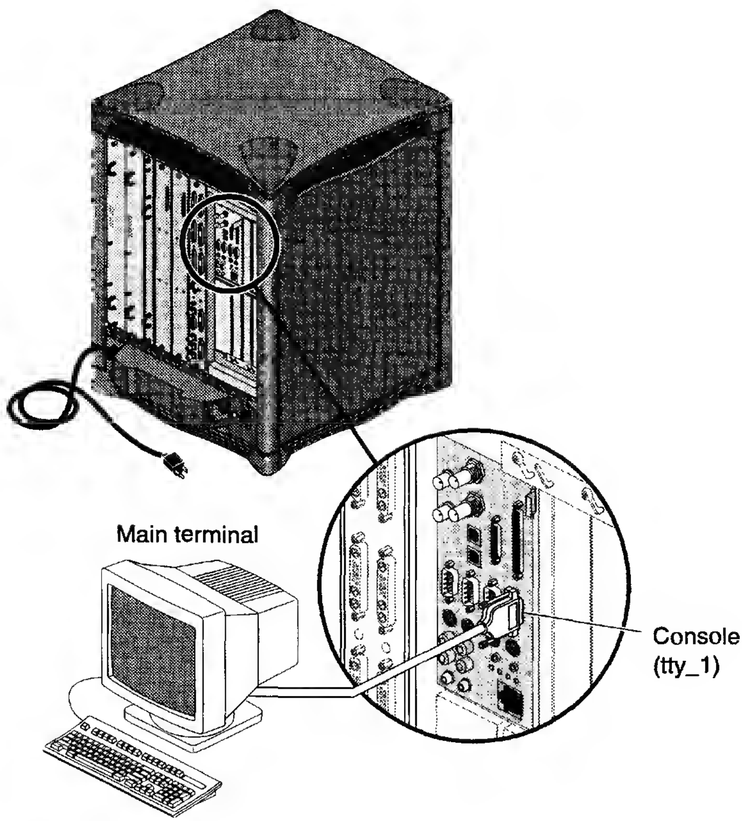


Table 34. Onyx2 Serial Port Pin Assignments, RS-232 Mode

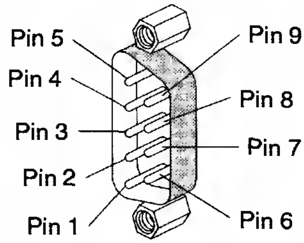
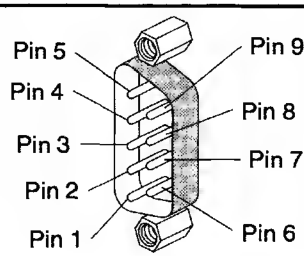
Pin	Assignment	Pin Numbers
1	Data Carrier Detect (DCD)	
2	Receive Data (RD)	
3	Transmit Data (TD)	
4	Data Terminal Ready (DTR)	
5	Signal Ground (SG)	
6	Data Set Ready (DSR)	
7	Request To Send (RTS)	
8	Clear To Send (CTS)	
9	Not Connected	

Table 35. Onyx2 Serial Port Pin Assignments, RS-422 Mode

Pin	Assignment	Pin Numbers
1	Reserved	
2	Receive Data (RXD –)	
3	Transmit Data (TXD –)	
4	Transmit Data (TXD +)	
5	Signal Ground (GND)	
6	Receive Data (RXD +)	
7	Output Handshake (HSKo)	
8	Input Handshake (HSKi)	
9	(Reserved)	

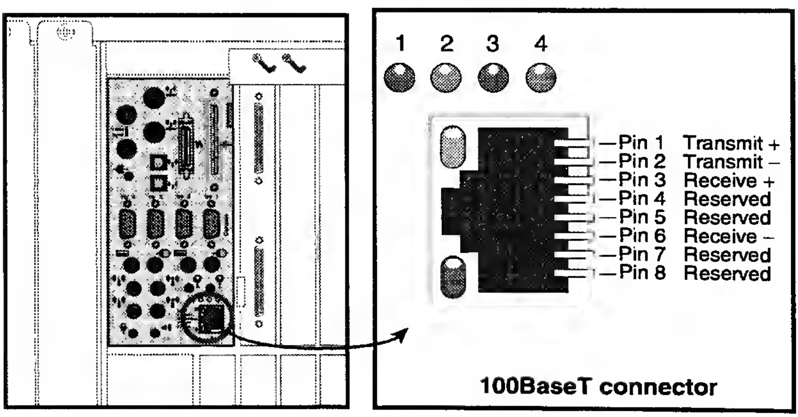
Onyx2 100BaseT Ethernet Port

There are two LEDs on the RJ-45 Ethernet port (refer to Figure 52). The top (green) LED illuminates only when the system is transmitting. The bottom (yellow) LED illuminates whenever it senses a packet on the wire, including packets not destined for your system.

The four LEDS above the RJ-45 Ethernet connector have the following functions:

- The yellow LED on the far left (LED 1) illuminates to indicate SCSI activity on the BaseIO single-ended SCSCI connector.
- The green LED (LED 2) illuminates to indicate 100 MB-per-second packet activity.
- The yellow LED on the right (LED 3) indicates when the Ethernet is operating at full duplex rates of transfer or receive.
- The green LED on the far right (LED 4) shows the Ethernet link test. It illuminates when the link state is valid.

Figure 52. Onyx2 100BaseT Ethernet Port Pin Assignments



Onyx2 SCSI Port Pin Assignments

A single, external 68-pin SCSI connector is provided on the IO6G panel (refer to Figure 53 and Table 36). This connector supports both Ultra SCSI and SCSI-2 devices. The connector sends single-ended SCSI signals only. Optional additional SCSI ports can be implemented using XIO option boards. The dash (–) preceding a signal name indicates that the signal is low.

Note: 8-bit devices that connect to the P cable leave the following signals open: –DB(8), –DB(9), –DB(10), –DB(11), –DB(12), –DB(13), –DB(14), –DB(15), –DB(P1). All other signals are connected as shown in Table 36.

Figure 53. Onyx2 Deskside SCSI Connector Pin Assignments

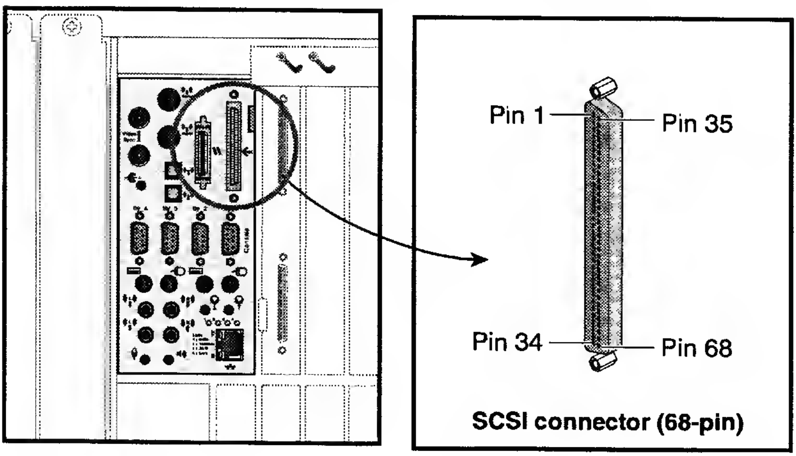


Table 36. Onyx2 SCSI Port Pin Assignments

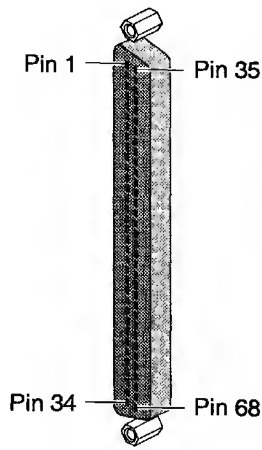
Pin	Assignment	Pin Numbers
1	Ground	
2	Ground	
3	Ground	
4	Ground	
5	Ground	
6	Ground	
7	Ground	
8	Ground	
9	Ground	
10	Ground	
11	Ground	
12	Ground	
13	Ground	
14	Ground	
15	Ground	
16	Ground	
17	TERMPWR	
18	TERMPWR	
19	Spacing	
20	Ground	
21	Ground	
22	Ground	
23	Ground	
24	Ground	
25	Ground	
26	Ground	
27	Ground	
28	Ground	
29	Ground	
30	Ground	
31	Ground	

Table 36. Onyx2 SCSI Port Pin Assignments (continued)

Pin	Assignment
32	Ground
33	Ground
34	Ground
35	–DB(12)
36	–DB(13)
37	–DB(14)
38	–DB(15)
39	–DPARH
40	–D0
41	–D1
42	–D2
43	–D3
44	–D4
45	–D5
46	–D6
47	–D7
48	–DPAR
49	Ground
50	Ground
51	TERMPWR
52	TERMPWR
53	Reserved
54	Ground
55	–ATN
56	Ground
57	–BSY
58	–ACK
59	–RST
60	–MSG
61	–SEL
62	–C/D

Pin Numbers

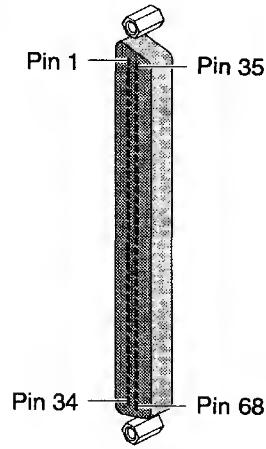
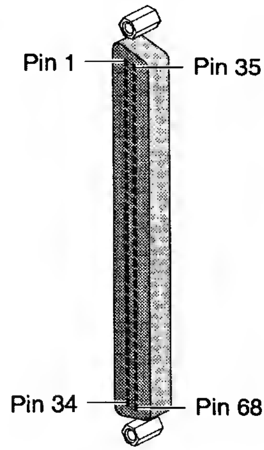


Table 36. Onyx2 SCSI Port Pin Assignments (continued)

Pin	Assignment	Pin Numbers
63	–REQ	
64	–I/O	
65	–DB(8)	
66	–DB(9)	
67	–DB(10)	
68	–DB(11)	

Onyx2 Parallel Port Pin Assignments

The BaseIO board supports one IEEE 1284-C 36-pin parallel port connector. The location of this connector is shown in the following figure. Pinouts for the parallel port connector are listed in Table 37.

Suitable cables for use with this port should be marked "IEEE 1284-compliant." For most parallel printers, you can use a cable with an IEEE 1284-C connector at the Onyx2 end and an IEEE 1284-B connector (also known as a Centronics-style) at the printer end.

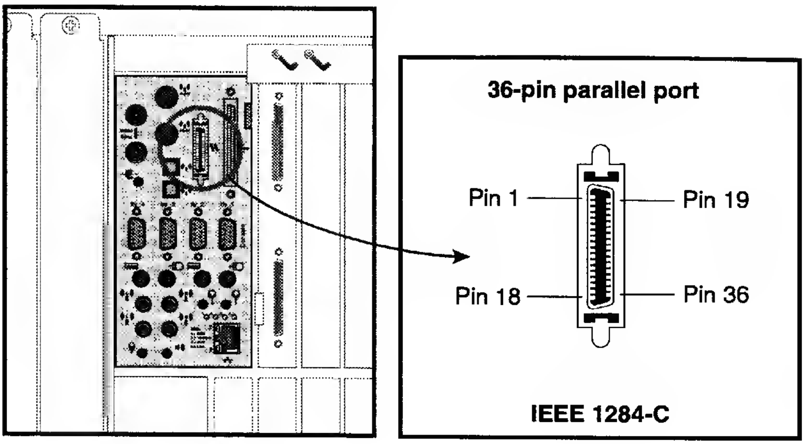


Table 37. Parallel Port Pin Assignments

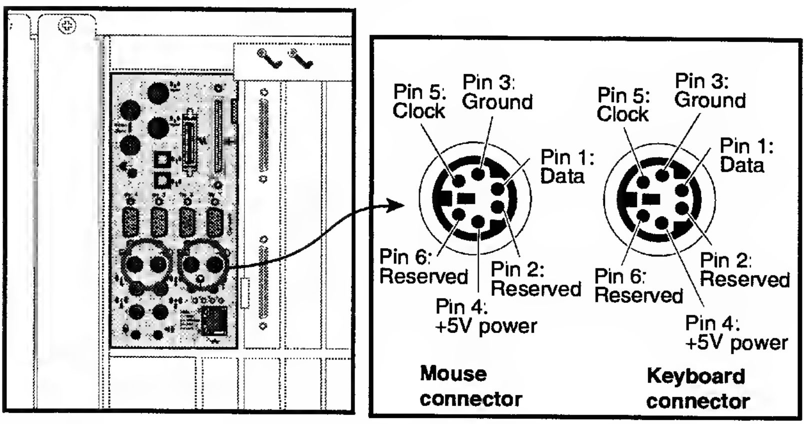
Pin	Assignment
1	STB
2	DATA1
3	DATA2
4	DATA3
5	DATA4
6	DATA5
7	DATA6
8	DATA7
9	DATA8
10	ACK
11	BUSY
12	PE
13	ONLINE
14	PR/SC
15	NOPAPER
16	Not Connected
17	NOINK
18	Not Connected
19–30	Signal Ground (GND)
31	RESET
32–36	Not Connected

Mouse and Keyboard Port Pin Assignments

Figure 54 shows the location of the connectors and their pinouts. There are two sets of keyboard and mouse connectors on the rear of the IO6G. If your system uses one keyboard and mouse, attach them to the primary keyboard and mouse connector ports. These primary ports are located on the right side of the BaseIO panel.

You can plug the keyboard and mouse cables directly into the IO6G panel. However, in cases where your monitor, keyboard, and mouse are located away from the system, use the included extension cable. Each system comes with a 24-foot (7.3 m) keyboard and mouse extension cable.

Figure 54. Mouse and Keyboard Pin Assignments



When your system uses one keyboard and mouse, the system assigns the `/hw/input` file to the primary port and the `/hw/input1` file to the secondary port. The operating system links these files to special character files in the `/dev` directory. You can use these files to assign a specific keyboard/mouse set to a particular X display.

When your system contains a second BaseIO-G, the system assigns the `/hw/input2` file to the secondary port and the `/hw/input3` file to the primary port. The system assigns `/hw/inputn` numbers to subsequent BaseIO-Gs in the same manner (refer to Table 38). The Onyx2 system supports a maximum of 8 keyboard/mouse sets.

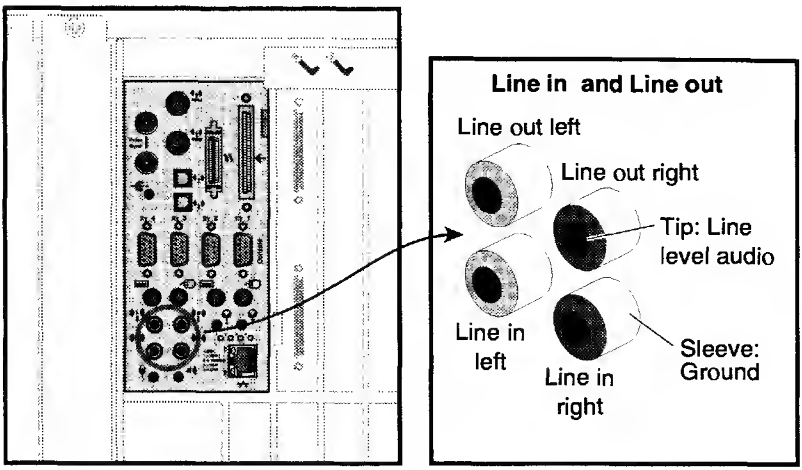
Table 38. Assignment of hwgraph Files to Primary and Secondary Ports

BaseIO-G	hwgraph File	Port
1	/hw/input	Primary
1	/hw/input1	Secondary
2	/hw/input2	Secondary
2	/hw/input3	Primary
3	/hw/input4	Primary
3	/hw/input5	Secondary
4	/hw/input6	Secondary
4	/hw/input7	Primary

Analog Line In and Out (RCA) Connections

You may connect audio equipment to the line level inputs and outputs using standard shielded RCA-type connectors (refer to Figure 55). For best results, always route these analog signal cables away from power cords. The right channel is color coded red, and the left channel is white.

Figure 55. Line In and Stereo Out Connections

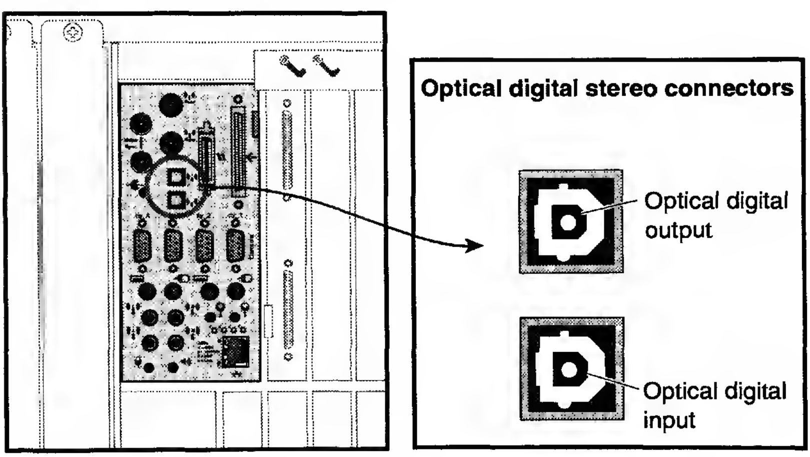


Optical Digital Audio Interface Connections

The single-jack ADAT optical connectors are located directly above serial port 3 (refer to Figure 56). These ports can be used with multitrack digital audio recording input and output devices.

These connections support optical input and output of eight channels at up to 24 bits and up to 48 KHz sample rates. Use standard plastic fiber interconnecting cables. You will need two cables: one for input and one for output. The Onyx2 system ships with connector cover plugs over the input and output ports. These must be removed before using the optical connectors. Retain these dust covers for use during shipment or if you discontinue ADAT use.

Figure 56. Optical Digital Audio Interface



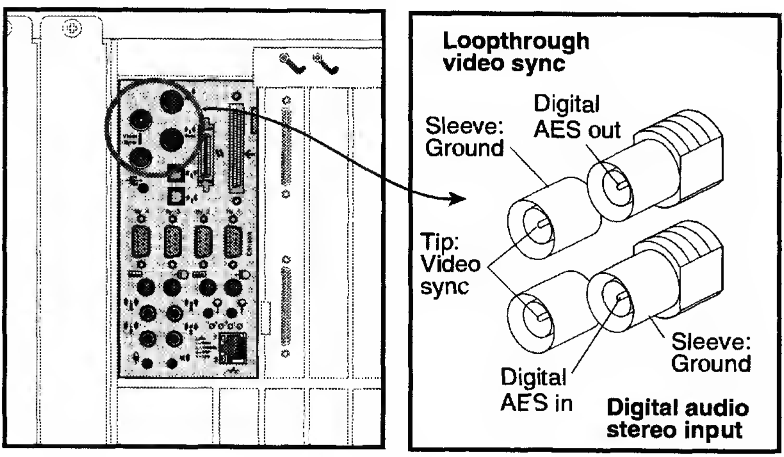
Loopthrough and Digital Audio Connections

The AES-3id-1995 digital audio connectors support 75-ohm signals at a nominal 1.0V (peak-to-peak) signal level. You should use a 75-ohm coaxial cable with standard BNC connectors for interconnections (such as with digital video recorders).

Some equipment supporting AES-3id-1995 digital audio signals uses 3-pin XLR connectors that support balanced 110-ohm signals. To successfully interconnect with equipment of this type, install a digital audio BALUN adapter at the equipment's XLR connector points. The BALUN adapter connects the 3-pin XLR to a 75-ohm BNC connection. The 75-ohm coaxial cable then connects between the Onyx2 system and the BALUN adapter.

Note: 110- to 75-ohm digital audio BALUN adapters are available in male and female versions. You need one of each type when using both the input and output AES-3id-1995 signal connectors.

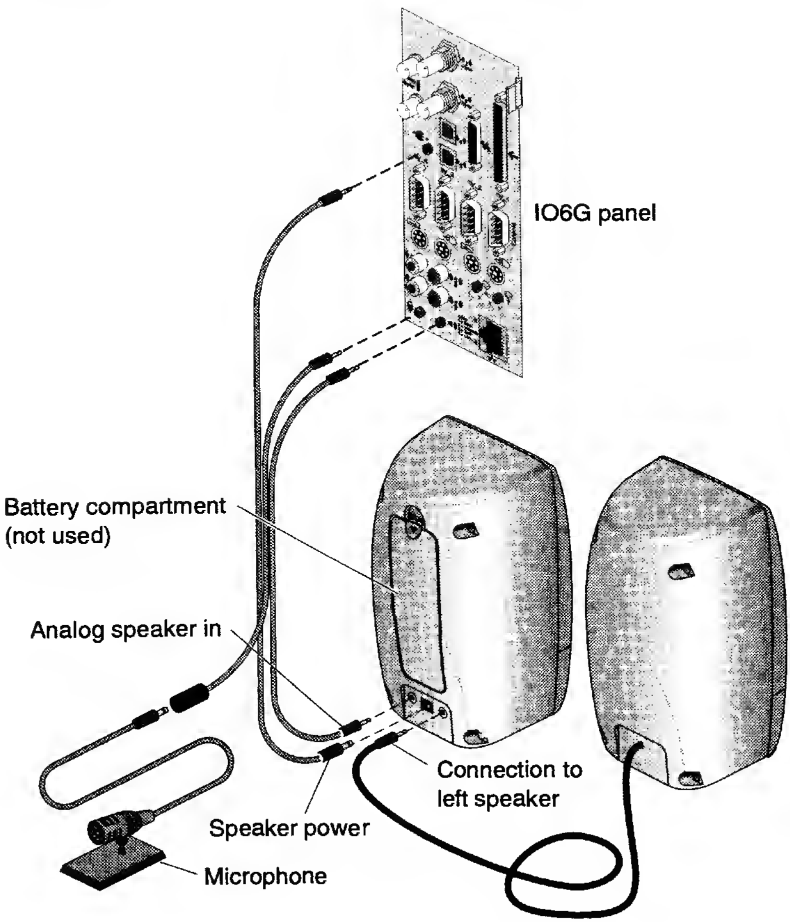
Figure 57. Loopthrough and Digital Audio Connectors



Speaker and Microphone Connections

The Onyx2 BaseIO panel uses a 30-foot (9.1 m) three-connector bundled cable to make connection to a microphone and a pair of speakers (refer to Figure 58).

Figure 58. Speaker and Microphone Connections to IO6G



External Interrupt Connections

The external interrupt interface uses two 1/8-inch (3.5-mm) stereo jacks: one to generate interrupts and one to receive interrupts. The jacks are located on the BaseIO and are wired as indicated in Table 39.

Table 39. External Interrupt Port Conductors

Conductor	Function
Tip	Interrupt (active low)
Ring	+5 V
Sleeve	Chassis ground and cable shield

When constructing interrupt cables, use two-conductor, shielded cable, wired straight through (tip-to-tip, ring-to-ring, sleeve-to-sleeve using the shield). For more information on the external interrupt interface, refer to the `ei`(7) man page.

DG5 Board Cabling Connections

This section describes the following topics:

Table 40 lists the DG5-8/VIO5H connectors and their functions. Figure 59 shows the 13W3 pinouts for the monitor connectors on the DG5 1/O panel; each 13W3 uses the same pinout pattern.

Table 40. DG5-8/VIO5H Connectors

Label	Type	Function
Monitor 0 through 7	13W3	Variable high-resolution monitor outputs
S-Video	4-pin mini-DIN	Interface to SVHS VCR or monitor
CMPST 1	RCA jack; BNC	Interface to composite monitor or VCR
StereoView ™	9-pin sub-D	Interface to StereoView device
Genlock In	BNC	Interface to video mixer
Genlock Out	BNC	Loopthrough connection
Swap Ready	BNC	Interface to other graphics systems

Figure 59. DG5/VIO5H 13W3 Connector Pinout

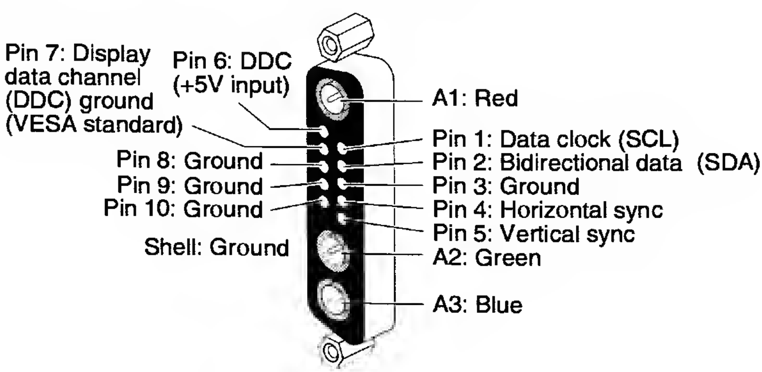


Figure 60. Onyx2 DG5-8 Display Board

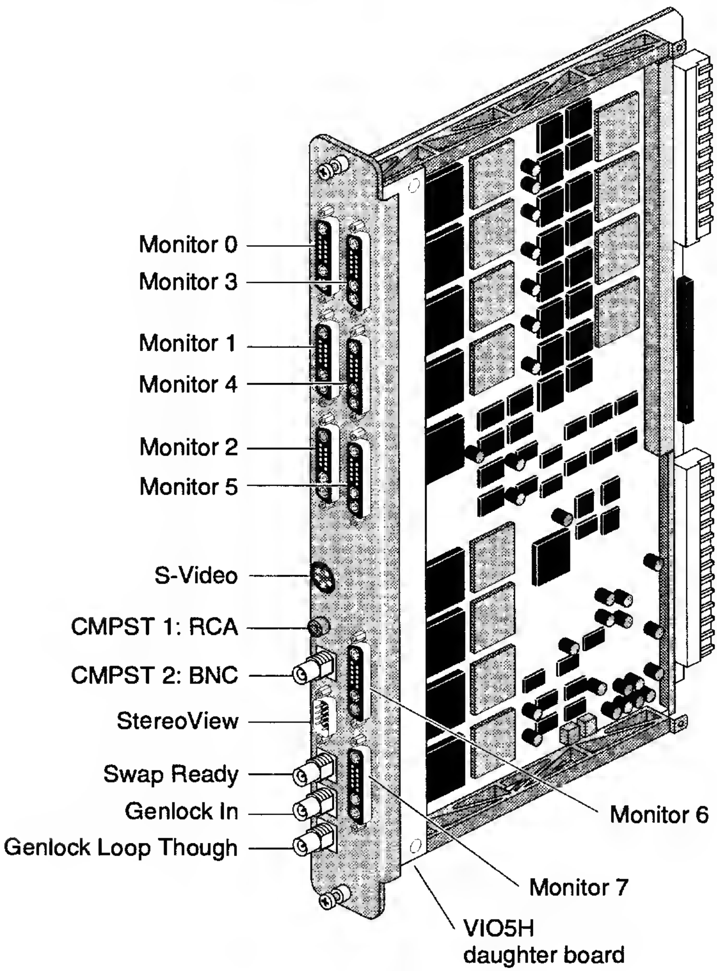


Figure 61. DG5 with Optional GVO Connectors

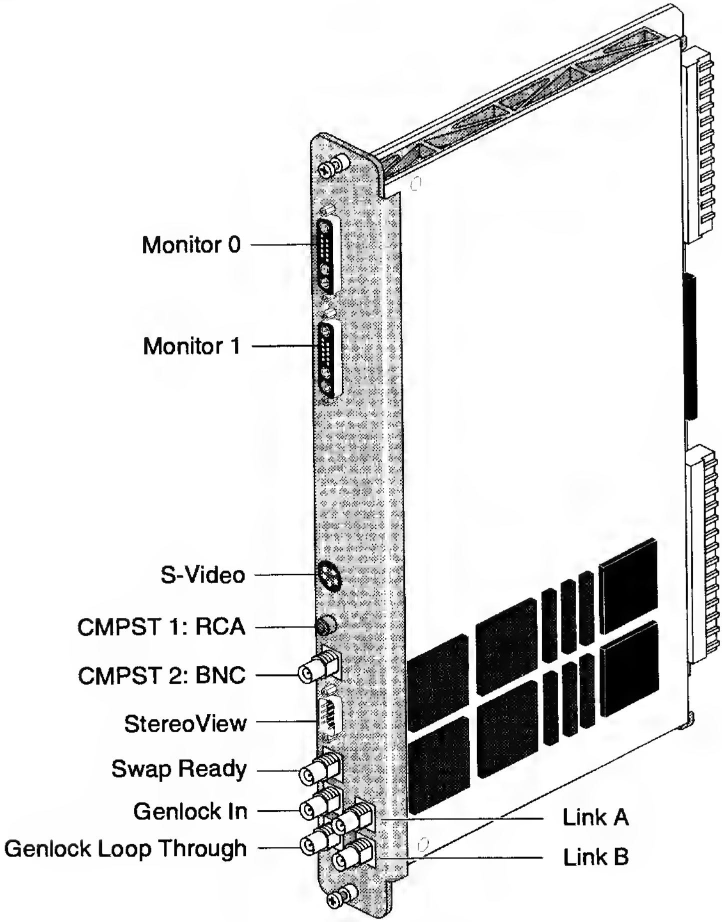


Figure 62. DG5-DPLEX Board

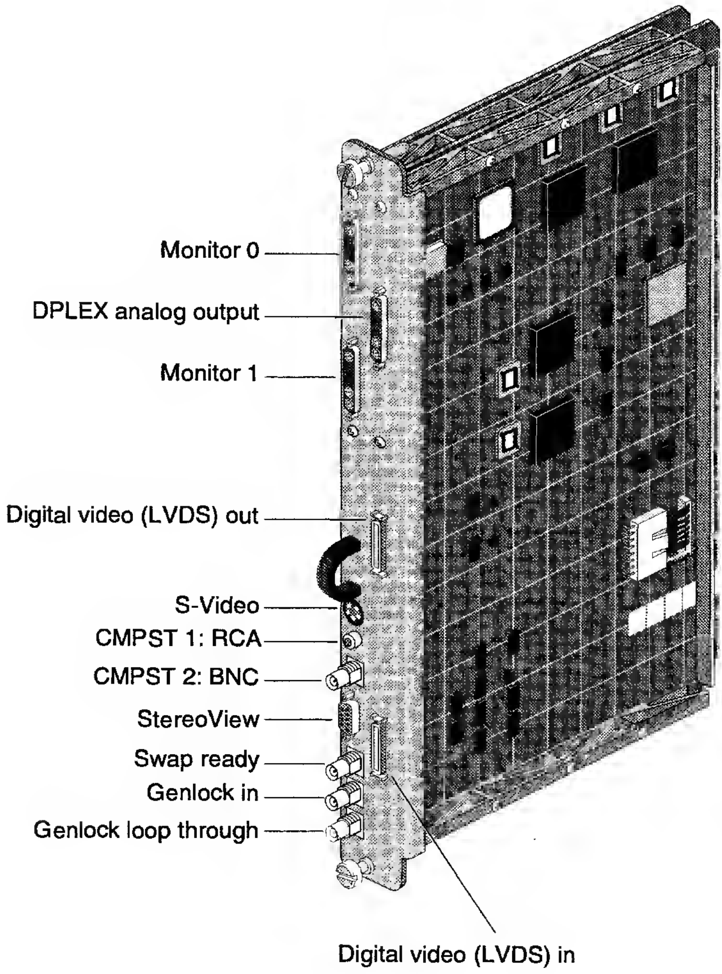


Figure 63. DG5-DD02 Board

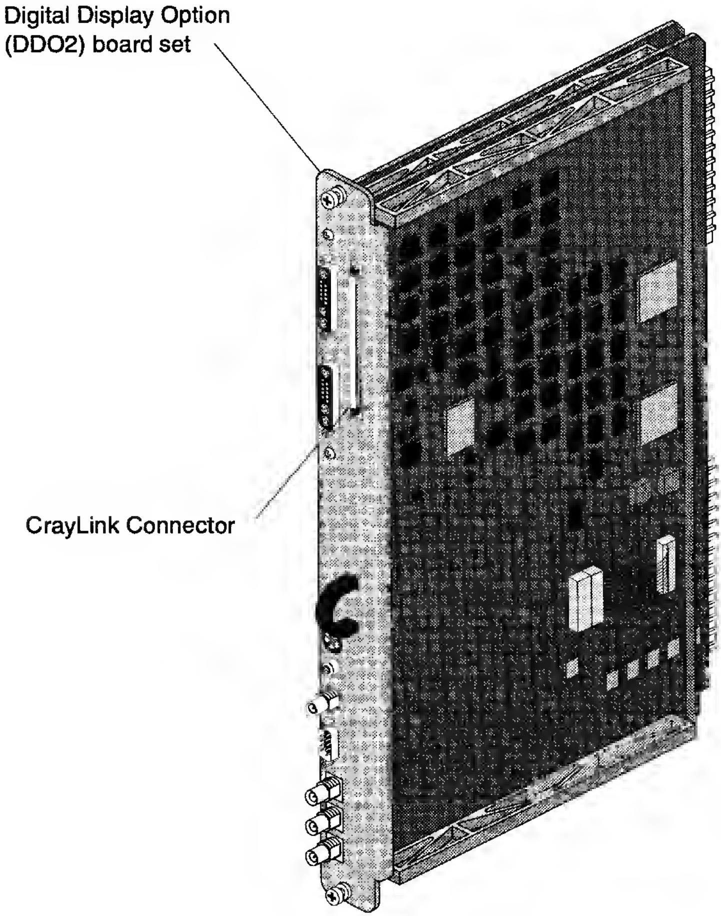


Figure 64. Onyx2 Monitor Cabling

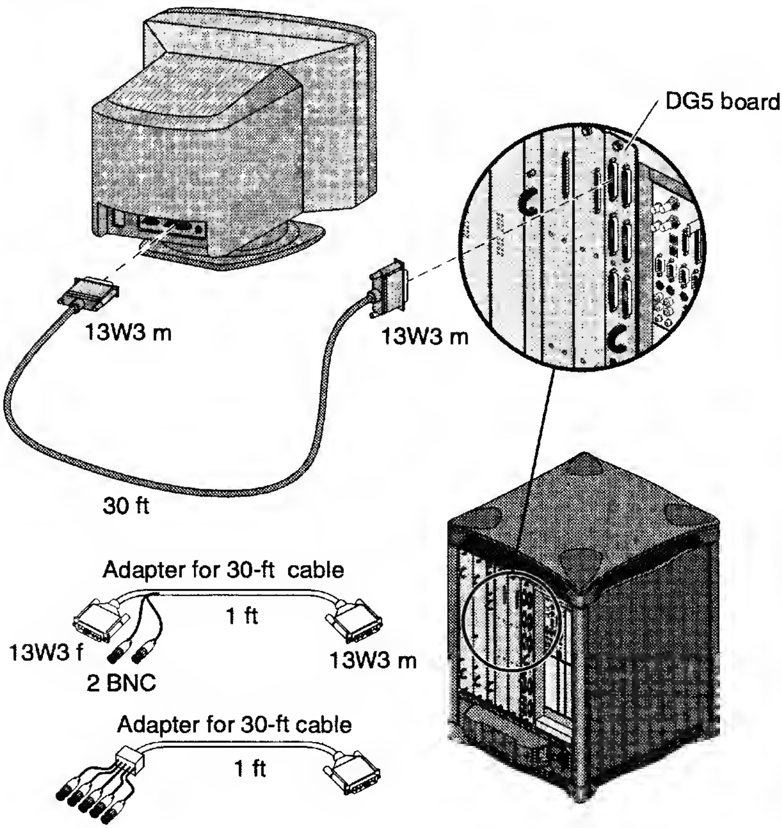
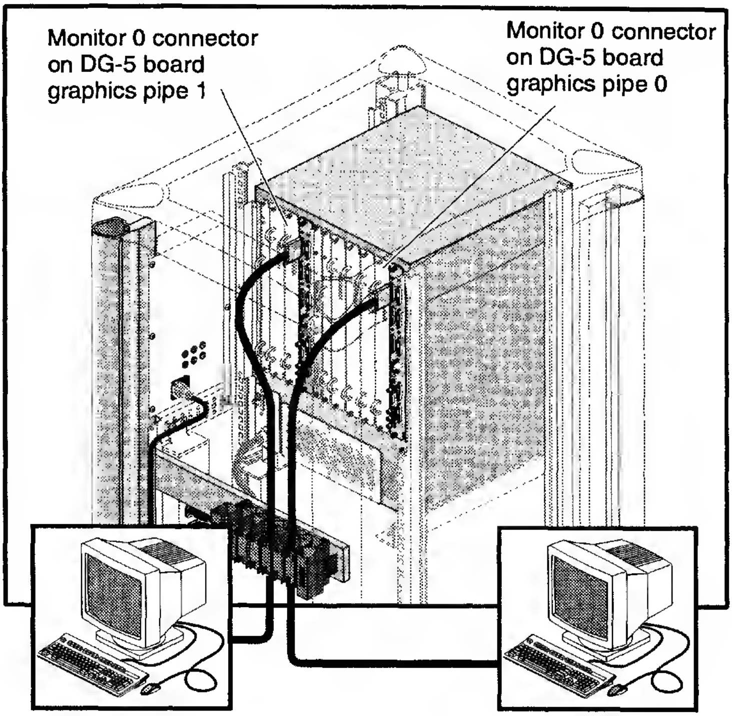


Figure 65. Connecting Monitors to a Multipipe Graphics Module



Configuring Xtown on Onyx2 Systems

Xtown and IO6G control should be evenly distributed between the available nodes in each 8P/12 module and their XBOW ASICS.

As shown in Table 41, nodes N1 and N3 control XIO slots 1 through 6 using XBOW0. Nodes N2 and N4 control XIO slots 7 through 12 using XBOW1.

In 4-node systems, each node controls a different quadrant of XIO slots. In addition, an IO6G in XIO slot 1 supports two keyboard and mouse connections.

Table 41. Configuring Xtown Boards on Onyx2 Systems

2-Node Systems	
Node Number	**XIO Slot(s)**
N1	IO1 – IO6
N2	IO7 – IO12

4-Node Systems	
Node Number	**XIO Slot(s)**
N1	IO1, IO3, IO5
N2	IO7, IO9, IO11
N3	IO2, IO4, IO6
N4	IO8, IO10, IO12

Note: XIO slot IO2 is reserved for the PCI expansion option.

Table 42 shows the optimal Xtown placement for an 8P/12 module. Remember that you cannot configure more than one graphics pipe per node.

Table 42. Optimal Xtown Board Placement for an 8P/12 Module

Nodes	Pipes	Node Placement	Xtown Placement	Comments
1	1	N1	IO2-IO6	N1 controls IO6G on XBOW0 Use IO2, IO3, IO4, IO5, or IO6 for Xtown
2 – 4	1	N1, N2, N3, N4	IO7	N1 controls IO6G on XBOW0. N2 controls IO7 on XBOW1
2	2	N1, N2	IO2, IO7	N1 controls IO6G and IO2 on XBOW0 N2 controls IO7 on XBOW1
3	2	N1, N2, N3	IO2, IO7	N1 controls IO6G and IO2 on XBOW0 N2 controls IO7 on XBOW1
4	2	N1, N2, N3, N4	IO2, IO7	N1 controls IO6G and IO2 on XBOW0 N2 controls IO7 on XBOW1
3	3	N1, N2, N4	IO2, IO7, IO8	N1 controls IO6G and IO2 on XBOW0 N2 controls IO7 on XBOW1 N4 controls IO8 on XBOW1
4	3	N1, N2, N3, N4	IO2, IO7, IO8	N1 controls IO6G N2 controls IO7 on XBOW1 N3 controls IO2 on XBOW0 N4 controls IO8 on XBOW1
4	4	N1, N2, N3, N4	IO2, IO3, IO7, IO8	N1 controls IO6G and IO3 on XBOW0 N2 controls IO7 on XBOW1 N3 controls IO2 on XBOW0 N4 controls IO8 on XBOW1

XIO Board Configuration Rules

There are six rules, for both single-module and multiple-module systems, that must be followed when configuring the Origin 2000 system:

- Each Origin 2000 module must have at least one node board.
- There must be at least one BaseIO board in the system (whether you have a single-module or multiple-module system).
- The BaseIO board must be placed in slot IO1.
- If present, the PCI expansion must be placed in slot IO2 (beneath the BaseIO board).
- Neither a BaseIO board nor an MSCSI board can ever go into slot IO2.
- To use all 12 of the XIO slots in a module, there must be at least two node boards installed: one node board must be placed in an odd slot (either N1 or N3) and one node board must be placed in an even slot (either N2 or N4).
- Use the following guidelines for GSN, TPU, and HIPPI boards:
    - Move the boards out of slots 7 through 12–this moves heavy traffic off the node 2-to-crossbow link.
    - Move all such boards out of slots 1 and 8–these are main noise sources.
    - If any of these boards is installed in slots 7 through 12, move other I/O cards out of slots 1 and 8 (noise is reciprocal).

Caution: Remember, when installing the odd-numbered XIO boards 3 through 9 and the even-numbered XIO boards 4 through 10, each pair of boards must be inserted with their component sides facing toward each other, as shown in Figure 66 and Figure 67.

Table 43 shows the recommended XIO installation order.

Figure 66. XIO Board Slot Locations

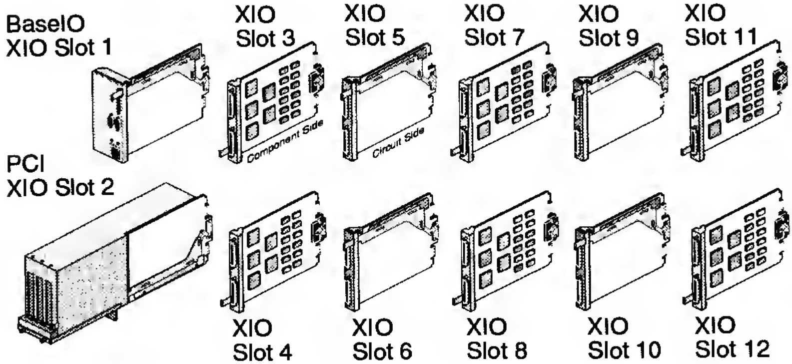
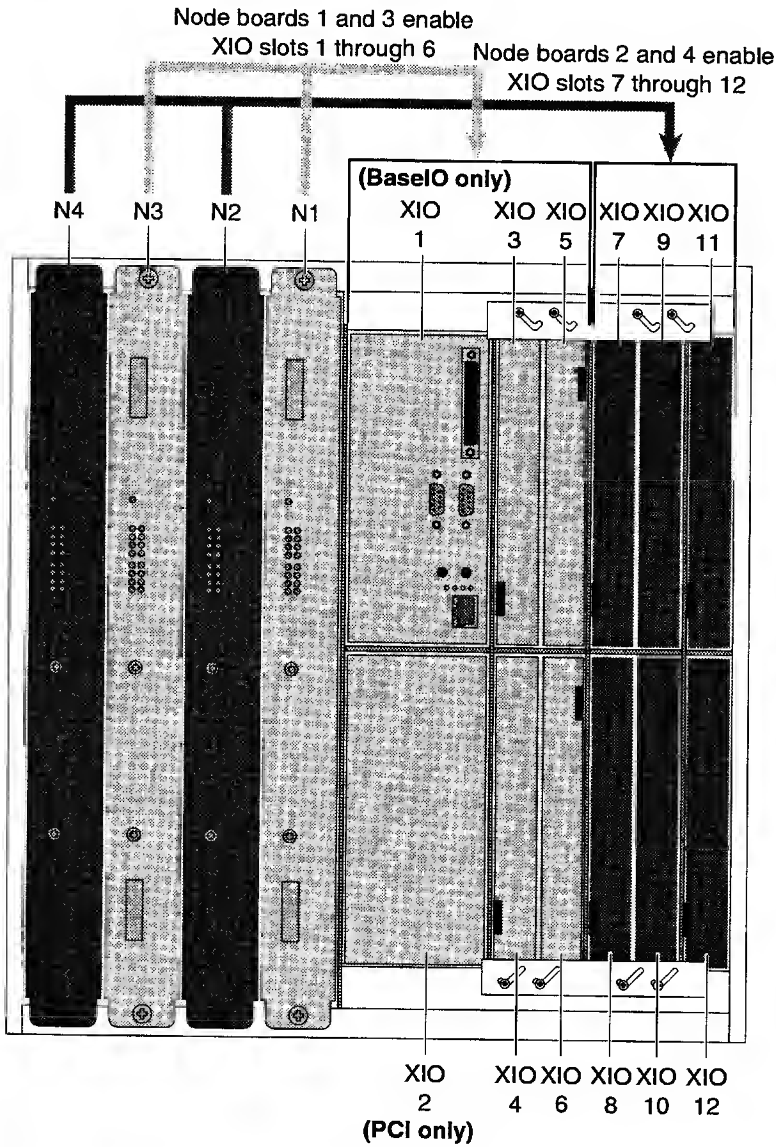


Table 43. Installing XIO Boards to Equalize Bandwidth and Control

Start Loading XIO Slots at:	Node Board that Controls the XIO Slot
IO1	Node 1
IO7	Node 2
IO2	Node 3
IO8	Node 4
IO3	Node 1
IO9	Node 2
IO4	Node 3
IO10	Node 4
IO5	Node 1
IO11	Node 2
IO6	Node 3
IO12	Node 4

Figure 67. XIO and Node Board Numbering



Command Interpreters

This section describes the following topics:

PROM Monitor

The command monitor is option 5 on the System Maintenance Menu. The command monitor (hereafter referred to as PROM monitor) is an interactive shell in which you can set and unset environment variables, display a hardware inventory, perform various tests, and load other programs (including the IRIX OS).

This section describes the following topics:

PROM Monitor Commands Numbering

- Module and node numbering starts with 1
- Base is decimal

PROM Passwords

The PROM password can be set to restrict operation of certain PROM modes. With a password set, an attempt to do anything other than a standard system boot requires that the password be reentered. The system "remembers" the password after the system is powered off.

Clearing the PROM Password

Log in as root and use the `nvram` command to set the `passwd_key` variable to a null string by entering **`nvram passwd_key ""`**

Setting the PROM Password from the PROM Monitor

If you wish to set your PROM password from within the PROM monitor, perform the following steps:

1. Log in as root.
2. When you see the message:
   ```
   Starting up the system...\
   To perform system maintenance instead, press <Esc>
   ```
   Press Esc to display the System Maintenance Menu
3. Enter **`passwd`** and then enter a password. Whenever you access the PROM monitor, you will be required to enter this password.

Hidden PROM Monitor Commands

Table 45 on page 140 describes hidden PROM monitor commands.

Table 44. PROM Monitor Commands

Command	Description
`auto`	Autoboots the machine.
`boot [-f` *file*`] [-n] [`*args*`]`	Loads and executes the specified file.
`enable -m` *X* `-s` *Y* `-MEM` *Z* `disable -m` *X* `-s` *Y* `-MEM` *Z* *X* = module ID in decimal *Y* = node board slot ID (n1, n2, etc.) *Z* = bank number (0 through 7) Example: `enable -m 3 -s n1 -MEM 0`	Enables and disables a memory bank. Memory is marked disabled at the DIMM-bank level. That is, if you replace a disabled DIMM without enabling its bank, the new DIMM will remain disabled. Also, if you move a node board, any "disabled" DIMM banks will still be disabled in the new location. Note that memory bank 0 cannot be manually disabled. The only situation in which bank 0 may be disabled is if all the following conditions are true: - Bank 0 is identified as bad during the boot process. - There is memory present in bank 1. - The memory in bank 1 is not already disabled. If all three of these conditions are true, the system disables bank 0, swaps bank 0 with bank 1, and runs from bank 1.
`enable -m` *X* `-s` *Y* `-CPU` *Z* `disable -m` *X* `-s` *Y* `-CPU` *Z* *X* = module ID in decimal *Y* = node board slot ID (a, b, ab) *Z* = CPU (a or b)	Enables or disables a CPU at the node-board level. That is, if a node board has a CPU marked "disabled" and you move that node board, that CPU will remain disabled in the new location.
`enable -m` *X* `-s` *Y* `-PCI` *Z* `disable -m` *X* `-s` *Y* `-PCI` *Z* *X* = module ID in decimal *Y* = slot ID (IO1, IO2, etc) *Z* = PCI ID (0 through 7)	Enables or disables a PCI bus at the BaseIO level. That is, if you move the BaseIO board, the "disabled" entry moves with it.
`exit`	Returns the PROM menu.
`help [`*command*`]`	Prints descriptions for one or all commands.
`init`	Reinitializes the PROM log.

Table 44. PROM Monitor Commands (continued)

Command	Description
`hinv [-g] [-v] [-m] [-mvvv] [-l` *<path>*`]`	Displays hardware inventory. The `hinv` output is slightly different than the output produced under the IRIX OS.
`ls [`*device*`]`	Displays a directory listing for a device.
`modnum`	Lists all the current modules in the system.
`mvmodule` *new_moduleid old_moduleid*	Changes module ID.
`off`	Turns off the power. Supported only on a subset of systems with software power control.
`opts`	Displays the build options.
`passwd`	Sets the PROM password.
`ping [-f] [-i` *sec*`] [-c` *num*`] target`	Pings server at target.
`printenv [`*env_var_list*`]`	Displays environment variables. `printenv root` Displays the current setting of the root variable.
`reset`	Resets the machine.
`resetenv`	Resets the environment to standard values.
`resetpw`	Resets the PROM password.
`setenv` *env_var string*	Sets the value of an environment variable.
`setpart`	Makes partitions.
`single`	Boots IRIX OS to single-user mode.
`unsetenv` *env_var*:	Removes a variable from the environment.

Table 44. PROM Monitor Commands (continued)

Command	Description
`update`	Stores the current system inventory in the BaseIO NVRAM. Every time the system is powered on, the current system inventory is compared with this stored inventory. If differences are detected, they are displayed to the console and the message "CHK INV" is shown on the MSC alphanumeric display. If you see such a message and there has been no change in the system configuration, you should try to determine why the current inventory is different from the stored inventory. If there has been an intentional change to the system configuration, you can eliminate the `Chk Inv` message by running the `update` command at the PROM monitor prompt.
`version`	Displays PROM version.

Table 45. Hidden PROM Monitor Commands

Command/Syntax	Description
`pod`	Puts the system into the POD command interpreter
`serial`	Prints encrypted versions of the serial numbers that it detects in the midplane serial number NICs
`flashio`	Downloads new microcode to the BaseIO PROM
`flashcpu`	Downloads new microcode to the IP27 PROM
`checkclk`	Checks the node/midplane clock speed
`enableall -y -list`	Enables all disabled components. The `-show` option lists the disabled components.

PROM Monitor Environment Variables

Table 46 lists and describes the PROM monitor environment variables. Enter the `printenv` command to display the current settings for the PROM monitor environment variables.

Table 46. PROM Monitor Environment Variables

Variable	Description
`AutoLoad`	If set to **yes**, boot IRIX into the run level that is specified in the `/etc/inittab` file if the system is reset or powered on. If set to `no`, stop the boot process at the PROM monitor main menu and wait for user intervention.
`console`	If set to `d`, the system console is communicating with the `tty0` port of the BaseIO board that is selected by the `ConsolePath` variable. If set to `g`, the console is communicating with the graphics console without the Silicon Graphics logo.
`ConsoleIn`	Displays the hardware graph pathname that indicates where the console is connected when the `console` variable is set to `d`.
`ConsoleOut`	Displays the hardware graph pathname that indicates where the console is connected when the `console` variable is set to `d`.
`ConsolePath`	Displays the hardware graph of the pathname that is communicating via the BaseIO console port.
`CPUfreq`	Defines CPU speed.
`dbaud`	Specifies the baud rate for the BaseIO `tty0` console port.
`dbgtty`	Displays the `tty` port that the kernel debugger `symmon` communicates with when active.
`diskless`	If set to `1`, specifies that the system is diskless and must be booted over the network. This is not supported for Origin series systems and should always be set to `0`.
`gConsoleIn`	Displays the hardware graph pathname that indicates where the console is connected when the `console` variable is set to `g`.

Table 46. PROM Monitor Environment Variables (continued)

Variable	Description
gConsoleOut	Displays the hardware graph pathname that indicates where the console is connected when the console variable is set to g.
netaddr	Specifies the network address when a system is booted across an Ethernet network.
nonstop	If set to 1, the boot process halts when missing hardware is detected and prompts the user to continue with the boot process. For Origin series systems, this is not supported. This variable should always be set to 0. A message is displayed to alert the user about the missing hardware and the boot continues.
OSLoader	Specifies the operating system loader. For IRIX, this is sash. This variable is stored in nonvolatile RAM but is normally left unset, which allows the PROM to automatically configure it at system power-on.
OSLoadFilename	Displays the filename of the operating system kernel. For IRIX this is /unix.
OSLoadPartition	Displays the disk partition where the operating system kernel is located.
ProbeAllScsi	If set to y, the power-on diagnostics probe and test all BaseIO SCSI buses. If set to n, the power-on diagnostics probe and test only the console BaseIO SCSI buses.
rbaud	Defines the baud rate for a remote terminal on the ttyd2 port (not supported for Origin series systems).
root	Defines where the root filesystem is located. You must use the hardware graph device name when the system disk is on a SCSI bus other than 0 or 1.
sgilogo	If set to y, displays the Silicon Graphics logo on the MMSC display when PROM monitor 5 menu choice is selected. If set to n, the Silicon Graphics logo is not displayed.
SystemPartition	Displays the disk partition where the OS loader (sash) is located.
volume	Specifies the system speaker volume.

Environment Variables Stored in Nonvolatile RAM

The environment variables that are stored in nonvolatile RAM are important variables that are not changed frequently. Table 47 lists and describes these environment variables.

Use the IRIX `nvram` command (you must be logged in as root) to set or list the environment variables that are stored in nonvolatile RAM.

Table 47. Environment Variables Stored in Nonvolatile RAM

Variable	Description
`netaddr`	Specifies the local network address for booting across the Ethernet.
`dbaud`	Specifies the diagnostics console baud rate. IRIX uses the dbaud rate for the diagnostics console during the system start-up. You can change the baud rate by setting this variable; the following baud rates are acceptable: 75, 110, 134, 150, 300, 600, 1200, 2400, 4800, 9600, and 19200. You can also change the baud rate by pressing the Break key. However, pressing the Break key changes the baud rate temporarily; the baud rate reverts to the value specified by the dbaud or rbaud variables when you press the reset switch or issue an init command.
`rbaud`	Specifies the remote console baud rate. The following baud rates are acceptable: 75, 110, 134, 150, 300, 600, 1200, 2400, 4800, 9600, and 19200.
`bootfile`	Specifies the file that is used during the autoboot process (normally a standalone shell [sash]). This variable is valid for pre-ARCS PROMs only. ARCS PROMs store this information in the OSLoader variable.

Table 47. Environment Variables Stored in Nonvolatile RAM (continued)

Variable	Description
`bootmode`	Specifies the type of boot for the pre-ARCS PROMs. ARCS PROMs store this information in the AutoLoad variable. There are three boot types: c - performs a complete cold autoboot by using the file that the bootfile variable specifies, boots sash, boots kernel, and runs power-on diagnostics m - (default) initiates the command monitor, clears memory, and runs power-on diagnostics d - initiates the command monitor, does not clear memory, and does not run power-on diagnostics
`autopower`	When set to y, allows systems that have software power control to automatically reset after a power failure.
`console`	Specifies which console to use. g - (default) use graphics console without the Silicon Graphics logo.
`keybd`	Specifies the keyboard type. The default is "df." Available settings depend on the exact PROM revision but may include some or all of the following settings: USA, DEU, FRA, ITA, DNK, ESP, CHE-D, SWE, FIN, GBR, BEL, NOR, PRT, CHE-F or US, DE, FR, IT, DK, ES, deCH, SE, FI, GB, BE, NO, PT, frCH on systems with the keyboard layout selector For some systems, JP is also acceptable to specify a Japanese keyboard.
`diskless`	Specifies that the system is diskless and must be booted over the network. For ARCS systems, diskless system environment parameters should be set as follows: diskless=1 SystemPartition=bootp()host:/path OSLoader=kernelname

Table 47. Environment Variables Stored in Nonvolatile RAM (continued)

Variable	Description
`monitor`	Specifies the monitor resolution for an unrecognized monitor brand on Indy systems. h or H = high-resolution monitor The default is a low-resolution monitor.
`nogfxkeybd`	When set to 1, specifies that the keyboard connection is not required.
`notape`	Specifies that no tape drive is attached to the system. If a tape drive is attached to the system, set this variable to 1 so that you can access a tape drive on another system on the network.
`volume`	Specifies the system speaker volume numerically.
`pagecolor`	Specifies the background color of the textport using a set of 6 hexadecimal RGB values.
`prompoweroff`	When set to y, this variable specifies that the system should return to the PROM monitor before powering off during the shutdown operation. **Note:** This variable is for Indy systems only.
`rebound`	When set to y, specifies that the system should automatically reboot after a kernel panic.
`sgilogo`	If set to y, specifies that the Silicon Graphics, Inc. logo and related information is displayed on the PROM monitor graphical screen.
`diagmode`	Specifies the mode of power-on diagnostics. When set to v, the diagnostics are verbose and extensive.

Device Naming

Because Origin systems can have thousands of devices, a more intuitive device naming convention was created:

The Hardware Graph command (`hwgraph`) provides the following functions:

- Describes a large number of devices in an easy-to-understand format
- Shows the hardware connectivity of each device
- Is visible to the user as a UNIX filesystem
- Structure is located in `/hw` with links into the `/dev` directory
- Structure is created by `/dev/MAKEDEV` and `/etc/ioconfig`
- Occupies no user disk space (it is a pseudo filesystem)

For example:

`/hw/module/1/slot/io1` = The BaseIO board of module 1

`/hw/module/2/slot/r1` = The R1 router of module 2

Displaying the System Hardware Inventory

You can display the system hardware configuration and inventory by using the `hinv` command, which has the following command line options:

- `-v` provides verbose output: NASIDs, processor speed, memory configuration, etc.

  Note: Under IRIX, the -v option does not list the NASIDs.

- `-m` displays hwgraph filename and physical manufacturing information: board name, part number, and barcode number
- `-mvvv` provides more verbose manufacturing information
- `-g` displays the `hwgraph` filename only

MMSC Commands

The multimodule system controller (MMSC) is used to control a single Origin 2000 rack system. When multiple Origin 2000 racks are attached to form a single system, each rack has its own MMSC; these MMSCs are then attached to each other via a private Ethernet connection.

Each MMSC can accept commands from several different sources, including consoles connected via direct serial connection or modem, individual modules (or "bays") in a rack via their module system controllers (MSCs), and other MMSCs via the private Ethernet network.

In addition, one rack in an Origin 2000 rack system provides a display and control panel to generate commands.

This section describes the following topics:

MMSC Command Interpreter Numbering

- Numeric parameters are entered in hexadecimal format
- Module numbering and rack numbering start with 1

Specifying Destinations

Many commands are intended for specific racks (MMSCs) and/or bays (MSCs). At the lowest level, racks are addressed with small integers starting with 1, and MSCs within a rack (bays) are addressed with letters that represent their position (U and L for Origin 2000 systems).

Modules are also addressed with small integers. They are written in hexadecimal format and may be in the range of 1 through FF. The mapping of a module number to its physical (rack and bay) location is arbitrary and is handled by the BaseIO (IO6) PROM. The MMSC obtains this mapping from the IO6 PROM when either the MMSC or the MSC is reset. It can also rebuild the current mappings by using the `scan` command (refer to Table 48 for a description of the `scan` command).

One or more addresses can be combined into a list. A list of rack or module addresses is made up of one or more addresses or ranges separated by commas with no intervening white space. A range, in turn, is a series of contiguous values specified as two addresses separated by a hyphen, again with no intervening white space.

The following list shows valid module address ranges:

```
1
1,3
1-f
1-4,7,39-3b
```

Origin 2000 Rack and Module IDs

The recommended Origin 2000 rack ID configuration uses sequential numbering from left to right, beginning with the left rack as Rack 1.

> Type `r * mod` at the MMSC prompt to verify that the racks are numbered according to the number scheme that Figure 68 illustrates.

Figure 68. Recommended Rack IDs

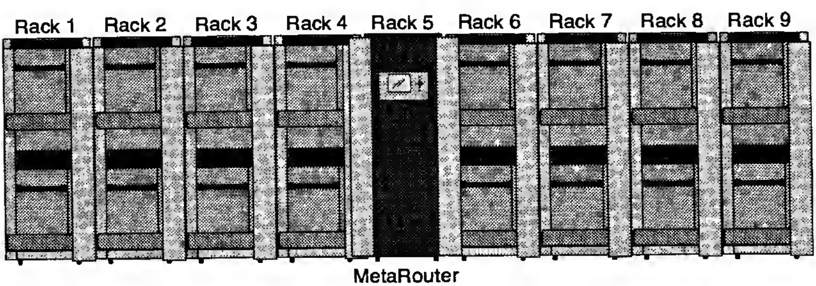
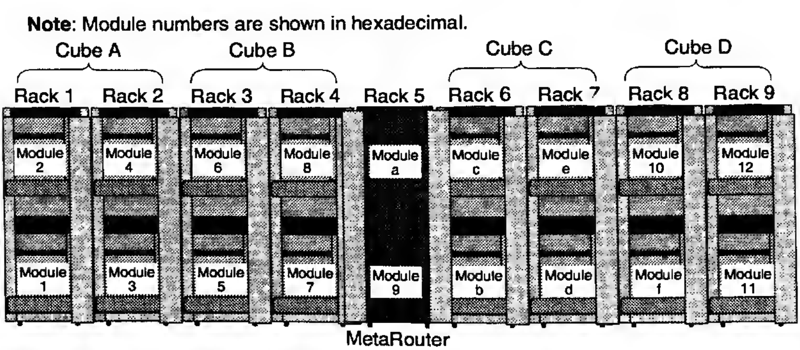


Figure 69 shows the recommended module IDs.

Figure 69. Recommended Module and Cube IDs



Physical Destinations

A physical destination is used to refer to one or more specific bays without regard to any logical module or partition designations. These are typically used for maintenance commands, such as powering a single bay off to replace a board. A complete physical destination specification consists of one or more "rack/bay pairs" such as:

`rack` *rlist* `bay` *blist*

The keywords *rack* and *bay* can be abbreviated as `r` and `b`, respectively.

A rack/bay pair selects each of the bays in *blist* on each of the racks in *rlist* (think of it as a "product"). Specifying more than one pair extends the selection accordingly (think of it as a "sum"). Thus, the long destination:

```
rack 1-3 bay u,L    rack 4 bay L
```

is equivalent to:

```
rack 1 bay u
rack 1 bay l
rack 2 bay u
rack 2 bay l
rack 3 bay u
rack 3 bay l
rack 4 bay l
```

Instead of specifying a list of addresses for *rlist* or *blist*, it is also possible to use one of several keywords. These include:

`rack all` (abbreviation: "*")

Selects all known online addresses. Addresses that are offline are skipped. The `scan` command can be used to update addresses if the MMSC's online or offline selections seem to be inaccurate. A command can still be directed to a suspected offline address by explicitly specifying the address rather than using the `all` keyword.

`rack local` (abbreviation: ".")

Valid only for *rlists*. Selects the local rack (the rack containing the MMSC that initially accepts the command).

Logical Destinations

A logical destination is used to refer to individual modules by their module numbers rather than their physical position in a system. Logical destinations are used more commonly than physical destinations, because these are the types of addresses that the IRIX OS and the various PROMs use. A logical destination is formed by the keyword module followed by a module list:

`module` *mlist*

The keyword `module` can be abbreviated as `m`.

As with physical destinations, *mlist* can be replaced with the keyword `all` to specify all known modules. There is a special case when no modules happen to be defined (this might occur, for example, if none of the modules have been powered on). In this case, `module all` is equivalent to `rack all bay all`.

Many MMSC commands pertain only to a particular rack and not to a specific bay within it. If a logical address is used with such a command, the bay address that is implied by the module number is ignored.

Both physical and logical destinations may be specified for the same command, as long as one is not embedded in the middle of another. Thus, the following destinations are valid:

```
rack * bay u module 3,5-7
r 1 b ul    m 8   rack 3 bay 1
```

But, the following destinations are not valid, or at least they may not give the expected results:

`rack module 2 3 bay u` (invalid)
`r 3 m 5 b u` (valid, but equivalent to: `r 3 b *   m 5   r . b u`)

When the MMSC executes a command, logical destinations are converted internally to physical destinations using the module number mappings known to the MMSC at the time the command is executed. If the MMSC's mappings are no longer valid (for example, if a module has been powered off or is otherwise unavailable), then commands may time out or be directed to the wrong module. The `scan` command can be used to update the MMSC's mappings manually if necessary.

MMSC Command Syntax

In general, an MMSC command looks like this:

[*escape*] [*dest*] *command* [*args*] <CR>

where:

[*escape*]

An MMSC-escape character, typically Ctrl-t by default. This is used in console and MSC pass-through modes to indicate the beginning of an MMSC command (refer to Table 48).

[*dest*]

A destination spccification.

Note: When you do not specify a destination, the command uses the current destination. To determine the destination of the command, enter `mod` to list the destination by module numbers, or `dest` to list the destination by a rack-bay number.

command

An MMSC command (refer to Table 48). An MMSC command may be in uppercase or lowercase (or both); it will be converted to uppercase by the MMSC before it is processed. If the command is not a valid MMSC command, it is assumed to be an MSC command and is passed to the bay(s) addressed by *dest* along with *args* without further translation.

args

Zero or more arguments to command.

<CR>

The Enter key.

Note: If an MMSC-escape character is entered in the middle of an MMSC command, all characters between it and the last MMSC-escape character are discarded. This can be useful if the current state of a console or other MMSC connection is currently unknown. In addition, the kill character (Ctrl-u by default) also discards the characters on the command line.

Command Descriptions

Table 48 provides a basic description of the MMSC commands. Many commands accept an optional destination specification, while others do not. Those commands marked with an asterisk (*) will invoke destination specifications. Unless otherwise specified, each of these commands can be executed by a user at the basic authority level.

Note: The system interprets commands that are not included in Table 48 as MSC commands.

Table 48. MMSC Commands

Command/Syntax		Description
`authority` [*level* [*pw*]]		Changes the authority level of the console to *level*; valid authority levels are: basic, supervisor, or service; the supervisor and service levels may require a password [*pw*]. If *level* is not specified, then the authority level associated with the console from which the command was entered is displayed.
`bs` *char*		Sets the backspace character to *char*.
`bs ?`		Prints the current backspace character.
`cecho [on|off]`		Turns echoing on and off.
`cecho`		Toggles echoing.
`com` *port*	*	Sets or displays communication settings of *port*; *port* is a number (1 to 6) that corresponds to a serial port on the addressed MMSC.
`com` *port* `cmd on|off`	*	Indicates whether MMSC and MSC commands are accepted from *port*. The `cmd` subcommand is used to indicate whether or not MMSC and MSC commands are accepted from a port. If off is specified, then MMSC and MSC commands are not accepted from the port. Ensure that at least one console has specified `cmd on`. The default setting for ports 1 and 5 is `cmd on`, and the default setting for ports 2, 3, 4 and 6 is `cmd off`.

Table 48. MMSC Commands (continued)

<table>
<tr><th>Command/Syntax</th><th></th><th>Description</th></tr>
<tr><td><code>com</code> port <code>function</code> func</td><td>*</td><td>Sets the function of port to func. port is a number from 1 to 6 that corresponds to a serial port on the addressed MMSC:

<table>
<tr><th>port</th><th>Label</th><th>Default Function</th></tr>
<tr><td>1</td><td>CONSOLE</td><td>TERMINAL</td></tr>
<tr><td>2</td><td>UPPERBAY</td><td>UPPER</td></tr>
<tr><td>3</td><td>LOWERBAY</td><td>LOWER</td></tr>
<tr><td>4</td><td>BASEIO TTY1</td><td>SYSTEM</td></tr>
<tr><td>5</td><td>ALTERNATE CONSOLE</td><td>ALTCONS</td></tr>
<tr><td>6</td><td>TEST</td><td>DEBUG</td></tr>
</table>

<table>
<tr><th>func</th><th>Function of Associated Port</th></tr>
<tr><td>TERMINAL</td><td>Communication with the user terminal device.</td></tr>
<tr><td>UPPER</td><td>Communication with the MSC in the upper bay of the rack.</td></tr>
<tr><td>LOWER</td><td>Communication with the MSC in the lower bay of the rack.</td></tr>
<tr><td>SYSTEM</td><td>Communication with the operating system. This is the port through which both the user and the MMSC communicate with the IRIX OS. It is typically connected to the TTY1 port on the master BaseIO board.</td></tr>
<tr><td>ALTCONS</td><td>Remote service port. This would typically be connected to a modem that is used for communication with an SGI service center.</td></tr>
<tr><td>DAEMON</td><td>Communication with a system controller daemon in the IRIX OS, such as <code>ffscd</code>. Such a daemon would ordinarily be used for generating bar graph data on the MMSC display. This port would typically be connected to a second serial port on the BaseIO board.</td></tr>
<tr><td>DEBUG</td><td>MMSC debugging log. This is useful if some sort of MMSC error has occurred. In that event, the debugging log may contain additional information.</td></tr>
</table>

Any given function can be assigned to only one port. If func specifies a function that is already assigned to another port, then the other port will have its port changed to an “unassigned” state.

Ensure that, at a minimum, the TERMINAL function is assigned to a port, or else it may become impossible to communicate with the MMSC.</td></tr>
</table>

Table 48. MMSC Commands (continued)

Command/Syntax		Description
com *port* oob on|off	*	Indicates whether the out-of-band (oob) data that is received from *port* should be intercepted and processed. The default setting for all ports is oob off.
com *port* rxbuf|txbuf *value*		Changes the size of the serial port's receive and transmit buffers, respectively. The default for both is 4096. If the system and terminal ports have different speeds, it may be necessary to increase the size of the transmit buffer on the slower port or else data may be lost. In extreme cases, serial buffer overflows have been known to disable a serial port, so underestimating the buffer size should be carefully avoided.
com *port* speed *baudrate*	*	Sets the baud rate of *port* to *baudrate*; valid baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200.
console		Enters console pass-through mode.
console [*args*]		Sends *args* to the system console port.
cons [*args*]		Sends *args* to the system console port.
dest		Displays or modifies the current default destination.
direct		Enters the direct-input mode on the current console and the other console on the MMSC.
downloader disable|enable	*	Sets or clears the "serial downloader" flag in the addressed MMSC's initialization PROM. The command downloader enable will set the flag, and the command downloader disable will clear it.
end *char*		Sets the end-of-command character to *char*.
end ?		Prints the end-of-command character.
esc *char*		Sets the MMS-escape character to *char*.
esc ?		Prints the MMS-escape character.
exit		Leaves the current input mode and returns to the console mode.
exit *char*		Sets the exit character to *char*.
exit ?		Prints the current exit character.

Table 48. MMSC Commands (continued)

Command/Syntax		Description
`flash [from system]`	*	Flashes a new firmware image into nonvolatile storage on the MMSC for the addressed rack; image is from the console of a running IRIX system.
`flash from console`		Flashes a new firmware image into nonvolatile storage on the MMSC for the addressed rack; image is from the terminal device.
`hwflow on|off` *port*		Enable or disable hardware flow on port.
`help [`*cmd*`|ALL]`		Displays information about the MMSC commands.
`kill` *char*		Sets the kill character to *char*.
`kill ?`		Prints the current kill character.
`log [`*log*`] clear`	*	Discards the contents of *log*. *log* — Function MSC — A log of all messages end other output from the addressed MSC. SYSTEM — A log of all output generated by the BeselO boerd on the eddressed rack, typically consisting of output from the operating system. TERMINAL — A log of all output that has been sent to the main terminal console ettached to the addressed rack. This is different from the SYSTEM log in thet it mey contein output from the MSC and MMSC input modes, as well as any messages received from an MSC (typically generated by various PROMs during system initielization). ALTCONS — A log of all output that hes been sent to the elternate console attached to the addressed rack. DEBUG — A log of ell debugging messages produced by the eddressed rack's MMSC. DISPLAY — A log of each command issued by the MMSC display end the corresponding response.
`log [`*log*`] dump [`*num*`]`	*	Dumps *log*; *num* indicates how many lines to dump.
`log [`*log*`] disable|enable`	*	Enables or disables the logging of data to *log*.
`log [`*log*`]info|?`	*	Displays information about *log* (size, enable/disable state, etc.)
`log [`*log*`] lines` *num*	*	Sets the maximum size of *log* to *num* lines.

Table 48. MMSC Commands (continued)

Command/Syntax		Description
`log [`*log*`] length` *num*	*	Sets the average line length of lines in *log* to *num*.
`mmsc`	*	Enters MMSC mode.
`mmsc [`*args*`]`	*	Sends *args* to the MMSC on the addressed rack(s).
`mmsg [on|terse|off]`	*	Controls the display of unsolicited messages from the MSC. `on` - Echoes all messages to the input source of their own lines `off` - Discards messages silently `terse` - Echoes messages but omits the identifying prefix
`mmsg rack [`*rackid*`]`	*	Sends messages generated by the addressed racks to the TERMINAL device that is attached to the rack specified by *rackid*.
`mmsg altrack [`*rackid*`]`	*	Specifies a rack whose ALTERNATE console port should receive unsolicited messages.
`msc`	*	Enters MSC mode.
`msc [`*args*`]`	*	Sends *args* to the MSC on the addressed bay(s).
`nap_time`		Prints the current nap interval.
`nap_time [`*value*`|default]`		Sets the console's nap interval to *value* or to the system default value.
`options [`*value*`]`		Sets the option flags to *value*.
`pager{back|fwd|quit}` *char*		Controls the built-in pager used by the MMSC to display large blocks of output. `back` - *char* specifies the character that is used to scroll backwards `fwd` - *char* specifies the character that is used to scroll forward `quit` - *char* specifies the character that is used to discontinue the output
`pager [`*info*`|?]`		Displays information about the current pager settings.
`pager lines` *val*		Sets the number of lines in a single page output to *val*.

Table 48. MMSC Commands (continued)

Command/Syntax		Description
`pager {on|off}`		Turns paged output on or off.
`password {set|setmmsc}` *passwd*	*	Changes the password. The password may be one of the following values. ***passwd*** — Corresponding password msc — The password on the addressed MSC. This is the same password that is specified with the MSC `pas` command. This will be passed along to the MSC before any restricted commands when the user is in the supervisor or service authority level. supervisor — The password used to enter the supervisor authority level. service — The password used to enter the service authority level. You must be at the service authority level to change this password.
`password unset` *passwd*	*	Removes the password.
`printenv` [*all*] `[default]`	*	Displays the names and values of any environment variables that have nondefault settings.
`rackid [`*value*`]`	*	Changes the rack ID of the addressed rack to *value*.
`rat`		Enters remote access tool (RAT) mode.
`reset_mmsc`	*	Restarts the addressed MMSC(s).
`reset_nvram`	*	Resets the contents of nonvolatile storage on the addressed MMSC(s) to default values.
`rmsg [on|error|off]`		Controls the echoing of responses to MSC or MMSC commands.
`scan`	*	Checks for the addressed MSCs and updates the internal table of module numbers to physical address mappings.
`setenv` *var* `[[=]` *value*`]`	*	Changes the setting of the environment variable *var* to *value*.
`steal`		Steals the system console.
`unsetenv` *var*	*	Restores the default value for environment variable *var*.
`unsteal`		Places the other device into console mode.
`ver`	*	Returns a string that indicates the MMSC firmware revision.

* = Command honors destination specifications

The MMSC intercepts certain MSC commands before it passes them to the MSC. Table 49 describes these commands.

Table 49. Intercepted MSC Commands

Command	Description
`pas s` *pw*	Converts the command into the MMSC command "`password set msc` *pw*"
`pwr u` `pwr d`	Forwards the command to the addressed MSCs, one rack at a time, with a time delay between each
`ver`	Prints the MMSC version string

Authority Levels

The MMSC restricts access to certain commands that could potentially interrupt the system. The authority level of the user determines which commands are available. Each authority level other than basic may have a password associated with it.

Note: When you do not have the proper authority level to execute a command, the `err perm` message displays on the screen.

Basic Authority Level
The user is allowed to display information and manipulate the console characteristics, but commands that affect system operation are not allowed. When a user with basic authority issues commands to an MSC, each command is preceded by the MSC command `pas`, which revokes any supervisor/diagnostic mode password that the MSC may have.

Supervisor Authority Level
Commands that affect the system are permitted. Each command entered by the user is preceded by the MSC command `pas` *pw*, where *pw* is the MSC password. This enables the user to issue restricted MSC commands if the MSC's keyswitch is not in the Diag position. These commands are followed by the MSC command `pas` to revoke the password.

Service Authority Level
Commands that affect the operation of the MMSC are permitted. This includes commands to reconfigure the MMSC serial ports and to change the various passwords. The service authority level is a superset of the supervisor authority level.

The `authority` command (refer to Table 48 on page 153) can be used to change the current authority level that is associated with a user on a console.

MSC Commands

MSC commands can be destructive to the system operation. Execute these commands only when the MSC is in supervisor mode or an `err perm` error occurs.

MSC Command Interpreter Numbering

- Numeric parameters are always in hexadecimal format
- All commands return OK and possibly some hexadecimal values
- Module numbering and node numbering start with 1

Place the MSC in supervisor mode by performing one of the following actions:

- Position the front-panel keyswitch to Diag.

  or
- Issue the pas *xxxx* command; the default MSC password is **none**.

MSC Command Syntax

- All MSC commands must be prefixed with Ctrl-t.
- Commands are visible only if echoing (ech) is on
- All commands are three characters long
- Some commands take parameters
- All numeric parameters are in hexadecimal format
- All commands return one of the following messages:
  - "OK"
  - "OK" and some hexadecimal value(s)
  - One of the three error responses listed in Table 50

Table 50. MSC Error Responses

Response	Description
err perm	Permission denied. The keyswitch must be in the diagnostic position, or a password must be entered using the pas command.
err cmd	Unrecognized command.
err arg	Preliminary invalid command argument(s).

Table 51 lists and describes the MSC commands.

Table 51. MSC Commands

Command	*	Description	Rstd
`aut`	*	Displays 0 if the system is not set to automatically power on and 1 if the system is set to automatically power on; the default is 0 (Origin 200 systems only).	
`aut 0`		Turns off automatic power-on mode (Origin 200 systems only).	
`aut 1`		Turns on automatic power-on mode; the MSC issues a `pwr u` command when you power on the MSC (Origin 200 systems only).	
`clr`	*	Resets all MSC options to their power-up defaults.	Yes
`dbg`	*	Displays the virtual and physical debug switch bytes.	
`dbg` *V P*	*	Sets the debug switch bytes to *V* and the physical debug switch bytes to *P*.	Yes
`dsp` *M*		Displays the message *M* on the 8-character alphanumeric display; *M* may contain up to 8 ASCII characters other than `NUL`.	
`dsc` *N C*		Modifies the *N* character on the MSC display; *N* is a digit from 0 to 7, and *C* is any ASCII character other than `NUL`. For example, `dsp testing` would overwrite the first seven characters with "testing," while leaving the eighth character unaltered.	
`ech 0`	*	Turns off echoing.	
`ech 1`	*	Turns on echoing.	
`ech`	*	Toggles echoing; by default, echoing is on after a reset.	
`fan`	*	Displays the status of the fans: `n` = normal-speed fan okay `h` = high-speed fan okay `f x` = bit map of failed fans for Origin 200 system `f xyz` = bit map of failed fans for Origin 2000 system	
`fan n`	*	Sets the fans to normal speed.	Yes
`fan h`	*	Sets the fans to high speed.	Yes
`key`		Returns the status of the key switch.	

Table 51. MSC Commands (continued)

Command	*	Description	Rstd
`mod`		Displays the number of the module that contains the MSC.	
`mod` *xx*		Sets the number of the module that contains the MSC to *xx; xx* is a hexadecimal number from 01 to ff.	Yes
`nmi`	*	Sends a hardware NMI to all node cards in the MSC's module.	Yes
`pas` *xxxx*	*	Places the MSC into supervisor mode; *xxxx* is the password.	
`pas s` *xxxx*	*	Set the password to *xxxx*. Passwords must be four characters.	Yes
`pwr`	*	Displays whether the system is powered up (u) or powered down (d).	
`pwr u`	*	Powers up the system.	Yes
`pwr d`	*	Powers down the system.	Yes
`pwr d` *N*	*	Waits *N* seconds and then powers down the system; *N* is a hexadecimal number from 5 to 258.	Yes
`pwr c` *N*	*	Powers down the system, waits *N* seconds, and powers up the system; *N* is a hexadecimal number from 5 to 258.	Yes
`rst`	*	Sends a hardware reset to all node cards in the MSC's module.	Yes
`rsw`	*	Returns the current debug switch settings as an inverted hexadecimal byte.	
`sel` *cpu*	*	Selects which CPU will receive the ACP input.	
`sel`	*	Displays which CPU currently receives the ACP input.	
`sel auto`	*	Selects the CPU that outputted information last as the CPU that receives the ACP input.	
`sel none`	*	Selects no CPU to receive the ACP input.	
`see` *cpu*		Displays the output from *cpu* (CPUs are designated by slot and A or B); discards all other output.	
`see`	*	Indicates which CPUs are being displayed.	
`see all`	*	Displays output from all CPUs.	

Table 51. MSC Commands (continued)

Command	*	Description	Rstd
tmp	*	Displays o, h, or n, to indicate the temperature status of the system: o = overtemperature h = high operating temperature n = normal operating temperature	
ver	*	Displays the MSC firmware revision number.	
vlm [3|5|v] [l|h|n]		Margins the power supply voltages and the midplane termination voltage by 5%: 3 = 3.45 Vdc for the power supply 5 = 5.0 Vdc for the power supply v = DC voltage for the midplane termination h = high-voltage margin l = low-voltage margin n = normal voltage margin	Yes

Rstd = Restricted

* = Origin 200 system MSC commands

MSC Alphanumeric Display Messages

The MSC has an eight-character alphanumeric display that can display a number of messages. Table 52 lists these messages and their definitions.

Table 52. MSC Alphanumeric Display Messages

MSC Message	Definition
CHK INV	The current system configuration is different from the stored hardware inventory. Run update from the command monitor to eliminate this message (refer to Table 44 on page 138).
E5E5E5E5	Hex code status for Node 1, Node 2, Node 3, and Node 4.
FAN FAIL	A system fan has failed. If it is fan 1, 2, or 3, the system shuts down. A service call should be placed as soon as possible.
HBT TO	The system has registered a heartbeat time-out. A nonmaskable interrupt is generated, followed by a system reset.
KEY OFF	The MSC keyswitch has been turned to standby.
M FAN FL	More than one fan has failed and the system has shut down.
MOD 1 C	Module 1 is driving the console.
NMI	The MSC keyswitch has been turned to the diagnostic position, and the nonmaskable interrupt (NMI) button has been pushed.

Table 52. MSC Alphanumeric Display Messages (continued)

MSC Message	Definition
OVR TEMP	The system temperature has exceeded acceptable limits and the system has shut down.
P*n* M*id*	P*n* where *n* is the partition number. M*id* where *id* is the module ID.
PFW FAIL	The input power has failed or dropped below acceptable parameters. The system has shut down.
POK FAIL	A power OK failure occurred on an unidentified board.
POK N 0	A power OK failure occurred on the first node board.
POK N 1	A power OK failure occurred on the second node board.
POK N 2	A power OK failure occurred on the third node board.
POK N 3	A power OK failure occurred on the fourth node board.
POK RT 0	A power OK failure occurred on the first router board.
POK RT 1	A power OK failure occurred on the second router board.
POWER UP	The system is being powered on from the front panel switch.
PS HITMP	The internal power supply unit is running at higher than normal temperatures.
PS OT FL	The system's power supply temperature has exceeded safety limits and the system has shut down.
PS FAIL	The internal power supply has failed and the system has shut down.
PSTMP OK	The power supply operating temperature is OK.
PWR CYCL	The MSC has received the command to power cycle from the console or a remote user.
R PWR DN	The system is being powered off remotely via the MSC serial connection (typically by an MMSC).
R PWR UP	The system is being powered on remotely via the MSC serial connection (typically by an MMSC).
RPWR FL	The redundant power supply has failed.
RPWR OK	Redundant power has been restored from nonredundant mode.
RESET	The MSC keyswitch has been turned to the diagnostic position, and the reset button has been pushed.
SP INT 1	The MSC's firmware generated a spurious timer interrupt signal.
SP INT 2	The MSC's firmware generated a spurious clock signal.
SYS OK	The system is operating normally.
TEMP OK	The system temperature is within normal operating parameters.

MSC Hardware Debug Switches

Two sets of debug switches are maintained in the MSC:

- Eight physical debug switches, numbered 1 through 8
- Sixteen virtual debug switches, numbered 1 through 16

These switches are set by the MSC `dbg` *xx yy* command, where *xx* and *yy* are hexadecimal bytes. The most significant bit of *xx* corresponds to debug switch 16, while the least significant bit of *yy* corresponds to debug switch 1. Using the `dbg` command without arguments displays the current settings. Ordinarily, both sets of hardware debug switches should be set to zero.

Note: SGI does not recommend that you change DIP switch settings. If you wish to modify the power-up sequence, use the virtual switches.

Origin 200 DIP Switches

To access the Origin 200 DIP switches, remove the small EMI panel from the side of the chassis. Origin 200 system DIP switches are slightly different in appearance from the Origin 2000 DIP switches. The DIP switches on both systems are OFF when they are switched away from the labeled numbers.

Origin 2000 DIP Switches

To access the Origin 2000 DIP switches, remove the front cover from the MSC (refer to Figure 70). To change the hardware debug switch settings, use a sharp stylus to press the switch in on the top (switch ON) or bottom (switch OFF). The debug switch block shown in Figure 71 has switches 1, 2, and 6 set to ON and all others OFF. Table 53 through Table 66 describe the DIP switches.

Figure 70. MSC Hardware Debug Switch Location

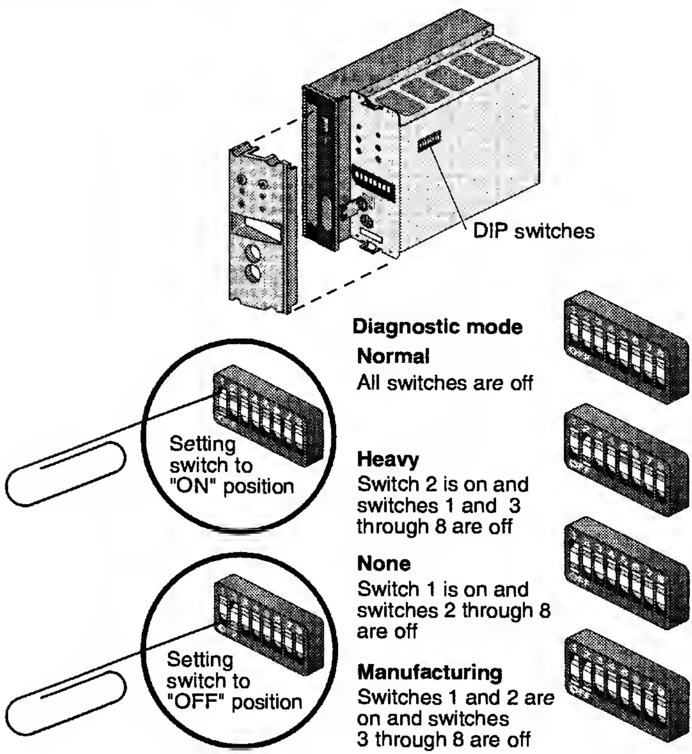


Figure 71. MSC Hardware Debug Switch Settings

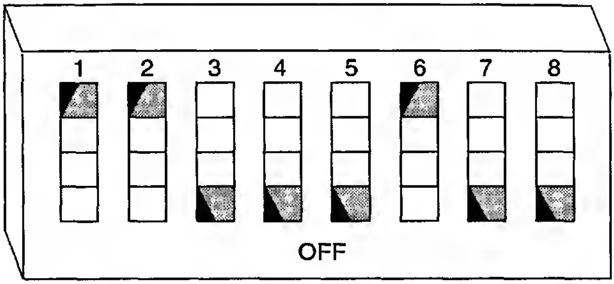


Table 53. Physical DIP Switches 1 and 2 - Diagnostic Level

1	2	Description
OFF	OFF	Normal - Tests each part of the system for basic functionality, using only relatively fast tests to expedite system boot while catching any blatantly troubled hardware.
OFF	ON	Heavy - Runs the most thorough diagnostics available on each part of the system. They may take a very long time to complete, especially the memory tests. It may be desirable to run them after installing new hardware or if the system is having problems thought to be hardware-related.
ON	OFF	None - Performs no diagnostics, and the system will boot as fast as possible. This might be used when debugging software such as kernel drivers, when there is complete confidence in the hardware.
ON	ON	Manufacturing - Runs extensive diagnostics and outputs special FRU (field replaceable unit) information. Console input and output are handled through the system controller port, which must be connected to Silicon Graphics manufacturing equipment.

Table 54. Physical DIP Switch 3 - Information Level

3	Description
OFF	PROM displays normal boot status messages.
ON	PROM displays very detailed information messages during boot, interspersed with the normal boot status messages. The switch applies only to the nodes in the module on which it is set.

Table 55. Physical DIP Switches 4 and 5 - Boot Stop Point

4	5	Description
OFF	OFF	Never - Allows the boot to proceed all the way through to the IRIX OS (default).
OFF	ON	Local - Enables the boot to proceed up to the point where it would normally load and jump to the BaseIO PROM. Instead of continuing, all nodes enter cached (Cac) POD Mode. If this switch is set on any module, it will be propagated to all modules.
ON	OFF	Global - Enables the boot to proceed up to the point where it would normally load and jump to the BaseIO PROM. Instead of continuing, the master node enters cached (Cac) POD Mode and all of the slaves enter the Slave Loop. If this switch is set on any module, it will be propagated to all modules.
ON	ON	Memoryless - Stops as soon as possible after setting up just the bare minimum portion of the system required to enter POD mode. All nodes enter dirty exclusive (Dex) POD Mode even if there is no local memory. **Caution:** If this switch is set to ON in one module, the system that contains the module will not boot properly.

Table 56. Physical DIP Switch 6 - Default Environment

6	Description
OFF	The PROM reads all PROM Log environment variables and IO6 NVRAM settings and does not use the system defaults.
ON	The PROM ignores all PROM Log environment variables and IO6 NVRAM settings and uses the system defaults. This may be useful for proceeding if any of the variable storage mechanisms contain data that is preventing the system from booting. This switch applies only to the module on which it is set.

Table 57. Physical DIP Switch 7 - Bypass IO6

7	Description
OFF	The PROM does not bypass the first IO6 card.
ON	The PROM bypasses the first IO6 card that is found and tries to boot from the second one found. This may help to boot the system if the first BaseIO board is not working, without having to physically remove the card. This switch applies only to the module on which it is set.

Table 58. Physical DIP Switch 8 - Bypass Global Master

8	Description
OFF	The node that would ordinarily become the global master will not become a slave.
ON	The node that would ordinarily become the global master will become a slave, and the next CPU in line will become the global master. This switch applies only to the module on which it is set.

Table 59. Physical DIP Switch 10 - Master or Slave Mode

10	Description
ON	The system is in master mode.
OFF	The system is in slave mode.

Note: This physical DIP switch is for the Origin 200 system only.

Virtual DIP Switches

Table 60. Virtual DIP Switch 9 - Override CPU Disabling

9	Description
OFF	CPUs can be disabled using the POD `disable` command.
ON	All CPUs that would otherwise be disabled due to the DisableA or DisableB environment variables being set (see POD mode `disable` command, Table 68 on page 178) will no longer be disabled after a reset. This is useful for getting out of the situation in which all CPUs in the system have accidentally been disabled simultaneously.

Table 61. Virtual DIP Switch 10 - Override System Partitioning

10	Description
OFF	Not used.
ON	Not used.

Table 62. Virtual DIP Switch 11 - Use Default Console

11	Description
OFF	The default system console is defined by the ConsolePath environment variable in the IO6PROM.
ON	If no user-defined console can be located by means of the ConsolePath environment variable in the BaseIO PROM, and Switch 11 is set to ON, then the first serial device found in each module will be treated as a console device. The module that contains the Global Master CPU will become the overall system console.

Table 63. Virtual DIP Switch 12 - Router Oven Mode

12	Description
OFF	Does not allow the system to boot partway even though some router(s) may have more than one Xpress link.
ON	Switch 12 is a special function used by manufacturing that allows systems to boot partway even though some router(s) may have more than one Xpress link. This is normally an invalid configuration, but it is used by Silicon Graphics to run router tests on all router ports in sparsely populated test fixtures.

Table 64. Virtual DIP Switch 13 - Show Error State

13	Description
OFF	Disables Hub chip error state dump.
ON	Switch 13 causes the complete Hub chip error state to be dumped at system boot time. The state is still not dumped if the system was just powered on because the error state at power-on is random. Sometimes, if a system crashes the Hub error state after reset, it is useful to developers. It is not displayed by default because there are often extraneous errors that would alarm users.

Table 65. Virtual DIP Switch 14 - Ignore Autoboot

14	Description
OFF	The IO6PROM will not ignore the autoboot environment variable.
ON	If Switch 14 is set, the BaseIO PROM will ignore the autoboot environment variable and go to the 5-item PROM menu. This is an alternative to pressing ESC on the console when the system displays "Starting up the system."

Table 66. Virtual DIP Switches 16 and 17

15	16	Description
OFF	OFF	Reserved for future use.
OFF	ON	Reserved for future use.
ON	OFF	Reserved for future use.
ON	ON	Reserved for future use.

POD Mode Commands

Power-on diagnostic (POD) mode is a command interpreter present in the PROM. The main reasons for using POD mode are:

- Nullify the MSC password
- Nullify the monitor password
- Flash (upgrade) the microcode of an IP27 board
- Generate a hardware error dump and run the FRU analyzer program
- Test local memory
- Use a memory error report to identify a failing DIMM pair

This section contains the following topics:

POD Command Interpreter Numbering

The POD command interpreter uses the following constants:

- Hexadecimal numbers are prefixed with 0x
- Octal numbers are prefixed with 0
- Binary numbers are prefixed with 0b
- All other numbers are decimal
- Module numbering and node numbering start with 0

Use the following suffixes to enter large constants:

g	Gigabytes
m	Megabytes
k	Kilobytes

POD Modes

POD can run in three modes:

- Dirty exclusive (Dex)
- Cached (Cac)
- Uncached (Unc)

Dex Mode

In Dex mode, POD requires very few system resources to run. It does not require memory; instead, it accesses its program text directly from the PROM and uses the microprocessor's data cache as memory for its stack. The secondary cache is not used. NMI and uncaught exceptions typically result in a Dex mode POD prompt.

Dex mode is generally very slow because PROM instruction fetches are very slow. Do not perform long memory tests or flash remote PROMs from Dex mode. Certain commands do not execute in Dex mode, especially the commands that attempt to program the PROM. In addition, when you load data, store data, or run memory tests, ensure that you do not access cached memory addresses while the POD interpreter is in Dex mode.

Cac Mode

In Cac mode, POD places its program text, data, and stack in cached memory. This may occur only after memory has been probed and configured. POD runs out of cached memory very quickly. When the PROM takes an exception or an NMI and enters POD mode, the PROM enters Dex mode. You may be able to reenter Cac mode by using the `go cac` command.

Unc Mode

In Unc mode, POD places its program text, data, and stack in uncached memory. This is similar to Cac mode, except the POD does not use the cache and it runs slower.

You can force the PROM to enter the POD mode at different stages of the boot process by setting the Debug switches. The POD mode prompt uses a 6-field format (refer to page 176).

POD Prompt

The POD prompt contains the following fields:

- The node slot number and CPU that you are communicating with
- The NASID (NUMA Address Space ID)
- The port that you are communicating with
- The current POD mode

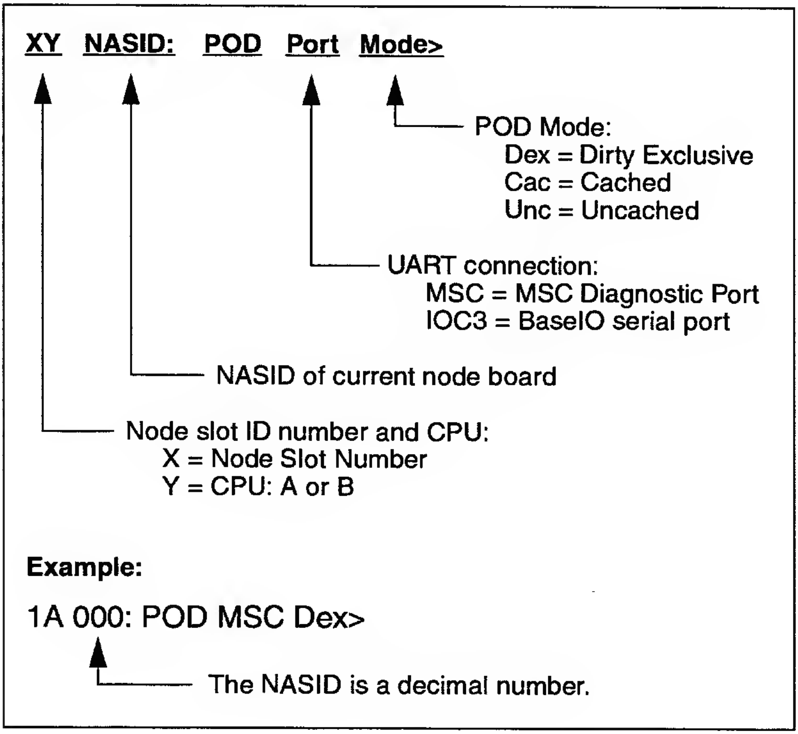


Table 67 lists the editing characters at the POD prompt.

Table 67. POD Mode Editing Characters

Character	Function
Ctrl+C	Abort the last POD mode command (works for many POD mode commands, including running the memory tests, flashing the PROM, and executing the loop and repeat commands).
Ctrl+H	Backspace one character.
Backspace	Backspace one character.
Delete	Backspace one character.
Ctrl+R	Redisplay the line.
Ctrl+U	Erase the line.
!!	Use the contents of previous command line.
!$	Use the last word from previous command line.
Ctrl+x Ctrl+y	Use the contents of previous command but substitute first occurrence of string x with string y.

Environment variables set with the setenv command are substituted wherever the name of an environment variable appears in backward single quotation marks (`and`). For example:

```
Set "range" to? n:1 u: 0x3ff0000 2m
POD Dex Hub 000 1A> dirtest `range`
POD Dex Hub 000 1A> dirinit `range`
POD Dex Hub 000 1A> memtest `range`
```

For commands that take other commands as arguments (for example, loop), you can enter a single command or you can enclose multiple commands in brackets { }. The following examples show both methods:

```
POD Dex Hub 000 1A> loop pi_rt_count
POD Dex Hub 000 1A> loop {ld pi_rt_count; sd pi_rt_count 0}
```

Table 68 describes the POD mode commands. The `help` or `?` command prints a command summary. When you supply a command-name argument with the `help` or `?` command, the `help` or `?` command prints the summary of the specified command.

Table 68. POD Mode Command Reference

Command/Syntax	Description
`?` [*CMDNAME*]	Display help information.
`adline` *LINE*	Dump the specified data cache line.
`adtag` *LINE*	Dump the specified data cache tag.
`ailine` *LINE*	Dump the specified instruction cache line.
`aitag` *LINE*	Dump the specified instruction cache tag.
`altregs` *LINE*	Use the alternate registers.
`asline` *LINE*	Dump the specified secondary cache line.
`astag` *LINE*	Dump the specified secondary cache tag.
`bist` | `ae` | `lr` | `ar` [`n:`*NASID*]	Run the Hub self-test.
`call` *ADDR* [A0 [A1 [A2 [A3]]]]	Call a subroutine.
`chklink`	Run the local link connection test.
`clear`	Clear memory errors.
`clearlog` [`n:`*NASID*]	Clear specified log file.
`clearalllogs`	Clear all log files.
`cpu` [[`n:`*NASID*] `a`|`b`]	Switch to CPU where *NASID* is always 0 and CPU is a or b.
`crb` [`n:`*NASID*]	Dump I/O interface (II) CRBs.
`crbx` [`n:`*NASID*]	Dump the CRBs in a 137-column-wide format.
`dbg` [*VIRT_VAL PHYS_VAL*]	Display or change the virtual and physical debug DIP switch settings.
`delay` *MICROSEC*	Delay command execution.
`dgbrdg` [`mn` | `mh` | `mm`[`n`*NASID*] [`s`*SLOT*]]	Run the Bridge test.

Table 68. POD Mode Command Reference (continued)

Command/Syntax	Description
`dgconf [mn	mh	mm] [n`*NASID*`] [s`*SLOT*`]`	Run the PCICONFIG test.
`dgpci [mn	mh	mm] [n`*NASID*`] [s`*SLOT*`] [p`*PCINUM*`]`	Run the PCI bus test.
`dgsdma [mn	mh	mm	mx] [n`*NASID*`] [s`*SLOT*`] [p`*PCINUM*`]`	Run the serial DMA test.
`dgspio [mn	mh	mm	mx] [n`*NASID*`] [s`*SLOT*`] [p`*PCINUM*`]`	Run the serial programmed I/O test.
`dgxbow [mn	mh	mm] [n`*NASID*`]`	Run the XBOW test.
`dips`	Read and display the debug DIP switch settings.
`dirinit n:`*NASID* `b:`*BANK* `u:`*M R*	Initialize a range of memory to the "Unowned" state. Where *NASID*=NASID, *BANK*=BANK, *M*=starting address, and *R*=number of Megabytes to test.
`dirstate [`*STATE* `[`*BASE* `[`*LEN*`]]]`	Scan a range of addresses for the directory state that you specify.
`dirtest n:`*NASID* `b:`*BANK* `u:`*M R*	Perform directory test and initialization. Where *NASID*=NASID, *BANK*=BANK, *M*=starting address, and *R*=number of Megabytes to test.
`dis` *ADDR* `[`*COUNT*`]`	Disassemble the data stored in the specified memory locations.

`disable n:`*NASID* `[`*CPU A or B*`]`	Disable the specified CPU.
	For example:
	`disable n:1 "cpua"`
	`disable n:2 "mem1"`
	Note that POD mode syntax differs from the PROM monitor command syntax.

`disable n:`*NASID memx*	Where *x*=memory bank
`disc`	Discover the CrayLink interconnect.

Table 68. POD Mode Command Reference (continued)

Command/Syntax	Description
`dline` *LINE*	Dump the specified data cache line.
`dtag` *LINE*	Dump the specified data cache tag.
`dumpspool [n:`*NASID CPU A or B*`]`	Dump the PI error spool.
`ecc [on | off]`	Enable or disable ECC mode.
`echo` *"STRING"*	Echo a string on the console.
`edump_bri [n:`*NASID*`]`	Dump Bridge error information to memory.

`enable n:`*NASID* `[`*CPU A or B*`]`	Enable the specified CPU.
	For example:
	`enable n:1 "cpua"`
	`enable n:2 "mem1"`
	Note that POD mode syntax differs from the PROM monitor command syntax.

`enable n:`*NASID memx*	Where *x*=memory bank
`error`	Display errors.
`error_dump`	Dump the error information.
`exec [`*SEGNAME* `[`*FLAG*`]]`	Load and execute a segment.
`flash` *NASID* `[...]`	Program remote PROM.
`flush`	Flush and invalidate cache(s)
`for (`*CMD;EXPR;CMD*`)` *CMD*	Repeat the specified command(s) for the number of iterations determined by the specified expression.
`fru [1|2]`	Run FRU analyzer program.
`go dex|unc|cac`	Set memory mode.
`help [`*CMDNAME*`]`	Display help.
`hubsde`	Run the Hub send data error test.
`if (`*EXPR*`)` *CMD*	Execute the specified command(s) depending on the value of a conditional expression.
`iline` *LINE*	Dump the specified instruction cache line.
`im [`*BYTE*`]`	Set the processor interrupt mask.

Table 68. POD Mode Command Reference (continued)

Command/Syntax	Description
`initalllogs`	Initialize all PROM logs for all modules in the system; resets PROM variables to factory defaults.
`initlog [n:`*NASID*`]`	Initialize the PROM log.
`inval [i] [d] [s]`	Invalidate cache(s).
`ioc3`	Use the IOC3 UART.
`itag` *LINE*	Dump the specified instruction cache tag.
`jump` *ADDR* `[A0 [A1]]`	Invalidate cache and continue execution at a specific address.
`junk`	Use the junk bus UART.
`kdebug [`*STACKADDR*`]`	Debug the kernel.
`kern_sym`	Use the kernel symbol table.
`la` *ADDR* `[`*COUNT*`]`	Load byte data and display it as ASCII characters.
`lb` *ADDR* `[`*COUNT*`]`	Load a byte.
`ld` *ADDR* `[`*COUNT*`]`	Load a double word.
`lh` *ADDR* `[`*COUNT*`]`	Load a halfword.
`lw` *ADDR* `[`*COUNT*`]`	Load a word.
`log [n:`*NASID*`] [`*SKIP*`[`*COUNT*`]]`	Display entries from the PROM log for the specified node.
`loop` *CMD*	Repeat the specified command(s) forever.
`lw` *ADDR* `[`*COUNT*`]`	Load a word.
`maxerr` *COUNT*	Test the error limit.
`memcmp` *DST SRC LEN*	Compare data (by bytes) in one memory location to data in another memory location.
`memcpy` *DST SRC LEN*	Copy memory (by bytes) from one memory location to another memory location.

Table 68. POD Mode Command Reference (continued)

Command/Syntax	Description
`meminit n:`*NASID* `b:`*BANK* `u:`*M R*	Clear memory. Where *NASID*=NASID, *BANK*=BANK, *M*=starting address, and *R*=number of Megabytes to test.
`memset` *DST BYTE LEN*	Fill memory with a byte pattern.
`memsum` *SRC LEN*	Add data (by bytes).
`memtest n:`*NASID* `b:`*BANK* `u:`*M R*	Perform a memory sanity test. Where *NASID*=NASID, *BANK*=BANK, *M*=starting address, and *R*=number of Megabytes to test.
`modnic`	Display the NIC value of the current module.
`module [`*VAL*`]`	Display or change a module number.
`msc`	Use the system controller UART.
`nic [n:`*NASID*`]`	Display the Hub NIC.
`nm` *ADDR*	Look up a PROM address.
`nmi n:`*NASID* `[a | b]`	Send NMI to node.
`node [[`*VEC*`]` *ID*`]`	Get or set node ID.
`pa` *ADDR* `[`*BITNO*`]`	Print an address.
`pas ["`*STRING*`"]`	Display or change the MSC password.
`pb` *EXPR*	Print the value of an expression in binary.
`pcfg [n:`*NASID*`] [v]`	Dump pcfg structure.
`pd` *EXPR*	Print the value of an expression in decimal.
`pf [`*REGNO*`]`	Print the contents of a processor's COP1 floating-point register.
`po` *EXPR*	Print the value of an expression in octal.
`pr [`*GPRNAME* `[`*VAL* `| ~]]`	Print the contents of a register.
`printenv [n:`*NASID*`] [`*VAR*`]`	Print the value of the specified environment variable(s).

Table 68. POD Mode Command Reference (continued)

Command/Syntax	Description
`px` *EXPR*	Print the value of an expression in hexadecimal.
`qual [on | off]`	Enable or disable quality mode.
`rbist le | ae | lr | ar` *VEC*	Run the router self-test.
`reconf`	Initialize the memory/directory section (MD) of the Hub.
`repeat` *COUNT CMD*	Repeat count.
`reset`	Reset the system.
`rlog [n:`*NASID*`] [`*SKIP*`[`*COUNT*`]]`	Display entries from the PROM log of the specified Node from the most recent entry to least recent entry.
`rnic [`*VEC*`]`	Read the router NIC.
`route [`*VEC NASID*`]`	Set up a route.
`rtrsde`	Run the router send data error test.
`santest n:`*NASID* `b:`*BANK* `u:`*M R*	Run the sanity test. Where *NASID*=NASID, *BANK*=BANK, *M*=starting address, and *R*=number of Megabytes to test.
`sb` *ADDR* `[`*VAL*`[`*COUNT*`]]`	Store a byte.
`sc [`*"COMMAND"*`]`	Send a command to the MSC.
`scandir` *ADDR* `[`*LEN*`]`	Scan directory states.
`scr` *ADDR* `[`*COUNT*`]`	Read from the system controller NVRAM.
`scw` *ADDR* `[`*VAL* `[`*COUNT*`]]`	Write to the system controller NVRAM.
`sd` *ADDR* `[`*VAL*`[`*COUNT*`]]`	Store a double word.
`sdv` *ADDR COUNT*	Store a double word and verify it.
`segs [`*FLAG*`]`	List the segments.
`setenv [n:`*NASID*`]` *VAR* `[`*"STRING"*`]`	Set a variable.
`sf` *REGNO VAL*	Store the contents of a processor's COP1 floating-point register.
`sh` *ADDR* `[`*VAL*`[`*COUNT*`]]`	Store a halfword.
`slave`	Enter slave mode.

Table 68. POD Mode Command Reference (continued)

Command/Syntax	Description
`sleep` *SEC*	Sleep for a specified number of seconds.
`sline` *LINE*	Dump the specified secondary cache line.
`softreset n:`*NASID*	Soft reset a node.
`sr` *GPRNAME VAL*	Store the contents of a register.
`stag` *LINE*	Dump the specified secondary cache tag.
`sw` *ADDR* [*VAL*[*COUNT*]]	Store a word.
`talk [n:`*NASID* `a | b]`	Use the net UART.
`tdisable n:`*NASID* [*CPU A or B*]	Temporarily disable the specified CPU.
`time` *CMD*	Perform benchmark timing.
`tlbc` [*INDEX*]	Clear the TLB.
`tlbr` [*INDEX*]	Read the TLB.
`traplog`	Trap the log.
`unsetenv [n:`*NASID*`]` *VAR*	Remove a variable.
`verbose [0 | 1]`	Specify whether or not the power-on diagnostics run in verbose mode.
`version`	Display the PROM version
`vr` *VEC VADDR*	Perform a vector read operation.
`vw` *VEC VADDR VAL*	Perform a vector write operation.
`vx` *VEC VADDR VAL*	Perform a vector exchange operation.
`while` (*EXPR*) *CMD*	Repeat the specified command(s) while the specified expression evaluates to true.
`why`	Display the current exception and NMI status.

Running Power-on Diagnostics Manually

You can run several power-on diagnostics manually in an interactive mode in addition to invoking them automatically during the boot sequence.

Refer to the *Origin2000 and Onyx2 Power-On Diagnostics* manual, document number 108-0161-002, for detailed descriptions of each diagnostic test.

Power-on Diagnostics that Run from the POD Prompt

Table 69 lists and describes the power-on diagnostics that you can run from the POD prompt and indicates the command(s) to use to start the tests. Refer to Table 68 on page 178 for POD prompt command syntax.

Power-on Diagnostics that Run PROM Monitor Prompt

The power-on diagnostics that reside in the BaseIO PROM do not run from the `POD>` prompt; you must run these diagnostics from the PROM monitor prompt, which is `>>`. You can access the PROM monitor prompt by selecting `5` from the System Maintenance Menu.

Table 70 lists the power-on diagnostics that run from the PROM monitor command prompt. Refer to Table 44 on page 138 for the PROM monitor prompt command syntax.

Table 69. Power-on Diagnostics that Run from the POD Prompt

Test	Description
`chklink`	The local link connection test checks the connection between the local Hub and an attached Hub or Router. You should verify that the processor and the Hub are functional before you run this test. Example: `POD Dex 000 0A>` **`chklink`** If this test fails, a `RSLT chk_local_link FAIL` message is displayed on the console connected to the ACP. The failing FRU is the midplane (most likely), node board, or Router (least likely). The failure is caused by a failing local Hub, a failing attached Hub or Router, a failing local Hub-to-attached Hub interconnect, or a failing local Hub-to-attached Router interconnect.
`hubsde`	The Hub send data error test verifies that the Hub or Router attached to the local Hub can detect link errors. You should ensure that the processors and the Hub are functional before you run this test. Example: `POD Dex 000 0A>` **`hubsde`** If this test fails, a `RSLT hub_send_data_error FAIL` message is displayed on the console connected to the ACP. The failing FRU is the Router, node board, or midplane. A failure in this test indicates a bad local Hub, a bad attached Hub or Router, a bad local Hub-to-attached Hub interconnect, or a bad Hub-to-attached Router interconnect.
`rtrsde`	The Router send data error test verifies that the local Hub can detect link errors. You should ensure that the processors and the Hub are functional before you run this test. Example: `POD Dex 000 0A>` **`rtrsde`** If this test fails, a `RSLT rtr_send_data_error FAIL` message is displayed on the console connected to the ACP. The failing FRU is the node board (most likely), Router, or midplane/cable (least likely). A failure in this test indicates a bad local Hub, a bad attached Hub or Router, or a bad local Hub-to-attached Hub or Router interconnect. Because of the testing that has been performed by the time this test runs in the boot process, the most likely cause of failure is a failing attached link or a failing attached board or chip.

Table 69. Power-on Diagnostics that Run from the POD Prompt (continued)

Test	Description

`dgxbow`	The XBOW test checks for valid reset values in the XBOW registers and tests the XBOW interrupt hardware. In the following example, the `dgxbow` test runs in heavy mode on Node 1.
	
	Example: `POD Dex 000 0A>` **`dgxbow mh n1`**
	
	The XBOW test displays the following failure messages on the ACP:
	
	This message indicates the XBOW test failed.
	`xbow_sanity: Bad xbow id reg value` ***value***
	
	This message indicates the XBOW test failed.
	`xbow_sanity: Bad Xbow id Exp:` ***expected*** `Recv:` ***received***
	
	This message indicates a failing XBOW or XBOW-to-Hub connection.
	`xbow_sanity: Reg write Read miscompare.`
	`Exp:` ***expected***`:` `Recv:` ***received***
	
	This message indicates a failing XBOW chip.
	`xbow_sanity: Register access violation not detected.`
	
	This message indicates the XBOW test failed.
	`xbow_sanity: Status Register Clear on Read not working!!`
	
	This message indicates a failing XBOW, a failing XBOW-to-Hub connection, or a failing Hub.
	`xbow_sanity: Interrupt did not reach PI_INT_PEND0.exp:` ***expected*** `recv:` ***received***
	
	This message follows any of the previous XBOW failure messages. The failing FRU is either the midplane or a node board when the XBOW test fails.
	`RSLT xbow_sanity FAIL`

Table 69. Power-on Diagnostics that Run from the POD Prompt (continued)

Test	Description

`dgbrdg`	The Bridge test checks for valid reset values in the Bridge registers and tests the Bridge interrupt hardware. In the following example, the `dgbrdg` command runs in heavy mode on the Bridge ASIC on the I/O board in slot IO4 of the module that contains the Node from which you entered the command.
	
	Example: `POD Dex 000 0A>` **`dgbrdg mh s4`**
	
	Any of the following messages indicate a failing BaseIO board:
	`bridge_sanity: Bad bridge id register value (LSB != 1)`
	`bridge_sanity: Bad bridge id Exp:` ***expected*** `Recv:` ***received***
	`bridge_sanity: PCI bit set to 0 sn status reg`
	`bridge_sanity: Timeout val reg exp:` ***expected*** `recv:` ***received***
	`bridge_sanity: Bridge control reg exp:` ***expected*** `recv:` ***received***
	`bridge_sanity: Bad reset val in dev reg`
	`exp:` ***expected*** `recv:` ***received***
	`bridge_sanity: Interrupt status register value non-zero: value`
	`bridge_sanity: Bridge(wr): Exp:` ***expected*** `Recv:` ***received***
	`bridge_sanity: Interrupt status reg bit not set:`
	`exp:` ***expected*** `recv:` ***received***
	`bridge_sanity: Interrupt interrupt did not reach PI_INT_PEND0:`
	`exp:` ***expected*** `recv:` ***received***
	
	The following message indicates a failing Bridge, a failing XBOW chip, or a failing Hub:
	`bridge_sanity: Interrupt interrupt did not occur`
	
	The following message is displayed after any of the previous bridge test failure messages:
	`RSLT bridge_sanity FAIL`

`dgconf`	The PCICONFIG test (`dgconf`) verifies the PCI configuration space on the BaseIO board. In the following example, the `dgconf` command runs the PCICONFIG test in heavy mode on the BaseIO board in slot IO7 of the module that contains the Node from which you entered the command.
	
	Example: `POD Dex 000 0A>` **`dgconf mh s7`**
	
	The following messages indicate that the PCICONFIG test failed and the failing FRU is the BaseIO board:
	`io6config_space: IOC3 PCI_SCR reg reset val`
	`exp:` ***expected*** `recv:` ***received***
	`io6config_space: IOC3 PCI_ADDR reg reset val`
	`exp:` ***expected*** `recv:` ***received***
	`io6config_space: IOC3 PCI_ADDR reg write read error.`
	`wrote:` ***value*** `read:` ***value***
	`io6config_space: PCI` ***device*** `device QL command status reg`
	`resetval exp:` ***expected*** `recv:` ***received***
	`io6config_space: PCI device` ***device*** `QL class code and revision`
	`read as` ***value***
	`io6config_space: PCI device` ***device*** `QL base address reg reset`
	`val exp:` ***expected*** `recv:` ***received***
	`io6config_space: PCI device` ***device*** `QL base addr write read`
	`error. wrote:` ***value*** `read:` ***value***
	`RSLT io6config_space FAIL`

Table 69. Power-on Diagnostics that Run from the POD Prompt (continued)

Test	Description
`dgpci`	The PCI bus test (`dgpci`) checks the PCI bus with walking data tests. In the following example, the `dgpci` command runs the PCI bus test in normal mode on the Node from which you entered the command. The following example uses the IOC3 ASIC on the BaseIO board in slot number IO3 of the module that contains the Node from which you started the test. Example: `POD Dex 000 0A>` **`dgpci mn s3`** The following messages indicate a failing BaseIO board: `pcibus_sanity: IOC3 enet hashed low addr reg reset val exp:` ***expected*** `recv:` ***received*** `pcibus_sanity: PCI walk one test. exp:` ***expected*** `recv:` ***received*** `pcibus_sanity: PCI walk zero test. exp:` ***expected*** `recv:` ***received*** `RSLT pcibus_sanity FAIL`

`dgspio`	The serial programmed I/O (PIO) test (`dgspio`) uses the internal loopback features of the SuperIO chip to test the serial PIO functionality of the BaseIO board. The following example runs the serial PIO test in heavy mode on the Node from which you entered the command.
	Example: POD Dex 000 0A> **`dgspio mh`**
	The following messages indicate a failing BaseIO board.
	`serial_pio: UARTA read timed out on LSR<Data Ready>`
	`serial_pio: UARTB read timed out on LSR<Data Ready>`
	`serial_pio: UARTB error. Chars in FIFO after reset .....`
	`serial_pio: UARTA error. Chars in FIFO after reset .....`
	`serial_pio: UARTB error. Chars in FIFO after reset .....`
	`serial_pio failed with` ***number*** `errors`
	`RSLT serial_pio FAIL`
	The following message indicates a failing IOC3-to-SuperIO interface, a failing SuperIO chip, or a failing IOC3 chip. This message also appears if you do not install external loopback plugs on the ports before you run the test in external loopback mode.
	`serial_pio: UARTA sent:` ***value*** `recv:` ***value***
	`serial_pio: UARTB sent:` ***value*** `recv:` ***value***

Table 69. Power-on Diagnostics that Run from the POD Prompt (continued)

Test	Description

dgsdma	The serial DMA test (dgsdma) verifies that serial DMA can be performed. The following example runs the serial DMA test in heavy mode on the Node from which you entered the command. The test uses the IOC3 chip on the BaseIO board in slot IO1 of the module that contains the Node from which you started the test.
	Example: POD Dex 000 0A> **dgdma mh**
	The following messages indicate a failing BaseIO board.
	`serial_dma: PCI: could not get the ioc3 base addr`
	`serial_dma: Timeout on uartA dma ....`
	`serial_dma: Timeout on uartB dma ....`
	`RSLT serial_dma FAIL`
	The following message indicates that the data that was received is different from the data that was sent. This message indicates a failing SuperIO chip, a failing IOC3 chip, a failing Bridge, or a failing Hub. This message also appears if you do not install external loopback plugs on the ports before you run the test in external loopback mode.
	`serial_dma: Serial DMA exp:` *expected* `recv:` *received*

diag_scsi -t ram	The SCSI SSRAM test verifies that data can be loaded into SCSI SSRAM memory and read back successfully. It also verifies the "mailbox" mechanism before it performs this test. The following example runs the SCSI SSRAM test: >> **diag_scsi -t ram** [*options*] The SCSI SSRAM test returns the following failure messages. The failing FRU is the BaseIO board. `scsi_ram: SCSI mailbox error. erp:` *expected* `recv:` *received* `scsi_ram: QL: wrote:` *write_value* `read` *read_value* `at addr:` *address* `RSLT scsi_dma FAIL`
diag_scsi	Comprehensive SCSI test.

diag_scsi -t dma	Prerequisites: The CPU from which you run this test must be in Cac mode. Enter `go cac` before you run the test.
	The SCSI DMA test verifies that the SCSI DMA mechanism is functional. The following example runs the SCSI DMA test:
	>> **diag_scsi -t dma** [*options*]
	The failing FRU is the BaseIO board when the SCSI DMA test returns the following failure messages:
	`scsi_dma: LOAD SSRAM command failed`
	`scsi_dma: DUMP SSRAM command failed`
	`RSLT scsi_dma FAIL`
	The following message indicates a failing SCSI controller, Bridge chip, or Hub chip:
	`scsi_dma: DMA failure exp` *expected* `recv` *received* `at` *location*

Table 70. Power-on Diagnostics that Run from the PROM Monitor Prompt

Command	Description

`diag_scsi -t controller`	The SCSI self-test uses the SCSI self-test firmware to test the SCSI controller. The following example tests the SCSI controller:
	`>> ` **`diag_scsi -t controller`** `[`*options*`]`
	The failing FRU is the BaseIO board when the SCSI self-test returns the following failure messages:
	`scsi_controller: EXECUTE FIRMWARE Command Failed.`
	`scsi_controller: write data failure at` ***count*** `word`
	`scsi_controller: Bad status val ==>` ***value*** `..`
	`RSLT scsi_controller FAIL`

`diag_enet`	Prerequisites: Ensure that the following hardware is functional before you run this test:
	• Non-Ethernet-specific logic on the IOC3 ASIC
	• PCI bus address/data
	• Bridge ASIC
	• XTALK interface from Bridge ASIC to XBOW chip
	• XBOW chip, midplane, and all IP27 board components
	The comprehensive Ethernet test runs all of the Ethernet tests in an order that optimizes fault isolation. The following error messages indicate a failing BaseIO board:
	`enet_all: SSRAM test failed.`
	`enet_all: Ethernet PHY chip register test failed.`
	`enet_all: Ethernet TX clock test failed.`
	`enet_all: Ethernet NIC test failed.`
	`enet_all: IOC3 internal loopback DMA test failed.`
	`enet_all: Ethernet PHY chip internal loopback DMA test failed.`
	`enet_all: Ethernet PHY chip internal loopback DMA test failed.`
	`enet_all: Ethernet Twister chip internal loopback DMA test failed.`
	`enet_all: Ethernet external loopback DMA test failed.`
	`enet_all: Xtalk stress test failed.`

`diag_enet -t ssram`	The Ethernet SSRAM test checks the 128 KB Ethernet packet buffering SSRAM, the SSRAM's interface to the IOC3 ASIC, and the parity generation and checking logic in the IOC3. The following messages indicate a failing BaseIO board: `enet_ssram: ssram address walk error. Ex:` ***expected_value*** `Recv:` ***received_value*** `enet_ssram: ssram parity test error. Exp:` ***expected_value*** `Recv:` ***received_value*** `enet_ssram: **Data miscompare(A)** offset: offset_value exp:` ***expected_value*** `recv:` ***received_value***

Table 70. Power-on Diagnostics that Run from the PROM Monitor Prompt (continued)

Command	Description

`diag_enet -t phyreg`	The PHY register test checks the reset values of registers on the PHY chip.
	The following error messages indicate a failing BaseIO board:
	`enet_phy_reg: Read of PHY register 0 failed (returned 0xffff)`
	`enet_phy_reg: Timed out waiting for PHY to rest.`
	`enet_phy_reg: Timed out on MICR_BUSY before read.`
	`enet_phy_reg: Timed out on MICR_BUSY during read.`
	`enet_phy_reg: Timed out on MICR_BUSY before write.`
	`enet_phy_reg: Timed out on MICR_BUSY during write.`
	`enet_phy_reg: Reg 0 value wrong, exp:` ***expected*** `mask:` ***mask_value*** `got:` ***received***
	`enet_phy_reg: Reg 23 value wrong, exp:` ***expected*** `mask:` ***mask_value*** `got:` ***received***
	`enet_phy_reg: Reg 24 value wrong, exp:` ***expected*** `mask:` ***mask_value*** `got:` ***received***
	The following error message indicates that the test failed while checking a reset value in one of the registers that might change in future revisions of the PHY chip. This most likely indicates that the BaseIO board includes a newer version of the PHY chip and the diagnostic software has not been updated with this information. It could also indicate a failing PHY chip.
	`enet_phy_reg: Warning - PHY Identifier register #1 value` ***value*** `is not supported by this diag!`
	`enet_phy_reg: Warning - PHY Identifier register #2 value` ***value*** `is not supported by this diag!`
	`enet_phy_reg: Warning - PHY revision register value` ***value*** `is not supported by this diag!`
	The test failed while checking a reset value in one of the registers that might change in future revisions of the PHY chip. This most likely indicates that the BaseIO board includes a newer version of the PHY chip and the diagnostic software has not been updated with this information. It could also indicate a failing PHY chip.
	`enet_phy_reg: Warning - unexpected Reg` ***register*** `reset value, exp:` ***expected*** `mask:` ***mask_value*** `got:` ***received***

`diag_enet -t txclk`	The TX clock test verifies the presence of the TX clock sent from the PHY chip to the IOC3 ASIC. This test reads the IOC3 ETCSR register and checks the NO_TX_CLK bit. The following error message indicates a failing BaseIO board: `enet_tx_clk: IOC3 is not seeing the PHY TX clock.`

Table 70. Power-on Diagnostics that Run from the PROM Monitor Prompt (continued)

Command	Description

`diag_enet -t nic`	The NIC test checks the number in a can (NIC) device (a Dallas Semiconductor chip) on the BaseIO board; the NIC device stores the Ethernet MAC address. This test reads the NIC and then performs further testing if the results of the read are incorrect.
	
	The following error messages indicate a failing BaseIO board:
	
	`nic_eaddr: No presence pulse from NIC`
	`nic_eaddr: (check for an empty NIC socket)`
	`nic_eaddr: Invalid enet MAC command CRC byte` ***value***`, s/b 0x8d`
	`nic_eaddr: Invalid enet crc16` ***value***`, s/b 0xb001`
	`nic_eaddr: Invalid enet length byte` ***value*** `found in NIC, s/b 0x0a`
	`nic_eaddr: Invalid enet MAC crc16` ***value***`, s/b 0xb001`
	`nic_eaddr: Failed to find readable MAC NIC, rc` ***value***

`diag_enet -t ioc3_loop`	The IOC3 internal loopback test verifies Ethernet DMA by using the internal loopback features of the IOC3 ASIC. The following error messages indicate a failing BaseIO board:
	
	`enet_ioc3_loop: Timed out on EMCR_ARB_DIAG_IDLE trying to reset enet.`
	
	`enet_ioc3_loop: Unexpected EISR condition while waiting for TX_EMPTY (`***string1***`:` ***string2***`).`
	`enet_ioc3_loop: EISR value =` ***e_value***
	`enet_ioc3_loop: Elapsed time =` ***time*** `us`
	
	`enet_ioc3_loop: Bridge ISR bit value did not set after TX_EMPTY interrupt (`***string***`:` ***string***`). enet_ioc3_loop: ==> Bridge ISR =` ***value***
	
	`enet_ioc3_loop: Timed out polling RX buffer valid bit. (`***string***`:` ***string***`).`
	`enet_ioc3_loop:` ***value*** `of` ***value*** `packets were received.`
	
	`enet_ioc3_loop: RX data miscompare on packet` ***value*** `status word.`
	`enet_ioc3_loop: ==> exp: expected recv:` ***received*** `(`***string***`:` ***string***`)`
	
	`enet_ioc3_loop: RX data miscompare on packet` ***value*** `first word.`
	`enet_ioc3_loop: ==> exp:` ***expected*** `recv:` ***received*** `(`***string***`:` ***string***`)`

Table 70. Power-on Diagnostics that Run from the PROM Monitor Prompt (continued)

Command	Description

`diag_enet -t phy_loop`	The PHY chip loopback test checks Ethernet DMA by using the internal loopback features of the PHY chip on the BaseIO board. The PHY chip loopback DMA runs at 10 MB/s and also at 100 MB/s.
	The following message indicates a failing BaseIO board.
	`enet_phy_loop: Unexpected EISR condition while waiting for TX_EMPTY (`***string: string***`).`
	`enet_phy_loop: EISR value = `***e_value***
	`enet_phy_loop: Elapsed time = `***value***` us`

`diag_enet -t tw_loop`	The Twister chip internal loopback test verifies Ethernet DMA by using the internal loopback features of the Twister chip. The Twister chip loopback DMA runs at 100 MB/s. The Twister chip internal loopback test returns the same messages that the PHY chip internal loopback test returns.

`diag_enet -t ext_loop`	The external loopback DMA test verifies Ethernet DMA by using an external loopback cable or connector. This test verifies the autonegotiation feature and verifies the DMA at 10 MB/s and 100 MB/s. If the test fails, the failing component could be the BaseIO board, RJ45 connector, or the loopback cable/connector. The following error messages indicate that the external loopback DMA test failed.
	`enet_ext_loop: Phy did not complete auto-negotiation with itself via external loopback cable.`
	`enet_ext_loop: PHY reg 6 LP_AN_ABLE bit not set after auto-negotiation.`
	`==> Reg 0 = value`
	`==> Reg 4 = value`
	`==> Reg 5 = value`
	`==> Reg 6 = value`
	`*** Are you sure there is an external loopback?***`
	`enet_ext_loop: PHY reg 5 mode support bits (9:5) do not match reg 4 mode support bits after autonegotiation. Reg 4: `***value***` Reg 5: `***value***

`diag_enet -t ext_loop_10`	The external loopback DMA (10 MB/s) test is a version of the external loopback DMA test; this version has the following differences: it does not test autonegotiation and it runs at only 10 MB/s. Returns the same error messages as: `diag_enet -t ext_loop`

Table 70. Power-on Diagnostics that Run from the PROM Monitor Prompt (continued)

Command	Description
`diag_enet -t ext_loop_100`	The external loopback DMA (100 MB/s) test is a version of the external loopback DMA test; the differences are that it does not test autonegotiation and it runs at only 100 MB/s. Returns the same error messages as: `diag_enet -t ext_loop`

`diag_enet -t xtalk`	The XTALK stress test is a 100 MB/s DMA test that uses a continuous stream of DMA for 1 to 5 seconds. This test does not check the received data, but it does check the IOC3 and Bridge interrupt registers for errors. This test uses a data pattern that is designed to stress SSO on the LLP links in the XTALK path. The failure messages are the same as the failure messages for the Twister chip internal loopback test and also include the following failure messages. The FRU is the BaseIO board.
	
	`eenet_xtalk_stress: RX CONSUME did not increment from Xtalk stress loop` ***value*** `to` ***value***`.`
	`enet_xtalk_stress: TX CONSUME did not increment from Xtalk stress loop` ***value*** `to` ***value***`.`
	
	`enet_xtalk_stress: Unexpected EISR condition in Xtalk stress loop` ***value***`.`
	`enet_xtalk_stress: EISR value =` ***e_value***
	
	`enet_xtalk_stress: Unexpected Bridge ISR condition in Xtalk stress loop` ***value***`.`
	`enet_xtalk_stress: Bridge ISR value =` ***value***

Procedures

This section contains procedures and examples for the following topics:

Power-on Boot Process

The following pages describe the steps in the power-on boot process. The node board PROM microcode performs the following operations.

Note: Steps 1 through 6 produce no MSC or console output. Only node LEDs are active during this phase of the boot process.

1. Initialize the CPUs.
   - Set up the status registers
   - Clear the register files
   - Activate the node board LEDs
2. Test the primary CPU caches.
   - Test the instruction cache
   - Test the data cache
3. Set up dirty exclusive (Dex) mode.
   - Use the CPU primary data cache as memory
   - Initialize the stack
   - Start the boot procedure
4. Disable CPU A and/or CPU B.
   - Disable the CPUs as indicated by PROM environment variables

     - DisableA/DisableB
   - CPU disabling can be overridden by **virtual debug switch 9**
   - CPUs are completely separated from the Hub ASIC at this point
5. Arbitrate the local master (one per node card).
   - Checks for the presence of another CPU on the node board

     - A timeout occurs if the other CPU does not respond
   - The other CPU is disabled if it does not respond
6. Read the debug DIP switch settings from the MSC.

7. Determine the initial console device and initialize it.

    - Boot information now appears on the initial console device

    Note: Normally, the initial console port is the MSC diagnostic port.

8. Record error information that may indicate the cause of a prior crash.

    - Save the state of Hub error registers in the cache
    - Clear the state of Hub error registers
    - Enable SYSAD error checking

9. Initialize I/O.

    - Initialize the IO section of the Hub ASIC (II)
    - Initialize the XBOW ASIC
    - Initialize the Bridge ASIC
    - Initialize the PCI bus (on BaseIO)

10. Determine whether a BaseIO is present, and switch the console output to the tty_1 port of the BaseIO.

11. Display the PROM boot banner on the console.

    - Display the IP27 PROM version and size
    - Display the chip revisions
    - Display the slot ID and CPUs present
    - Display the other miscellaneous information

12. Configure the Hub ASIC clock speed.

13. Configure local memory.

    - Initialize the DIMM control registers
    - Probe the amount and type (premium or standard) of memory
    - Program the Hub memory configuration registers

14. Display any debug switches that are set to nondefault values.

    - Switch values are displayed on the console output

15. Determine the node's serial number, and provide it to the other nodes.

16. Run the Hub ASIC self-test if Heavy or Manufacturing diagnostics are selected.

    - Run the BIST (built-in self test) procedure, which automatically reboots the system

    - On reboot, if the test failed, stop with a failure LED value

17. Perform basic memory tests.

    - Ensure that address 0 of each bank is accessible

    - Disable banks that are not accessible

    - If **bank 0** is bad, swap it with **bank 1** (refer to the **SwapBank** environmental variable)

18. Download the PROM to memory.

    - Perform a memory test on the lower portion of local memory by using a small portion of PROM data

    - If the test passes, download a copy of the IP27 PROM into tested memory

    - Verify the download **checksum** to ensure that the PROM data downloaded correctly

19. Transfer the program counter to uncached RAM.

    - Runs faster than running out of the PROM
    - Does not require the secondary cache

20. Switch crucial structures from the cache into uncached memory.

    - Performs hardware inventory

21. Test and invalidate the secondary cache.

    - PROM assumes that all data in cache is invalid

22. Transfer the program stack to uncached RAM.

    - Call **UALIAS** mode
    - Discard the Dex mode data in the primary cache

23. Transfer the program counter and stack to cached RAM.
    - PROM now runs at top speed
24. Initialize the first 32 MB of bank 0 memory for use by the PROM.
25. Initialize permanent low-memory system data structures.
    - These structures persist across the IRIX kernel:

      KLDIR - Table for accessing all other low-memory structures
      NMI - Nonmaskable interrupt vector area
      KLCONFIG - System topology and configuration information
26. Run serial port diagnostics on the local BaseIO.
27. Each local master discovers its CrayLink interconnect topology.
    - Perform a depth-first search using **vector routing** operations
    - Build the PROM configuration data structure (**promcfg**) in low cached memory
28. Verify that all PROMs are running the same firmware version.
    - A PROM mismatch generates a **Barrier Sync Warning** on the console
29. Select the global master CPU.
    - The protocol uses **vector routing** operations
    - One global master is selected per system
30. The global master configures the CrayLink interconnect.
    - Determines the node ID for each node (not yet assigned)
    - Programs CrayLink routing tables in all Hub and Router ASICs
    - Informs each node of its node ID
31. All nodes switch back to uncached memory (required for changing the node ID).
    - Flush caches
    - Closely synchronize all nodes
    - Stop bus activity with tight loops internal to the microprocessor.

32. Change NASIDs.

    - Update the data structures that assume the NASID of each node is 0
    - Reinitialize coherence directories for the first 32 MB of memory

33. Switch back to cached memory in the new node ID space.

34. Test and initialize the rest of local memory (greater than 32 MB).

35. Check partitioning information.

    - Assign a partition master CPU to each partition
    - Reassign NASIDs if necessary
    - Erect partition fences to isolate partition memory space

    Note: Partitioning is not an officially supported feature of IRIX 6.5.

36. Initialize any headless nodes (nodes without functional CPUs).

    - Illuminate all LEDs
    - Disable the CPUs
    - Read the NIC
    - Probe, configure, and initialize memory
    - Initialize low-memory data structures
    - Initialize I/O

37. If debug switch 13 is set, or if the system is booting from a reset command, display hardware error state information.

    - Hub error registers are decoded in detail

38. Transfer control to the BaseIO PROM (can be inhibited by debug switch setting).

    - Decompress the BaseIO PROM image into memory from the console BaseIO

    Note: A failure at this point will cause the system to drop into POD mode.

    - If there is no console BaseIO in the system, decompress the internal IP27 PROM copy instead

- Jump to the entry point of BaseIO PROM.

Note: At this point, the boot process continues from the BaseIO PROM microcode.

39. Display the BaseIO PROM banner.

40. Initialize the global data structures.

41. Number the nodes in the system.

    - Start with zero

42. Synchronize the node clocks.

    - Display midplane clock speed mismatches

43. Initialize and test:

    - SCSI buses
    - Console port
    - Ethernet port
    - Graphics pipe, Graphics console, and audio (if applicable)

44. Compare the current system inventory with the inventory that is stored in PROM.

45. Print a summary of system configuration and diagnostics.

46. Fetch the PROM environment variables.

47. Display the PROM monitor main menu.

Configuring the System Console

If you are connecting a Silicon Graphics workstation to an Origin 200 server, you need one of three RS-232 serial cables:

- DB-9 to DIN 8, null modem cable; use this cable to connect the Origin 200 server to a Silicon Graphics Indy, Indigo 2, Indigo, Personal IRIS 4D/30, or Personal IRIS 4D/35 workstation
- DB-9 to DB-9, null modem cable with older cable wiring; use this cable to connect the Origin 200 server to a Silicon Graphics Onyx, Personal IRIS, or Power Series workstation.
- DB-9 to DB-9, null modem cable with IBM-compatible wiring; use this cable to connect the Origin 200 server to an Onyx2 workstation.

Connect the workstation and the Origin 200 server together using a serial cable; then use terminal emulation software (such as `cu`) to log in to the Origin 200 server.

The UNIX UUCP subsystem contains the program `cu` and several configuration files necessary for this procedure. To make sure the UUCP subsystem is installed on the workstation, enter the following command:

```
versions | grep uucp
```

Verify that the following line is displayed:

```
I eoe.sw.uucp 02/20/96 UUCP Utilities
```

If you do not see the above line, the UUCP subsystem is not installed on the workstation. Install the `eoe.sw.uucp` subsystem on the workstation.

Note: The UUCP software is required only on the SGI workstation and not on the Origin 200 system.

Use the following procedure to configure the workstation:

1. Log in as root (or use `su` to invoke superuser privileges) on the workstation that you will connect to the Origin 200 server.

2. Use a text editor (such as `vi`, `jot`, or `emacs`) to edit the file `/etc/inittab`.

   Find the line that corresponds to the serial port that you wish to use on the workstation. For example, the following example corresponds to port 2:

   ```
   t2:23:off:/sbin/getty -N ttyd2 co_9600 # port 2
   ```

   Note: You can use any port you wish on the workstation, but remember that the Origin 200 workstation's alternate console port (ACP) is port 1.

   If the third field in the line is the word "`off`" (as in the example above), exit the file and continue with Step 3.

   If the third field is the word "`on`," change it to "`off`." Then save the file, exit, and at the shell prompt enter the command:

   `telinit q`

3. Using a text editor, open the `/etc/uucp/Devices` file. Locate the following line:

   ```
   # Direct ttyd2 - 9600 direct
   ```

   Remove the number sign (#) and subsequent space at the beginning of the line as shown below:

   ```
   Direct ttyd2 - 9600 direct
   ```

   This change allows you to use port 2 on the workstation to connect to the Origin 200 server. If you are using a different serial port, replace `ttyd2` in the line above with the name of the port that you are using, for example `ttyd3` or `ttyd4`.

4. Edit the `/etc/uucp/Systems` file. At the end of the file, add this line:

   `Direct Any Direct 9600 -`

5. Save the file and exit.

6. Test the configuration on the workstation. Enter the following command at a shell prompt:

   `cu Direct`

   You should see the following message:

   `Connected`

   This message means that the `cu` process has successfully opened the serial port.

7. Enter the following command to exit from cu and return to the shell prompt:

   `~.` (A tilde character, followed by a period.)

Note: If you do not see the message "`Connected`," you may have made an error in one of the previous steps. Repeat this procedure, starting from Step 2. In particular, make sure that you reference the same port in each of the files `/etc/inittab` and `/etc/uucp/Devices`.

8. Connect one end of the serial cable into port 1 of the Origin 200 server, and the other end of the cable (either DB-9 or DIN 8) into the appropriate port on the workstation; for example, port 2.

9. On the workstation, enter this command at a shell prompt:

   `cu Direct`

   You should see this message:

   `Connected`

10. Press `<Enter>` once or twice. You should see a login prompt from the Origin 200 server:

    `IRIS login:`

11. At the `IRIS login:` prompt, enter **`root`**, **`setup`**, **`guest`**, or another account name (if you have already created one) as appropriate, and press `<Enter>`. You see some brief messages and the following prompt:

    `TERM = (vt100)`

12. Enter the following terminal type and press `<Enter>`:

    `iris-ansi`

Flashing Firmware (PROMs)

This section contains procedures for flashing firmware (PROMs) and is organized according to the following topics:

Note: Refer to "Restoring the GE EEPROM" procedure on page 253 for information about restoring the GE EEPROM.

Flashing the MMSC

The firmware for the Origin 2000 MMSC resides in nonvolatile storage. The MMSC has six serial ports. The ports are not interchangeable; each port has a special function.

Three methods are available to flash the firmware in the MMSC; the best method depends on your circumstances.

- Method 1: Flash from the local IRIX operating system
  This is the easiest and fastest method to flash a single rack or a multirack system. However, MMSC firmware must be loaded, running, and partially functional. Method 1 can be performed automatically or manually; both are described in the next bulleted list.
- Method 2: Flash from an IRIX operating system that is running on an Indy workstation.
  Use this method if the firmware in the MMSC is not functional or if there is no firmware loaded in the MMSC. This method is more complicated and slower than flashing from the Origin system.
- Method 3: Flash from a PC
  Use this method for the same reasons as for Method 2 and when you do not have an Indy workstation to flash from.

The following MMSC, MSC, and IRIX revision-level restrictions affect the flash procedure and how you must set some of the port attributes. In addition, there are COM port speed restrictions.

- MMSC version 1.3 and later and IRIX 6.5 allow the user to flash a single rack or a multirack system with one command (`flashmmsc -a`) and no cable reconfiguration. This is **Method 1- automatic**.
- Versions earlier than MMSC 1.3 and IRIX 6.5 require that you use a sequence of commands to manually flash each rack. This is **Method 1 - manual**.
- If the firmware in the MMSC that you are flashing is not functional; that is, you cannot connect to it or issue commands to it, you must force the firmware into serial downloader mode. To do this, connect the control/display panel on rack 1 to the affected MMSC, and perform a sequence of key combinations. This is **Method 2**.

- MMSC firmware version 1.2 and later supports hardware flow control on COM 2 and COM 3. These ports are cabled to the MSCs in the upper and lower bays. MSC firmware level 3.1 and earlier does not support hardware flow. Therefore, if your MMSC is at revision 1.2 or later, and your MSCs are at 3.1 or earlier, you must disable hardware flow control on ports COM 2 and COM 3.

Method 1: Automatic flashing of the MMSC

You may use Method 1 when:

- MMSC firmware is functional (able to accept commands)
- MMSC firmware is revision 1.3 or later
- IRIX operating system is revision 6.5 or later

Method 1 takes about 3 minutes per rack.

The following configuration allows you to issue commands to the MMSC: Indy workstation serial port **ttyd2** is connected to the console port (**COM1**) in rack 1's MMSC.

Perform the following steps to automatically flash the MMSC:

1. Establish a connection from the Indy workstation to the MMSC with the following `cu` command: **`cu -l ttyd2`**

2. Bring up IRIX either by selecting **option 1** or by issuing the **auto** command from the PROM monitor menu.

   Note: You must be logged in as **root** at the system console (COM 1) to use this command.

3. Issue the following IRIX command: **`flashmmsc -a`**

   Note: The message `Your MMSC firmware on all racks has now been upgraded` indicates that the firmware transfer is complete.

   The `flashmmsc -a` command performs the following functions:

   - Probes the system racks and modules and outputs the firmware revisions.

Note: The `flashmmsc -p` command also probes the system racks and modules and outputs the firmware revisions; however, you must be logged in as root at the system console (COM 1) to use this command.

- Kills the MMSC daemon.
- Sets rack 1's MMSC COM 1 port and the BaseIO's ttyd1 port's transfer rate to 115,200 Kbaud.
- Clears nonvolatile storage in the first MMSC.
- Transfers the MMSC binary image to the MMSC from the default path, **`/usr/cpu/firmware/mmscfw.bin`**. Refer to the man page to set `flashmmsc` to use an alternate path. **mmscfw.bin** is the latest version.
- Clears nonvolatile storage and then transfers the binary image to rack 2's MMSC through the inter-MMSC private Ethernet.
- Determines which, if any, MSCs support hardware flow control and disables flow control on the ports of all others.
- Resets the ports to 9600 baud.
- Resets the MMSCs.

Alternate Method 1: Manual flashing of the MMSC

Manually flash the MMSC when:

- MMSC firmware is at an earlier version than 1.3 (the comment in the screen output that says "less than 1.1" is incorrect)
- IRIX operating system is earlier than version 6.5

Perform the following steps to manually flash the MMSC from the local IRIX operating system.

1. Determine whether the hardware flow control is supported; if it is not supported, disable the hardware flow control on the MMSC ports that are connected to MSCs that do not support the hardware flow control.
2. Enter the following command at the system console to stop the MMSC daemon: **`/etc/init.d/sn0start stop`**

3. Enter the following command to manually flash the MMSC:
   `flashmmsc -m`
4. When prompted to enter an MMSC command, enter `^t rack 1 flash`. The rack number should be the number of the rack that contains the MMSC that you want to flash.

   The IRIX operating system upgrades the MMSC firmware. When the upgrade is complete, IRIX outputs the following message:
   `Transferred new image to MMSC`
5. Enter the following command to reset the MMSCs: `r * reset_mmsc`
6. When the console appears to hang, press the **Enter** key to return to the IRIX prompt.
7. Enter the following command to restart the MMSC daemon:
   `/etc/init.d/sn0start start`

Note: The MMSCs in rack 1 are now flashed.

8. Enter the following MMSC command to determine the firmware level of the MSCs: `version`

Note: If the MSC firmware level is below 3.2 and the hardware flow control is turned on in MMSC COM ports 2 and 3, you will not be able to communicate with the MSCs.

9. Enter the following MMSC commands to check the status of the MMSCs' COM ports: `com 2` and `com 3`

   If hwflow is set to "Y", enter the following MMSC commands to turn off the hardware flow: `com 2 hwflow n` and `com 3 hwflow n`.

Note: Do **NOT** change the baud rates of the COM ports on the MMSC to increase the flashing speed. Leave all settings at 9600.

Method 2: Flashing with the MMSC serial downloader from an Indy workstation

Use this method to flash the MMSC from a workstation when the MMSC firmware is not functional.

Before you begin, ensure that both the firmware image and the workstation program reside on the workstation disk. They are located at `/var/tmp/microcode`. The firmware image is called **mmscfw.bin** and the downloader program is called **flashmmsc-6.n**, where **n** is the revision of the IRIX operating system that is installed on the workstation.

Note: This method requires that the MMSC display/control panel be cabled directly into the MMSC that you want to force into downloader mode. For example, if you need to set the serial downloader mode in rack 2, you must move the MMSC control panel cable from rack 1 to rack 2.

1. Remove the MMSC power cord and immediately reinstall the cord.
2. Quickly move to the MMSC display panel, then press and hold the **Down** and **Execute** buttons until the downloader mode is entered. The MMSC front panel display indicates whether the MMSC is in downloader mode.

   Note: Once you set downloader mode with this method, the only way to reset the serial downloader mode flag is to download a firmware image.

3. Set up a UNIX window to issue commands to the workstation that is running the IRIX operating system.
4. Cable the workstation serial port ttyd1 to the COM port 5 of the MMSC that you are going to flash.
5. Enter the following command to download the firmware from the workstation: **`./flashmmsc6.2 -d -1 /dev/ttyd1 -f mmscfw.bin`**

   The message `Download complete, disconnecting from MMSC...Unable to restore termio info for serial line: Bad file number` indicates that the firmware has been successfully loaded.

Method 3: Flashing the MMSC from a PC

Use Method 3 when:

- An Indy workstation is not available to flash an MMSC
- An MMSC firmware binary image is loaded on the PC
- A terminal emulator program that is capable of XMODEM transfers is loaded on the PC; the program HyperTerminal will work

Method 3 also requires the following configuration:

- A terminal connected to the **console** (COM 1) port of the MMSC
- A PC connected to the **alternate console** (COM 5) port of the MMSC that you are going to flash via the PC's serial port
- The PC's terminal emulator program has the following configuration:
  Baud rate - 19,200 baud
  VT100 terminal type
  Data to 8
  None
  1
  Connected directly to the PC's COM 1
- The serial downloader program in the MMSC automatically sets COM port 5 to 19,200 when invoked and sets it to 9600 when the load is complete

Perform the following steps to flash the MMSC firmware from the PC:

1. Enable serial downloader by power cycling the MMSC and then pressing the **Down** and **Enter** buttons until the **Serial Downloader Enabled** message appears on the MMSC display panel screen.

   The message **Upload new image via Xmodem protocol** should appear on the PC. Start the transfer at the PC by using Xmodem protocol.

   You should see a series of dots on the MMSC front panel display as the firmware loads.

2. When the transfer completes, the MMSC resets itself and the message **MMSC reset complete** appears on the PC.

Flashing the Node Board

The PROMs need to be reprogrammed (flashed) whenever there is a node board upgrade. Node board PROMs may be flashed from the PROM monitor or from the IRIX OS using the `inst -f` command. Both methods use the PROM code from the system drive in the following file:

`/usr/cpu/firmware/ip27prom.img`

Flashing the IP27 and IP31 Node Boards from the IRIX OS

Use the following command:

```
flash -c
```

Flashing the IP27 and IP31 Node Boards from a CD-ROM

Use the following command:

flashcpu cdrom(0,6,7)/*pathname*

Flashing the IP27 Node Board from the PROM Monitor

Enter the following command on a single line:

flash -c -m *module number* **-s** *node number* **(dksc0,1,0)/usr/ cpu/firmware/ip27prom.img**

Flashing the BaseIO PROMs

Ensure that you flash the BaseIO PROMs to the latest revision level before you perform any upgrade procedures.

Power-on diagnostics will determine whether a BaseIO PROM mismatch exists.

Flashing the BaseIO from the PROM Monitor

```
flashio dksc(0,1,0)/usr/cpu/firmware/io6prom.img
```

Flashing the BaseIO from the IRIX OS

```
flash -i
```

Flashing the BaseIO from a CD-ROM

flashio cdrom(0,6,7)/*pathname*

Interpreting Status LEDs

This section describes the following topics:

Origin 200 Front Panel LEDs

Table 71. Origin 200 Front Panel Status LEDs

Front Panel LED Status	Meaning
Red, then Amber, then Green	MSC check OK
Green - blinking	System booting
Green - solid	System is up and OK
Amber - blinking	Normal power-down
Amber - solid	Fan failure (system still running)
Red - blinking during power-down	Power supply problem
Red - blinking then shut-down	Multiple fan failures, and/or over temperature condition
Red - solid then shut-down	Power supply failure during boot, or overtemperature condition

Node Board

Each node board has two side-by-side columns of eight discreet LEDs (refer to Figure 72). The left column presents a status value from CPU A, and the right column presents a status value from CPU B.

Table 72 describes the node board boot process LED values.

Table 73 describes the node board failure LED values.

Table 74 describes the node board LEDs that blink when an exception occurs before there is a tty port available to receive information about the exception. If the LEDs on a node board display any of these patterns, disable or replace the node board because it contains failing hardware.

Figure 72. Node Board LEDs

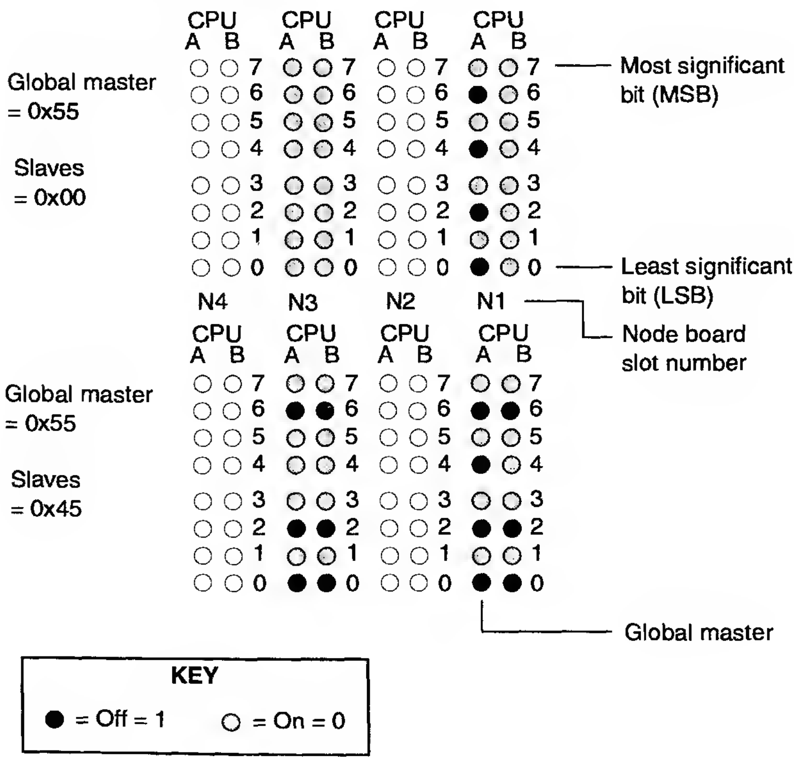


Table 72. Node Board Boot Progress LED Values

LED Value	Name	Description
0x01	INITCPU	Initialize the general-purpose registers (GPRs), floating-point registers (FPR), and COP0 registers.
0x02	TESTCP1	Test the COP1 registers.
0x03	RUNTLB	Switch to mapped mode.
0x04	TESTICACHE	Test the primary instruction cache.
0x05	TESTDCACHE	Test the primary data cache.
0x06	TESTSCACHE	Test secondary cache.
0x07	FLUSHCACHES	Flush all caches.
0x0a	INVICACHE	Invalidate the primary instruction cache.
0x0b	INVDCACHE	Invalidate the primary data cache.
0x0c	INVSCACHE	Invalidate secondary cache.
0x0d	INMAIN	Successfully jumped to the `main()` function.
0x0e	SPEEDUP	Prepare to increase PROM access speed.
0x0f	SPEEDUPOK	Successfully increased PROM access speed.
0x1a	MSCPROBE	Prepare to probe for presence of MSC.
0x1c	DONEPROBE	Completed the probe for the presence of the MSC.
0x1d	UARTINIT	Prepare to initialize selected UART.
0x1e	UARTINITDONE	Completed the initialization of the selected UART.
0x21	PODLOOP	Prepare to enter POD mode (C code portion).
0x22	PODPROMPT	Prepare to enter the POD prompt loop.
0x23	PODMODE	Prepare to enter POD mode (assembly code portion).
0x24	LOCALARB	Perform local arbitration (between CPU A and CPU B).
0x28	BARRIER	Prepare to perform first local barrier.

Table 72. Node Board Boot Progress LED Values (continued)

LED Value	Name	Description
0x2a	MAKESTACK	Prepare to configure Dex mode stack and data.
0x2b	MAIN	Code execution has reached the `main()` function.
0x31	NMI	Received external nonmaskable interrupt.
0x35	RTCINIT	Prepare to initialize the Hub real-time counter.
0x36	RTCINITDONE	Completed the initialization of the Hub real-time counter.
0x38	BARRIEROK	Successfully completed the first local barrier operation.
0x3c	JUMPRAMU	Prepare to jump to UALIAS space.
0x3d	JUMPRAMUOK	Successfully jumped to UALIAS space.
0x3e	JUMPRAMC	Prepare to jump to cached space.
0x3f	JUMPRAMCOK	Successfully jumped to cached space.
0x40	STACKRAM	Prepare to test the stack area of memory.
0x41	STACKRAMOK	Successfully tested the stack area of memory.
0x45	LAUNCHLOOP	Prepare to enter the slave launch loop.
0x46	LAUNCHINTR	Received a launch interrupt.
0x47	LAUNCHCALL	Call the `launched()` function.
0x48	LAUNCHDONE	Returned from the `launched()` function.
0x4a	MDIRINIT	Prepare to initialize the Hub MD and SIMM controls.
0x4b	MDIRCONFIG	Prepare to determine and configure the memory size.
0x4c	I2CINIT	Prepare to initialize the PCF8584 I2C chip.
0x4d	I2CDONE	Completed the initialization of the PCF8584 I2C chip.
0x4f	IODISCOVER	Prepare to discover Hub I/O.

Table 72. Node Board Boot Progress LED Values (continued)

LED Value	Name	Description
0x50	HUB_CONFIG	Prepare to write Hub configuration information into the KLCONFIG structure.
0x51	ROUTER_CONFIG	Prepare to write the router configuration information into the KLCONFIG structure.
0x52	INITIO	Prepare to initialize the I/O section of the Hub.
0x53	CONSOLE_GET	Prepare to probe the I/O section to locate the console.
0x54	CONSOLE_GET_OK	Successfully completed the probe of the I/O section for the console.
0x56	INITIODONE	Completed the initialization of the I/O section of the Hub
0x57	STASH2	Reset error state saved.
0x58	STASH3	Clear Hub error registers.
0x59	STASH4	Enable Hub error checking.
0x5a	IODISCOVER_DONE	Completed the discovery of the Hub I/O.
0x5b	NMI_INIT	Prepare to initialize the NMI handler area.
0x5c	TEST_INTS	Prepare to test Hub interrupts.
0x5d	IORESET	Prepare to perform early reset of Hub I/O section.

Node Board Failure LED Values

In addition to indicating boot progress, the LEDs indicate hardware failures detected by PROM diagnostics during the boot phase. If a fatal problem is found, the CPU sets the LEDs to a failure value between 0x80 and 0xa1 (as shown in Table 73) and automatically disables itself.

- This is the only indication of errors detected during power-on.
- If a CPU fails before the ACP is available, a solid code is displayed.
- If a CPU fails after the ACP is configured, a flashing code is displayed and that CPU enters Dex POD mode at the ACP.
- When PROM Monitor is entered, the LEDs display the following pattern (refer to Figure 72 on page 216):
  - Global Master = solid hex 55
  - Slave CPUs = flash a hex 00 and 45

Table 73. Node Board Failure LEDs Values

LED Value	Name	Reason
0x81	CP1	Register test failed.
0x82	RESTART	Restart master was unable to load the BaseIO PROM.
0x83	ICACHE	Primary instruction cache test failed (The failing FRU is the node board).
0x84	DCACHE	Primary data cache test failed (The failing FRU is the node board.)
0x85	SCACHE	Secondary cache test failed (The failing FRU is the node board).
0x86	KILLED	CPU disabled by another node.
0x87	RTC	Real-time counter not counting.
0x91	HUBLOCAL	Hub local arbitration failed; ignore this CPU.
0x93	PREM_DIR_REQ	All nodes must have premium DIMMs for this configuration.
0x97	MAINRET	Returned from `main()` function.
0x98	NOMEM	Node board does not have local memory.

Table 73. Node Board Failure LEDs Values (continued)

LED Value	Name	Reason
0x9a	DISABLED	CPU disabled by an environment variable.
0x9b	DOWNLOAD	Failure occurred while downloading the PROM code into RAM.
0x9c	COREDEBUG	Boot process cannot set the CORE debug registers.
0x9d	IODISCOVER	Failure occurred during the Hub I/O discovery process.
0x9e	HUB_CONFIG	Failure occurred while writing the Hub information into the KLCONFIG structure.
0x9f	ROUTER_CONFIG	Failure occurred while writing the router information into the KLCONFIG structure.
0xa0	HUBIO_INIT	Failure occurred while trying to initialize the Hub I/O interface.
0xa1	CONFIG_INIT	Failure occurred while trying to initialize the KLCONFIG structure.
0xa2	RTRCHIP	Failure occurred while testing the Router chip.
0xa3	LINKDEAD	Failure occurred while testing the LLP link.
0xa4	HUBBIST	Failure occurred while the Hub chip executed the built-in self test (BIST).
0xa5	RTRBIST	Failure occurred while the router chip executed the built-in self test (BIST).
0xa6	RESETWAIT	Waiting for a reset to occur.
0xa7	LLP_FAIL	LLP failed after the reset.
0xa8	LLP_NORESET	LLP never came up after the reset.
0xa9	BADMEM	Local memory is corrupted.
0xab	NET_DISCOVER	Failure occurred for the Hub network discovery.
0xac	NASID_CALC	Failure occurred for the NASID calculation.
0xad	ROUTE_CALC	Failure occurred for the route calculation.

Table 73. Node Board Failure LEDs Values (continued)

LED Value	Name	Reason
0xae	ROUTE_DIST	Failure occurred for the route distribution.
0xaf	NASID_DIST	Failure occurred for the NASID distribution.
0xb0	NO_NASID	Master did not assign a NASID.
0xb1	NO_MODULEID	Failure occurred for the module ID arbitration.
0xb2	MIXED_SN00	Origin 200 system is configured with an Origin 2000 system.
0xb3	ERRPART	Failure occurred in the partition configuration.
0xb4	MODEBIT	Failure occurred while copying the processor mode bits.
0xb5	BACK_CALC	Failure occurred while calculating the midplane frequency.

Table 74. LED Status Values for Exceptions

LED Value	Name	Description
0xf2	EXC_GENERAL	(Blinking) General exception
0xf3	EXC_ECC	(Blinking) ECC exception
0xf4	EXC_TLB	(Blinking) TLB exception
0xf5	EXC_XTLB	(Blinking) XTLB exception
0xf6	EXC_UNIMPL	(Blinking) Unimplemented exception
0xf7	EXC_CACHE	(Blinking) Cache error exception

Router Board

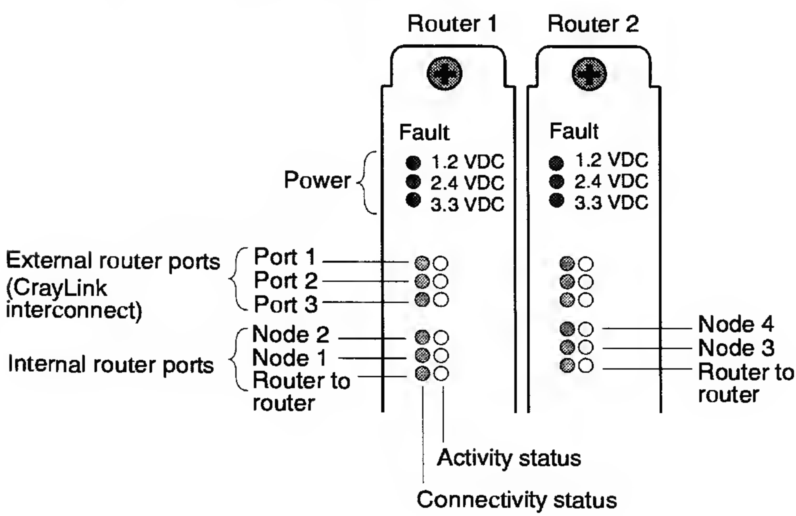
Note: The terms *port* and *link* have the same meaning for the router and MetaRouter boards.

The router board has two side-by-side columns of six discrete LEDs and three power LEDs on the top of the board (refer to Figure 73).

The left-side (green) LEDs indicate the connection status of router ports 1 through 6. They illuminate if a port has successfully maintained a connection with another device. The left-side LEDs turn off if the port is disconnected (physically or due to severe connection errors).

The right-side (yellow) LEDs are controlled by system software and typically illuminate when there is traffic across the corresponding link. Even though ports 4 through 6 are on the Origin 2000 and Onyx2 rack module midplane, the connection LEDs are still useful because they can indicate whether a node board is improperly seated. Only the rack router boards have status LEDs on the front bezel. The null and star router boards do not have LEDs.

Figure 73. Router Board LED Status

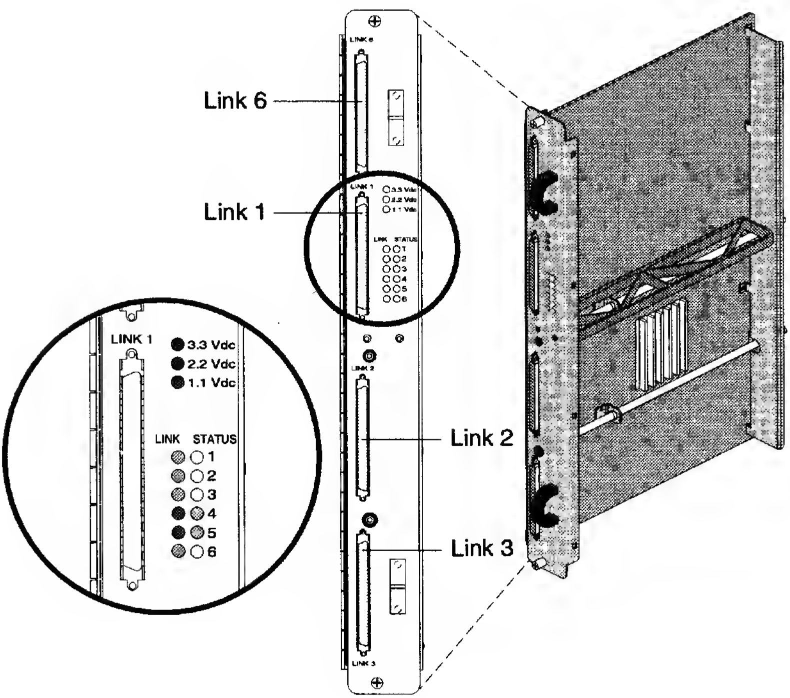


MetaRouter

The MetaRouter board contains a single router ASIC with six ports: 1 through 6. Four of the ports go to connectors located on the front panel of the board as shown below; these are external cable connectors and are labeled Link 6, 1, 2, and 3. Ports 4 and 5 go to the rear of the board and connect to the midplane through CPOP connectors; the 128-processor configuration uses these midplane connections for clock distribution and to read the midplane NIC.

Note: The terms *port* and *link* have the same meaning for the router and MetaRouter boards. Refer to Figure 73 for LED descriptions.

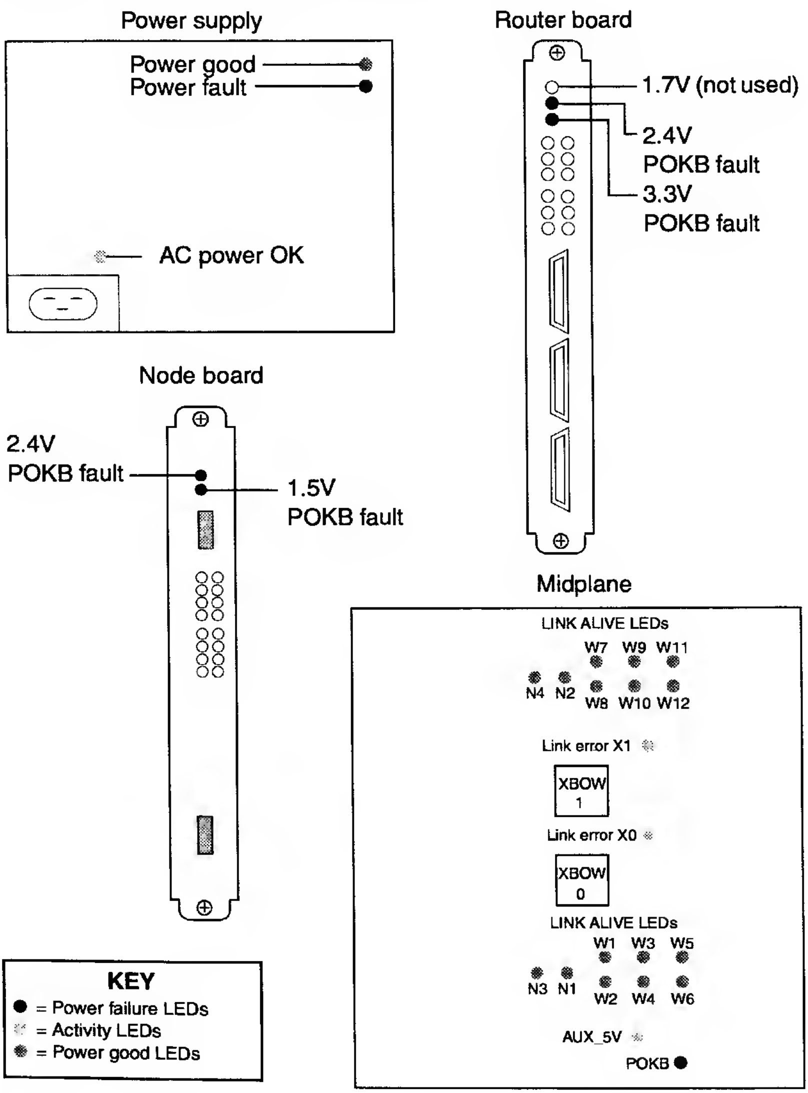
Figure 74. MetaRouter Board LEDs



Power LEDS

Figure 75 shows the power and status LEDs in an Origin series system.

Figure 75. Origin 2000 Power and Status LEDs



XIO-to-PCI Converter LEDs

Refer to Figure 76 and Table 75 for a description of the XIO-to-PCI Converter LEDs.

Figure 76. XIO-to-PCI Converter LEDs

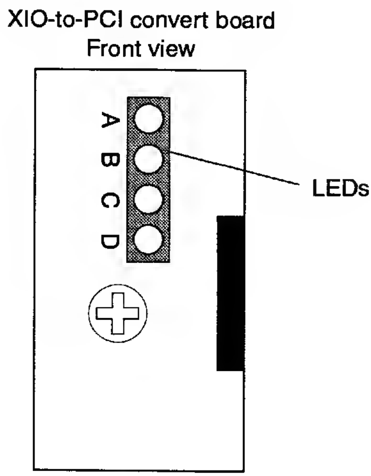


Table 75. XIO-to-PCI LED Descriptions

LED	Description
LED A: 33 MHz PCI Clock (Green)	On indicates PCI clock functioning at 33 MHz. Off indicates PCI clock not functioning at 33 MHz.
LED B: 100 MHz PCI Clock (Green)	On indicates PCI clock functioning at 100 MHz. Off indicates PCI clock not functioning at 100 MHz.
LED C: Buffer Full (Green)	On indicates buffer is full. Off indicates buffer is not full.
LED D: Max Retry Timeout (Green)	On indicates max retry timeout is in process. Off indicates max retry timeout is not in process.

Enabling IP27 Memory Banks

For the rare condition when all the IP27 memory banks are disabled, use the following procedure to re-enable the banks.

Also, this procedure is necessary only when you cannot get to the PROM monitor to use the `enable` command.

1. Power off the system.
2. Load the IP27 with known good memory from another board.
3. Install the IP27 in node 1 slot and back out all other nodes.
4. Attach a terminal to the MSC diagnostic port.
5. Power on the system.
6. At the terminal, enter **`Ctrl-t`** and then enter **`dbg 0 18`**.
7. At the terminal, enter **`Ctrl-t`** and then enter **`rst`**.

   The system resets and stops in memoryless POD mode (Dex).
8. From POD, enter **`go cac`**.
9. From POD, enter **`printenv`**.

   You should see the environmental **`DisableMemMask`** set.
10. From POD, enter **`unsetenv DisableMemMask`**.

    This causes an exception and drops POD back to Dex mode.
11. From POD, enter **`dbg 0 0`**.
12. From POD, enter **`reset`**. This enables the IP27 memory banks.

Entering POD Mode from the Command Monitor

1. Enter the following command to enter POD mode:

```
>> pod
Switching into Power-On Diagnostics mode...

1A 000: *** Software entry into POD mode from IO6 POD mode on node 0
1A 000: POD IOC3 Dex>
```

2. Switch to cache mode (refer to "POD Modes" on page 175) by entering the following command:

```
1A 000: POD IOC3 Dex>  go cac
Testing/Initializing memory

1A 000: *** Requested CAC mode on node 0
1A 000: POD IOC3 Cac>
```

3. If the system enters POD mode automatically, you can use the **why** command to determine why the system entered POD mode.

Reading the Hardware Debug Switches from POD

You can modify the hardware debug switches from POD by using the `dips` and `dbg` commands.

The `dips` command reads the MSC hardware switch block and debug switches (virtual and physical), performs an exclusive-OR function, and shows you the results.

The `dbg` command, like the MSC command `dbg`, sets or clears MSC debug switches.

Enter the following command to display the state for the hardware debug switches. In this example, hardware debug switch 3 is in the on position.

```
1A 000: POD IOC3 Cac> dips

Debug Switch No.:           16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01
ELSC dbg switches   00 00   -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
Hardware switches      00                           -- -- -- -- -- xx -- --
Final (dbg XOR hw)  00 00   -- -- -- -- -- -- -- -- -- -- -- -- -- xx -- --
```

-- = off
xx = on

Entering POD Commands on a Different CPU

Most POD commands affect only the target CPU, which the POD prompt indicates. If you want to enter commands on a CPU other than the global master, you must use one of the following methods:

- Use the command monitor or POD disable command to disable all the CPUs except the one that you want to access (not recommended).
- Use the POD command `cpu` to put the target CPU into the active state and inactivate the current global master CPU.

Note: POD and the MSC view the node numbering differently. The following table illustrates the numbering relative to POD and the MSC.

Physical Slot	1	2	3	4
POD node numbers	0	1	2	3
MSC node numbers	1	2	3	4

Syntax for the `cpu` command is: `cpu [[n:`*NODE*`] a|b]`

`n:`*NASID* = the NASID number
`a|b`= the target CPU

Example: Connect to CPU A of NASID 2

```
1A 000: POD IOC3 Cac>  cpu n:2 a

Switching to node 2 CPU A

3A 002: *** Taking over pod mode on node 2
3A 002: POD IOC3 Cac>
```

Example: Connect to CPU A of NASID 0

```
3A 002: POD IOC3 Cac> cpu n:0 a

Switching to node 0 CPU A

1A 000: *** Taking over pod mode on node 0
1A 000: POD IOC3 Cac>
```

Nullifying the MSC Password from POD Mode

If the MSC password is changed through a terminal that is attached directly to the ACP port, the MMSC will not recognize the new password and will not allow you to issue restricted commands from the MMSC to that MSC.

This condition is not a concern if you can move the key-switch to the diagnostic position. It is a problem, however, if you access the system from a remote location and no one is available to move the key-switch for you.

If you can get into POD remotely, then you can use the `pas` command (same name as the MSC `pas` command described on page 163) to read or set a password.

Enter the following command to nullify the password:

```
1A 000: POD IOC3 Cac> pas

ELSC password is: none
1A 000: POD IOC3 Cac>
```

Nullifying the Command Monitor Password

If entry into the command monitor is blocked by an unknown password, use the following procedure to remove it. Access to this computer system must be specifically authorized by the owner. Unless you are so authorized, your access may expose you to criminal and/or civil proceedings.

Caution: This procedure is not intended for non-SGI service personnel. It is included here for SGI System Support Engineers (SSEs).

If a password has been installed on the command monitor, first check with site personnel to verify that the password is valid.

This procedure is provided so SSEs can access the Command Monitor in an emergency situation.

1. Under the IRIX OS, log in as `guest`, and display the PROM variables by entering the `nvram` command.

   Caution: The following procedure not only sets the command monitor password to its default setting but also sets many of the other variables to their defaults, such as `netaddress`, `root`, `SystemPartition`, `OSLoadPartition`, etc.

2. Shut down IRIX and enter POD mode by entering `nmi` at the module containing the console BaseIO board (look for the C in the MSC display):

   ```
   Ctrl-t
   nmi <Enter>
   ```

   Note: You must perform the following operation from the global master (NASID = 000).

   If your system enters POD at another CPU (prompt is not `1A 000: POD MSC Dex>` or `3A 000:POD MSC Dex>`), then enter the following commands:

   ```
   ctrl-t
   sel 1 a <Enter
   ```

3. Enter the following command to enter cac mode:

   ```
   go cac <Enter>
   ```

To invalidate the NVRAM checksum, you will first enter the `lb` command to display the value of a byte, and then enter the `sb` command. Set the byte to some other value.

Note: If you are running microcode 5.18 or below, use the address 0x9200000086804bc in the following example. If you are running microcode 5.19 or above, use 0x9200000008680438 in place of the address shown below.

4. Enter `lb` (load byte) as shown below to display a dedicated location in NVRAM; then enter `sb` (store byte) to set the memory location to something other than its current setting to invalidate the checksum in the NVRAM.

   The example below shows that the byte was set to 09 (it will be set to 0 as shown below).

```
3A 000: POD IOC3 Cac> lb 0x920000000868Oxxx
92000000086804bc: 09
```

For an Onyx2 deskside system, change the 8 to an F.

```
3A 000: POD IOC3 Cac> sb 0x920000000868Oxxx 0
```

Where *xxx* = **4bc** if running microcode 5.18 or below,
or
438 if running microcode 5.19 or above.

0x92000000086804bc

6804bc is an address in NVRAM
8 is the XBOW port number
00 is the NASID
92 selects IO address
0x indicates the value is a hexadecimal number

The address (0x920000000868O*xxx*) is a dedicated location in the address map that points to the NVRAM on the console BaseIO.

5. Enter **`reset`** to reset the system.

6. Enter the command monitor and restore any variables that were reset to their default settings such as `netaddr` and `ProbeAllScsi`.

Correcting a PROM Revision Mismatch

You must use this procedure if the boot process hangs before the PROM monitor appears. This condition can occur when you replace an IP27 node board with a spare that has a defective PROM or a lower-level PROM version that cannot be reflashed automatically during the power-on boot process.

The POD's `flash` command copies the microcode from the global master to a specified IP27 node board.

Important: The target IP27 node board must have the same clock speed and secondary cache size as the global master.

If you cannot flash the IP27 because of a firmware version mismatch, remove all of the IP27 boards and install only the IP27 board with the lower-level firmware to determine whether you can enter the PROM monitor. If you can enter the PROM monitor on the IP27 board with the lower-level firmware, flash it using the "Flashing the Node Board" procedure on page 214 and then install all of the other node boards when the flash is complete.

Use the following procedure if the system hangs due to a PROM revision mismatch:

1. Enter the following commands at the ACP (ASCII terminal) to enter POD mode:

   Ctrl-t
   ech 1
   Ctrl-t
   MSC> **dbg 0 8** *Set debug switch 4 (Table 55 on page 170)*
   Ctrl-t
   MSC> **rst**

   The system reboots and enters POD Cac mode

```
*** Warning: ELSC debug (dbg) switches are non-zero
*** Boot stop requested at Global (2)
.
.
.
1A 000: *** Entering POD mode on node 0
1A 000: POD IOC3 Cac>
```

2. Use the `pcfg` command to display the configuration data structure. The configuration data contains the system topology, which will identify the IP27 node with the lower-level microcode version.

   ```
   1A 000: POD IOC3 Cac> pcfg
   ```

3. Now that you have identified the IP27 node with the lower-level microcode version (for example, IP27 node 2), use the POD mode `flash` command to update the firmware for node 2.

   ```
   1A 000: POD IOC3 Cac> flash  2
   ```

4. Set debug switch 4 to off.

   ```
   1A 000: POD IOC3 Cac> dbg 0 0
   ```

5. Reset the system.

   ```
   1A 000: POD IOC3 Cac> reset
   ```

Testing Memory with POD

The power-on diagnostics will detect solid memory errors, but when you experience intermittent failures, you can manually test memory and change the voltage margins to help locate the failure.

Use the following commands to test memory, clear memory errors, and display memory errors:

- The `memtest` command tests main memory. Specifically, it tests the data and ECC portion of each memory DIMM.
- The `dirtest` command tests the directory portion of each memory DIMM, and the addressing structure for directory memory is the same as main memory.
- The `clear` command clears memory errors.
- The `error` command displays memory errors.

Perform the following steps to test memory. For this example, bank 2 of NASID 4 is being tested.

1. From the System Maintenance Menu, select option 5 to enter the command monitor.
2. Enter POD mode.
   ```
   >> pod
   ```
3. Go into cache mode.
   ```
   1A 000: POD IOC3 Dex> go cac
   ```

4. Test the memory of bank 2, NASID 4.

```
1A 000: POD IOC3 CAC> memtest n:4 b:2 u:1m 128m
```

The command displays the following failure information:

```
        Store/load
--- MEMORY FAILURE (caught exception, node slot 1) ---
No hub MD error registered
CAUSE    0x000000000000c01c (INT:87------ <Data Bus Err>)
Address 0x9600000448000000
1A 000: POD IOC3 Cac>
```

Note: If you attempt to test a memory location that is not present, the command displays the following error message:

```
        Store/load
--- MEMORY FAILURE (caught exception, node slot 1) ---
CAUSE    0x000000000000c01c (INT:87------ <Data Bus Err>)
Address 0x9600000620400000
No MD errors found
1A 000: POD IOC3 Cac>
```

5. Break down the address.

```
1A 000: POD IOC3 Cac> pa u:0x448000000
```

The pa command displays the break down of the address:

```
Addr 0x9600000448000000:
       bit 0 is in UNCACHED MEMORY space on NASID 4
/hw/module/2/slot/n1: MMXL2 line 0
```

Use the information in the following subsections when troubleshooting memory.

Memory DIMM Locations and Combinations

Table 76 through Table 79 list the Origin series memory DIMM location, and combinations for main memory and extended memory. Figure 77 and Figure 78 show the DIMM locations on the node board.

Table 76. Main Directory DIMMs

DIMM Type	Chip Depth	Chip Width	Number of Chips/DIMM	SDRAM
A Yellow	2 1	8 16	18 1	16 Mbits
B Green	4 2	4 8	36 2	16 Mbits
C Red	16 8	4 8	36 2	64 Mbits

Table 77. Extended Directory DIMMs

DIMM Type	Chip Depth	Chip Width	Number of Chips/DIMM	SDRAM
D White	2	8	8	16 MB
E Blue	8	8	8	64 MB

Table 78. DIMM Location Chart

DIMM Triplet	DIMM Locations	Bank Number
1st	MMXL0, MMXH0, DIRX0	Bank 0
2nd	MMXL1, MMXH1, DIRX1	Bank 1
3rd	MMXL2, MMXH2, DIRX2	Bank 2
4th	MMXL3, MMXH3, DIRX3	Bank 3
5th	MMYL4, MMYH4, DIRY4	Bank 4
6th	MMYL5, MMYH5, DIRY5	Bank 5
7th	MMYL6, MMYH6, DIRY6	Bank 6
8th	MMYL7, MMYH7, DIRY7	Bank 7

Table 79. Corresponding Main Memory and Directory Memory Colors

Main Memory DIMM Size	Total Memory Per DIMM Pair	Main Memory DIMM Color	Directory Memory DIMM Color
32 MB	64 MB	Yellow	White
64 MB	128 MB	Green	White

Table 79. Corresponding Main Memory and Directory Memory Colors

Main Memory DIMM Size	Total Memory Per DIMM Pair	Main Memory DIMM Color	Directory Memory DIMM Color
128 MB	256 MB	Silver	Blue
256 MB	512 MB	Red	Blue

Figure 77. DIMM Combinations

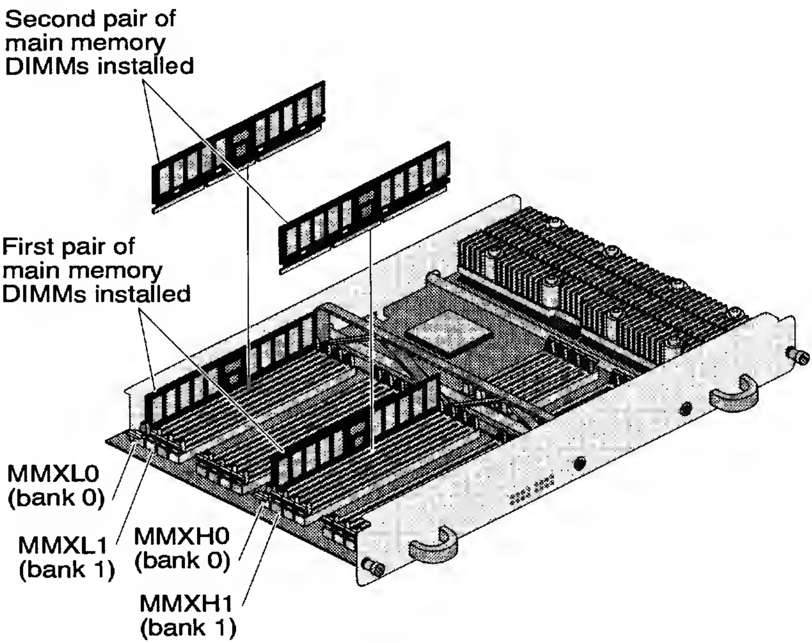
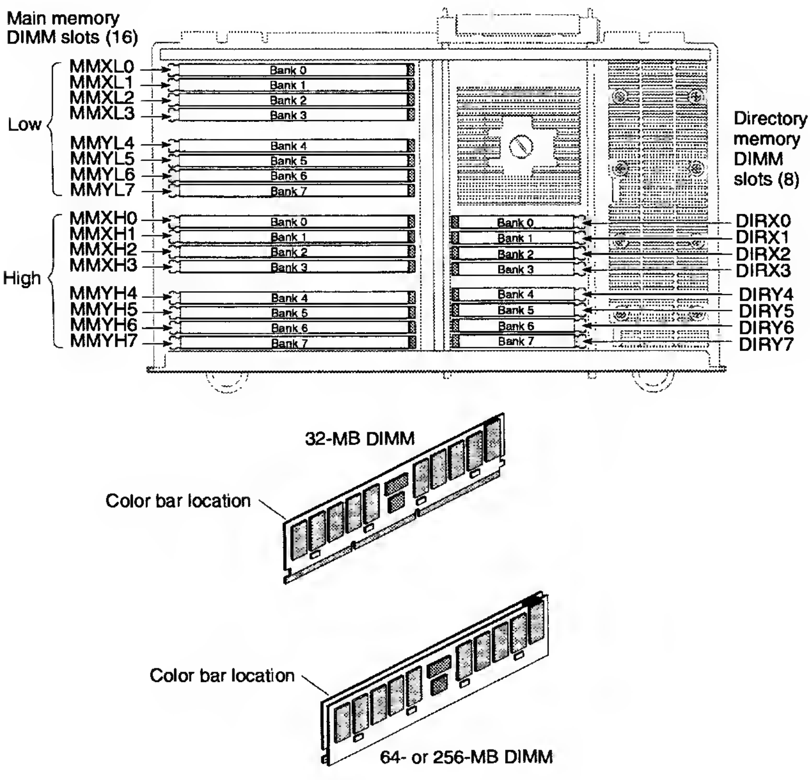


Figure 78. Main Memory and Directory DIMM Locations



Memory Addressing

Note: Origin systems use a 40-bit address to select a node, a bank on the node, and an address within the bank. (Refer to Figure 78.)

Address Bits	Bit Selection
39 – 32	NASID
31 – 29	Bank
28 – 0	Address

Note: Use the POD command `pcfg v` to determine the slot number of the node board.

Figure 79. Physical Memory Addresses

Node select bits (NASID) 1 of 256 — Banks select bits 1 of 8 — Address select bits 1 of 512

Decimal value									Bank bit 0	Bank bit 1	Bank bit 2	256M	128M	64M	32M	16M	8M	4M	2M	1M	512K	256K	128K	64K	32K	16K	8K	4K	2K	1K	512	256	128	64	32	16	8	4	2	1
Address bit	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0
Hex value	0				2				0				3				7				b				6				5				4				9			

NASID 2 — Bank Bits — Internal Address

Running MDK-based Tests

Table 80 provides quick-reference information and the command line syntax for running the MDK-based tests. Enter the commands shown in Table 80 from the command monitor prompt.

Table 80. MDK-based Tests Quick Reference

MDK Command Line Syntax	Failing FRU
`>>boot dksc(0,1,0)/stand/t5r.mdk` `MDK> t5-1`	Node board
`>>boot dksc(0,1,0)/stand/t5r.mdk` `MDK> t5-2`	Node board
`>>boot dksc(0,1,0)/stand/memory.mdk` `MDK> scache`	Node board
`>>boot dksc(0,1,0)/stand/memory.mdk` `MDK> mem.qs`	DIMM, node board, midplane, or router board
`>>boot dksc(0,1,0)/stand/memory.mdk` `MDK> memum.qs`	DIMM, node board, midplane, or router board
`>>boot dksc(0,1,0)/stand/memory.mdk` `MDK> memum.nb`	DIMM, midplane, or node board
`>>boot dksc(0,1,0)/stand/memory.mdk` `MDK> memum.in`	DIMM, node board, midplane, or router board
`>>boot dksc(0,1,0)/stand/memory.mdk` `MDK> mem.lo`	DIMM, node board, midplane, or router board
`>>boot dksc(0,1,0)/stand/memory.mdk` `MDK> memum.lo`	DIMM, node board, midplane, or router board
`>>boot dksc(0,1,0)/stand/t5r.mdk` `MDK> router`	Router board, node board, midplane, or cable
`>>boot dksc(0,1,0)/stand/stress.mdk` `MDK> ct_io_turbo`	Router board, node board, midplane, or cable
`>>boot dksc(0,1,0)/stand/MDK_router_sso` `MDK> router`	Router board, node board, midplane, or cable

Miscellaneous Diagnostic Information

Table 81 lists the directories and URLs that you may need for the diagnostic tools. Table 82 lists the directories and any commands that you may need for the system monitoring tools.

Table 81. Diagnostic Tool Information

Tool	Pertinent Directories and/or URLs
Availmon	/usr/etc/amconfig /usr/etc/amreport /var/adm/avail /var/adm/crash http://availmon.csd.sgi.com/cgi-bin/stopcode.pl
SVP	/usr/SVP/svp /usr/SVP/CONFIG /usr/SVP/RESULT
Remote access tool (rat)	/usr/bin/rat
FRU analyzer	/var/adm/crash
IRIX field diagnostics	/usr/diags/diags_interface/field_diag
Field stress diagnostics	/usr/diags/stress/fst /usr/diags/stress/RESULTS

Table 82. System Monitoring Tool Information

<table>
<tr><th>Tool</th><th>Pertinent Directories and/or Commands</th></tr>
<tr><td>xbstat</td><td>/usr/sbin/xbstat</td></tr>
<tr><td>xbmon</td><td>find /hw -name mon -print | grep xtalk
This command lists the pathnames for all XBOW ASICs.
ln -fs pathname /dev/xbmon
Note: The pathname identifies the XBOW ASIC that you want to monitor.</td></tr>
</table>

Running the field_diag User Interface

The `field_diag` user interface provides a common ASCII interface that you can use to run the IRIX based field diagnostics. You must have root privilege to run the `field_diag` user interface.

Perform the following procedure to start the `field_diag` user interface:

1. Enter the following command to change to the `/usr/diags/diags_interface/` directory:

   `cd /usr/diags/diags_interface`

2. Enter the following command to start the `field_diag` program:

   `./field_diag`

The `field_diag` interface will be displayed. `Ctrl-t` displays the output messages from the test that is currently running. Enter a lowercase `h` to toggle help on any highlighted test.

The `field_diag` program automatically identifies the hardware in the system and displays only the appropriate diagnostics.

Table 83. Field_Diag Diagnostic List

Option	Description
ATM	Selects the ATM LINC DMA verification test.
FPU-SP	Selects the floating-point unit (single-precision) test.
FPU-DP	Selects the floating-point unit (double-precision) test.
HIPPI	Selects the HIPPI-S LINC DMA verification test.
MIO	Selects the MediaIO board test.
MEMORY	Selects the memory test.
NETWORK	Selects the network thrasher test.
RAID	Selects the RAID verification test.
SCSI	Selects the SCSI thrasher test.
VHT_LOOPBACK	Selects the vicious HIPPI test (loopback test).
VHT_REMOTE	Selects the vicious HIPPI test (remote test).
VST_PING	Selects the vicious socket test (ping test).
VST_RW	Selects the vicious socket test (read and write test).
QUIT	Quits the field_diag program.

Running the Field Stress Diagnostics (FST)

The FST user interface provides a common ASCII interface that can be used to run and monitor the stress tests included in the field stress tool.

Note: You must have PERL programming language support installed on the system to use FST, and you need root privilege to run the FST user interface.

Perform the following procedure from the IRIX prompt to start the FST user interface, which lists the test options.

1. Enter the following command to change to the `/usr/diags/stress/` directory:

   ```
   cd /usr/diags/stress
   ```

2. Enter the following command to start the FST user interface:

   ```
   ./fst
   ```

Table 84. Field Stress Test Options

Test Option	Description
PANDORA	Runs the Pandora test suite for 30 minutes.
DEFAULT_STRESS_TEST	Runs a predetermined suite of stress tests for 15 minutes. This option is part of the Application test package.
GFX_STRESS_TEST	Runs a graphics stress test. (This option is available only for graphics systems.) This option is part of the Application test package.
NON_GFX_STRESS TEST	Runs a nongraphics stress test. This option is part of the Application test package.
TORPEDO	Compares test results of all processors to detect failures.
CLEAN	Deletes any processes that were left running by a previous stress test.

Test Output Files

The field stress test output files are stored in the `/usr/diags/stress/RESULTS` directory. Table 85 lists and describes each file.

Table 85. FST Test Output Files

File or Directory	Description
SYSLOG_ALL	Contains all relevant messages from **SYSLOG**.
TEST_MSG_LOG	Contains all of the messages that the field stress tool generates. It contains time-stamped entries that show which tests were run and which options were used to start them.
testcase.LOG.*timestamp*	Contains the hardware configuration information for the system being tested and a message that indicates why the test stopped. The *testcase* portion of the filename contains the filename of a test case file in the `/usr/diags/stress/TEST/` directory structure.
APPLICATIONS	This directory contains a log file for each application that the Application stress test runs. The filename format for these log files is: *testcase*.LOG.*processor_number.* The *testcase* portion of the filename contains the filename of a test case file in the `/usr/diags/stress/TEST/` directory structure. The *processor_number* portion contains the processor on which the test was run.
PANDORA	This directory contains the log files that the Pandora stress test generates. The filename format for Pandora log files is: *pandora*.LOG.*processor_number.* The *processor_number* portion of the filename contains the processor on which the test was run.

Using the linkstat Command

You can use the IRIX `linkstat` command to monitor Hub and router errors on your system. The `linkstat` command reports the following information:

- Router link performance and error rates
- Error rates on the Hub chip CrayLink network interface (NI) and I/O Interface (II) links
- Check-bit (CB) errors

  These errors indicate that a correctable LLP error occurred on a Hub or router ASIC.
- Sequence Numbering (SN) errors

  These errors indicate that data packets have arrived at their destination in the wrong order. You can ignore these errors unless they occur simultaneously with a large number of CB errors.

The `linkstat` command line options are shown below:

```
linkstat [-c] [-e] [-v] -a | hwgraph_filename
```

`-c`	Report information continuously every 5 seconds
`-e`	Only report error rates, not performance information
`-a`	Report on all links in the system

Using the xbstat Command

The `xbstat` command monitors the traffic to and from each of the active ports of the XBOW ASIC.

`xbstat` reports the following information:

- Total number of micropackets sent from and received at each link.
- The LLP retries at each link
- Cumulative counts since the beginning of the monitoring period (reported in micropackets)

If no time parameter is specified, `xbstat` monitors continuously.

Usage

```
xbstat [-t sec]
```

The `-t sec` determines the duration of time in units of seconds that the XBOW traffic will be monitored. `xbstat` is capable of monitoring only one XBOW at a time.

1. Use the IRIX command `find /hw -name mon -print` to find all possible XBOW `hwgraph` files that `/dev/xbmon` can point to.
2. Create a symbolic link between the selected `hwgraph` file and the `/dev/xbmon` file before invoking this command as shown below:

```
# ln -s /hw/module/1/slot/n1/node/xtalk/0/mon /dev/xbmon
```

After linking, the `/dev/xbmon` file should point to the selected XBOW in the `hwgraph`.

3. Enter **`xbstat`** to display the XBOW statistics as shown below:

```
Slot Flags   Source  Destination RcRtry TxRtry
IO11 0       0       0           0      0
N4   1       844     842         0      0
N2   0       0       0           0      0
IO8  0       0       0           0      0
IO10 0       0       0           0      0
IO12 0       0       0           0      0
IO7  0       0       0           0      0
IO9  0       0       0           0      0
```

The display updates automatically until you exit by entering <Ctrl-c>. If you want to view the statics of a different XBOW ASIC, you must link the `/dev/xbmon` device to the hwgraph file of the XBOW ASIC that you want to view.

The port number displayed by `xbstat` corresponds to an XIO widget number as shown in the table below:

Port Number	XIO widget ID	8P/12 Device
0	0x8	BaseIO
1	0x9	Node 1 or 2
2	0xA	Node 3 or 4
3	0xB	XIO slot 4 or 8
4	0xC	XIO slot 2 or 10
5	0xD	XIO slot 6 or 12
6	0xE	XIO slot 5 or 7
7	0xF	XIO slot 3 or 9

Troubleshooting Graphics

The following tips can help you isolate problems in the graphics subsystem:

- If the host system cannot communicate with the graphics subsystem, or behaves as if the graphics hardware does not exist, one of the following components may be the problem:
  - Origin 2000 midplane
  - Xtalk XIO board
  - Crosstown cable
  - Ktown board
  - KCAR midplane
  - GE14 (XG ASIC)

Note: You can usually isolate the failing FRUs through the use of limited board swaps.

Remember that `irsaudit` cannot diagnose the XBOW-to-GE14 path that the host system uses to communicate with the graphics subsystem. If this path is failing, look for the following console warning message:

```
WARNING: xbow_base: 0x9200000000000000 link: 14
Widget present, but link not alive!
```

The Xtalk link LED may be off if any of the components listed above are failing. Because the link LEDs are not normally visible, use the **xbmon** command to view the status of the XBOW port that connects to the Xtown board.

- If the PROM monitor takes an exception or hangs during the power-on boot process at the Initializing software and devices stage, the suspect boards are:
  - GE14
  - DG5

- When the IRIX operating system attempts to start the graphics subsystem and it takes an exception or hangs, the suspect boards are:
  - GE14
  - DG5

- If garbled text is displayed or a window cannot be opened, you should suspect the:
    - GE14
    - GE14 EEPROM microcode
- If a window opens and it has vertical lines (solid or dashed) or evenly-spaced pixel drop outs (holes), the suspect board is: RM

    For 2-RM or 4-RM board configurations, remove 1 or 3 boards to see if the problem goes away.

    Note: You cannot remove the RM0 board.

- When there is no monitor display or when the display cannot synchronize, you should suspect the:
    - DG5
    - Corrupt DG5 video RAM or GE14 EEPROM format
    - DG5 video RAM or GE14 EEPROM format that exceeds the graphics monitor's hardware specifications (invalid VOX parameters)

    Note: When you load a format into the GE14 EEPROM or DG5 Video RAM that exceeds the maximum CRT resolution or scan frequency specifications, the monitor goes black and loses synchronization.

- When colors are incorrect, the suspect boards are:
    - GE14
    - DG5

Restoring the GE EEPROM

When the microcode stored on the GE14 board becomes corrupted, or you download a bad combination format that causes the monitor to lose synchronization, you can reflash the EEPROM with the `ireeprom` command. No man page exists for `ireeprom`.

The following list describes the options for the `ireeprom` command:

`-w`	Write EEPROM from file
`-r`	Read EEPROM into file
`-e`	Erase EEPROM
`-c`	Erase EEPROM video combination
`-v`	Verify contents of EEPROM
`-i`	Print EEPROM information
`-F`	Forcibly load older revision
`-h`	Print this help message
`-p`	<pipenum>: Specify pipe number
`-f`	<datafile>: Specify microcode datafile

You can use the following procedure to restore or upgrade the EEPROM microcode:

1. Use an alternate console to log in to the system as `root`.
2. Change to the following directory:

   `cd /usr/gfx/KONA/bin`

3. Check the EEPROM before you flash it by using the command shown below. If the microcode is good, you should see the following output:

   `./ireeprom -i`

   ```
   ==== Pipe 0 ====
   Manufacturer code:       0x1f
   Device code:             0xc4
   State:                   Good
   Revision:                114|
   timestamp:               Fri May 30 15:46:24 1997
   Code+data length:        0x1caac
   Checksum:                0x009698fd
   ```

   For multipipe systems, use the `-p` option to specify the correct pipe.

4. Overwrite the microcode using the following command. If you use an older or identical version of the microcode to overwrite the EEPROM microcode, you must use the `-F` option.

   ```
   ./ireeprom -w -F -f /usr/gfx/ucode/KONA/tport.bin
   ```

5. Restart the graphics subsystem to download the new microcode:

   ```
   (/usr/gfx/stopgfx; /usr/gfx/startgfx)&
   ```

You can also use `ireeprom` to erase bad combination files in the EEPROM and restore a default combination. If you have downloaded a combination file that causes the monitor to lose sync, the screen will go black and you cannot log in to the graphics terminal. In this case, you cannot use the `setmon` command to restore a good combination because the X Window System server is not running.

1. Use an alternate console to log in to the system as `root`.

2. Change to the following directory:

   ```
   cd /usr/gfx/KONA/bin
   ```

3. Ease the EEPROM video combination:

   ```
   ./ireeprom -c
   ```

4. Restart the graphics subsystem to download the default video combination:

   ```
   (/usr/gfx/stopgfx; /usr/gfx/startgfx)&
   ```

   The system will restart with a default combination of 1280x1024.

Testing the Graphics Subsystem

When you use `irsaudit` without any options, all the graphic subsystem boards are tested: GE, RM, and DG boards.

Note: The standard output shown below was created on a system with two RM7 boards and a DG5-8 board. The test takes several minutes to complete.

Test the complete graphics system by entering the following command:

```
# /usr/diags/IR/bin/irsaudit&
```

The following two files are created at `/var/adm/crash/diags/gfx/IR` that `availmon` uses to track the system:

`irsaudit_number.rslt` and `irsaudit_number.sum`

Testing the GE14 Board

Log in as root and enter the following command to test the GE14 board:

```
# /usr/diags/IR/bin/irsaudit -ge&
```

The following two files are created at `/var/adm/crash/diags/gfx/IR` that `availmon` uses to track the system: `irsaudit_number.rslt` and `irsaudit_number.sum`.

For example: **irsaudit_19970720_173730.rslt**

Starting the X Window Server System

A reboot of the system starts the graphics system, but you can manually start the `Xsgi` server by using the `/usr/gfx/startgfx` command.

Displaying the Graphics Subsystem Information

Use the following command to display graphics subsystem information:

```
# /usr/gfx/gfxinfo
```

Forcing the Graphics Console to an ASCII Terminal

If console control is at the graphics keyboard (`console=g`) and a problem exists in the graphics hardware that prevents the initialization of the graphics subsystem during the power-on boot process, the microcode attempts to transfer the console to the ttyd1 port of the BaseIOG (IO6G) board. This enables a user to retain control of the system even though a problem exists in the graphics subsystem.

A problem in the graphics subsystem prevents access to the system by locking up the graphics console keyboard (the keyboard is inactive) and preventing the microcode from switching console control to the ttyd1 port.

The following procedure forces console control to the ttyd1 port when the console variable is set to `g` (`console=g`).

1. Power off the system.
2. Remove the graphics keyboard cable from the IO6G.
3. Power on the system.

   Late in the power-on boot process, the microcode attempts to initialize the graphics subsystem. Because the microcode does not detect the graphics keyboard, an error message is displayed and console control is automatically switched to the alternate console (the ASCII terminal).
4. From the ASCII terminal, start the system (boot IRIX).
5. After IRIX is running, you can use the irsaudit diagnostic to attempt to isolate the graphics problem.

   Note: Remember to reconnect the graphics keyboard after the problem is fixed.

Configuring Multiple Pipes and X Servers for IRIX 6.4

The IRIX 6.4 release is preconfigured with 4 shared memory input queue (`shmiq`) drivers, 1 input directory containing special character files that point to a keyboard/mouse set, and 3 `qcntl` nodes. This configuration provides support for a maximum of 3 pipes and a single graphics seat.

If your system contains more than 3 pipes, you should configure additional `shmiq` and `qcntl` files to support the number of pipes in your system.

1. Edit the `/var/sysgen/master.d/shmiq` file and change the number of `shmiq` drivers defined in the kernel. Each **shmiq** driver handles input events from user hardware devices through a dual port; in this case, a keyboard and mouse. Typically, you should configure *n*+1 `shmiqs`, where *n* is the number of pipes in your system:

   Change the statement: `#define NSHMIQS 4`

   To: `#define NSHMIQS 9`

2. The `qcntl` nodes are associated with the `shmiq` driver. The `qcntl` device allows the X Server to process character input from the `shmiq` driver. In the `/dev` directory enter the following command lines at the prompt to create additional `qcntl` nodes. Create one `qcntl` node for each pipe in your configuration:

   ```
   mknod qcntl3 c 55 3
   mknod qcntl4 c 55 4
   mknod qcntl5 c 55 5
   mknod qcntl6 c 55 6
   mknod qcntl7 c 55 7
   ```

   Note: You should not configure more than 8 `qcntl` devices (0 – 7)

3. To create additional `/dev/input` directories, plug keyboard/mouse sets into the appropriate BaseIOG ports and reset the system from the PROM monitor prompt.

4. Enter the command `autoconfig -fv` to create a new kernel; then reboot the system.

Configuring Multiple Pipes and X Servers for IRIX 6.5

All IRIX 6.5 systems with graphics capabilities are preconfigured with 9 shared memory input queue (`shmiq`) drivers, 2 input directories containing special character files that point to a keyboard/mouse set, and 8 `qcntl` nodes (maximum number of pipes); therefore, you do not need to configure the multiple pipes and X servers.

IRIX 6.5.2 systems with graphics capabilities are preconfigured with 17 shared memory input queue (`shmiq`) drivers, 2 input directories containing special character files that point to a keyboard/mouse set, and 16 `qcntl` nodes (maximum number of pipes); therefore, you do not need to configure the multiple pipes and X servers.

Appendix

Specifications and Site Planning

The following documents provide specifications for the Origin and Onyx2 systems:

007-3452-003	*Site Preparation for Origin Family, Onyx2, OCTANE and O2*
HR-04122-B	*Preparing for an Installation of a 33- to 256-processor System (Origin Series)*

Part Numbers

Refer to the *Illustrated Parts Catalog* for Origin2000, Onyx2, CRAY Origin2000, and MetaRouter Systems, publication number HMM-408-A, and `http://servinfo.csd.sgi.com` for part number information.

Hardware Diagrams

This section provides hardware diagrams for the following products:

Origin 200 Systems

Figure 80. Origin 200 System Board and CPU Daughter Board

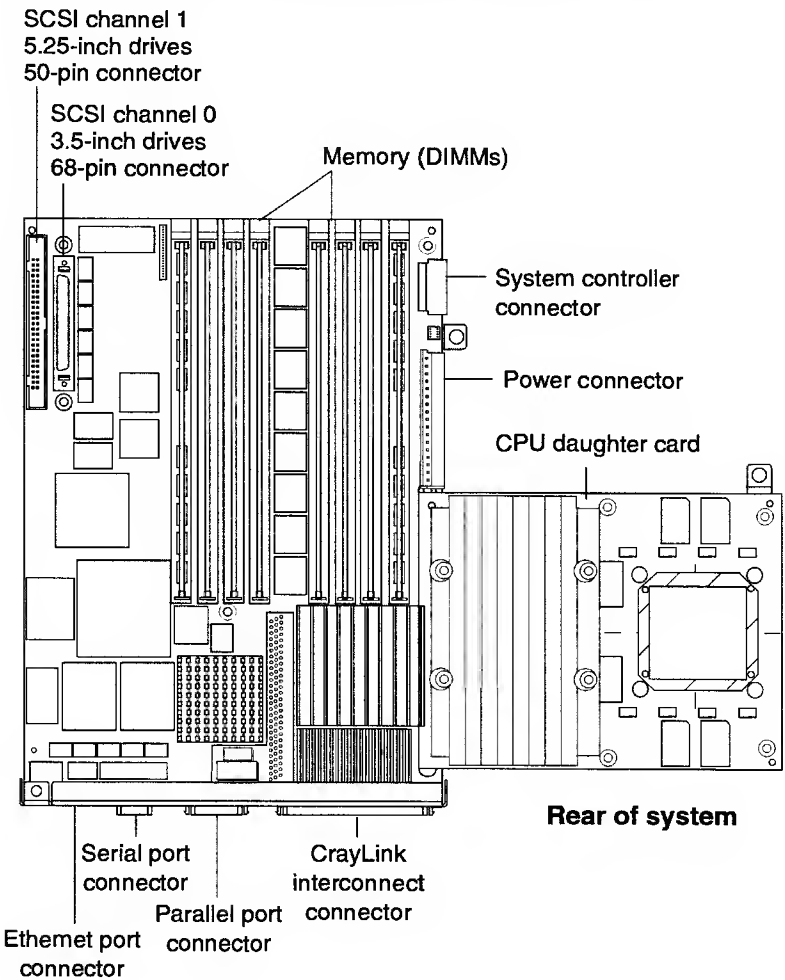


Figure 81. Origin 200 PCI Slots

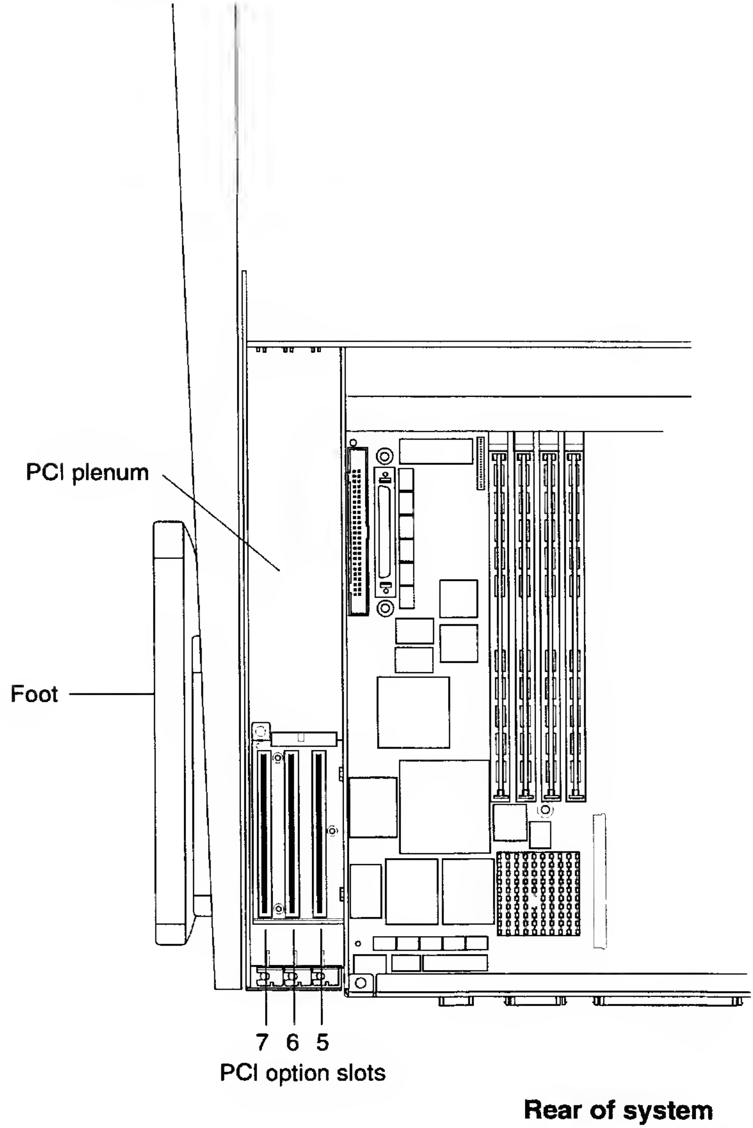
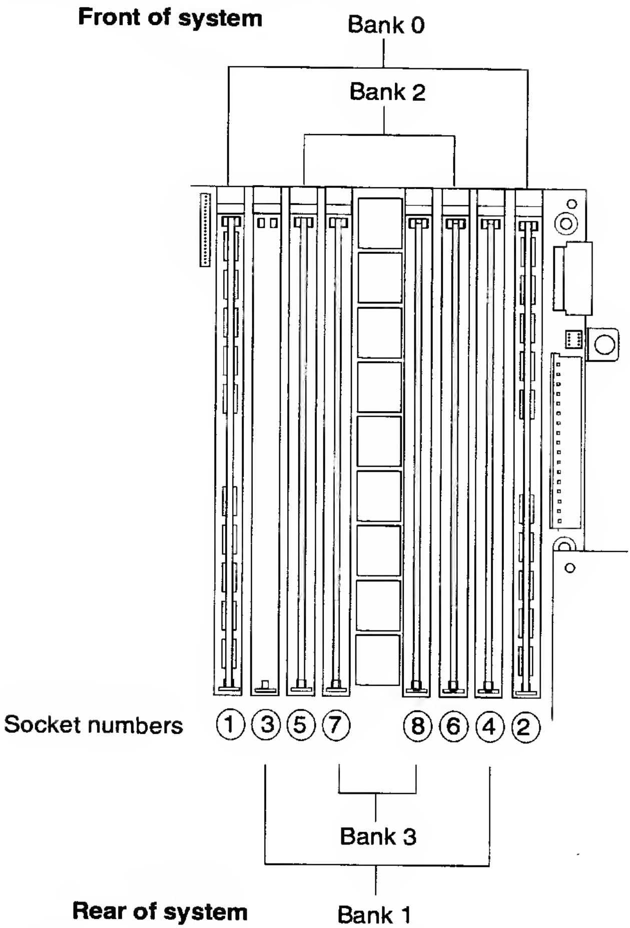


Figure 82. Origin 200 Memory Layout



Note: Socket numbers 1, 3, 5, and 7 are the lower half of the bank.
Socket numbers 2, 4, 6, and 8 are the upper half of the bank.

Origin 2000 Deskside Systems

Figure 83. Origin 2000 Deskside Front View

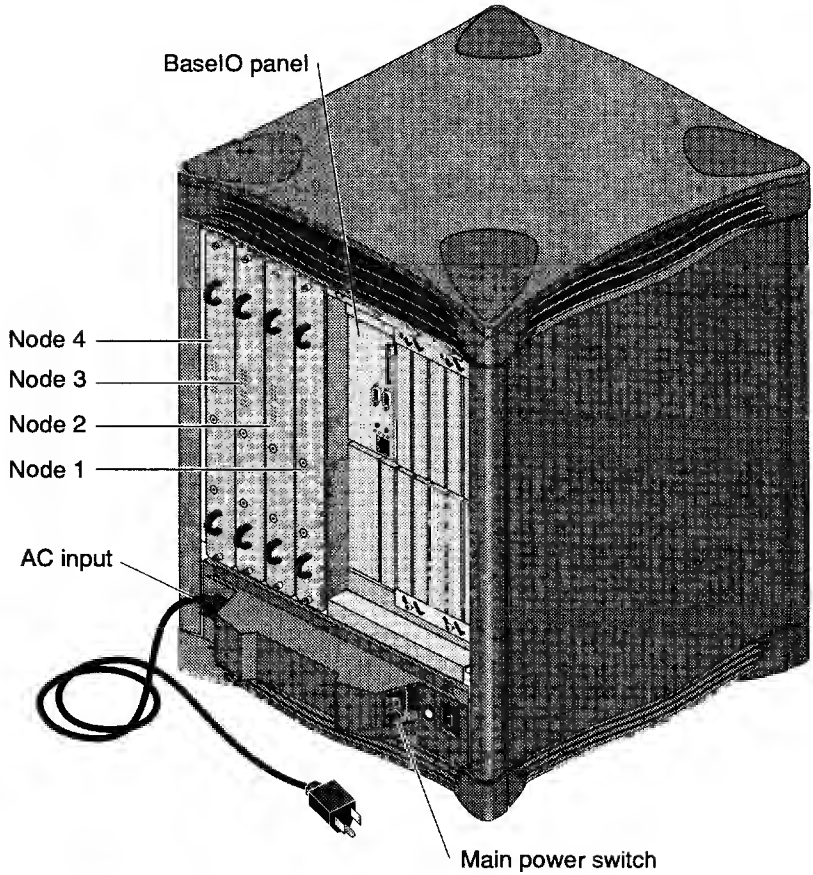
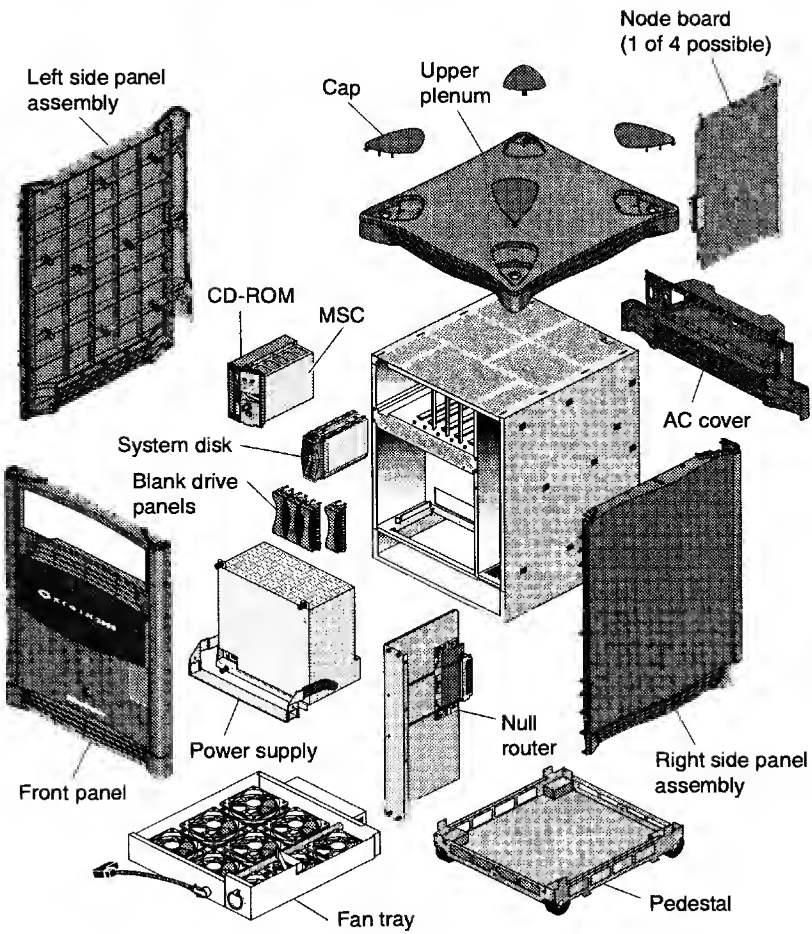


Figure 84. Origin 2000 Deskside Exploded View



Origin 2000 Rack Systems

Figure 85. Origin 2000 Rack System Front View

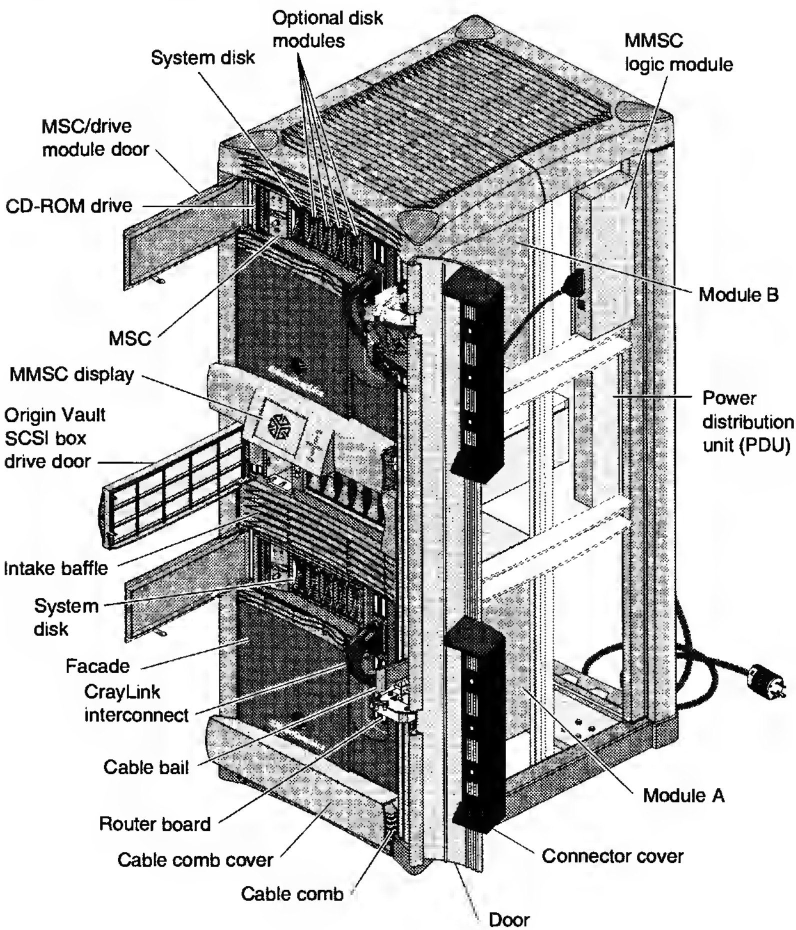


Figure 86. Origin 2000 Rack System Rear View

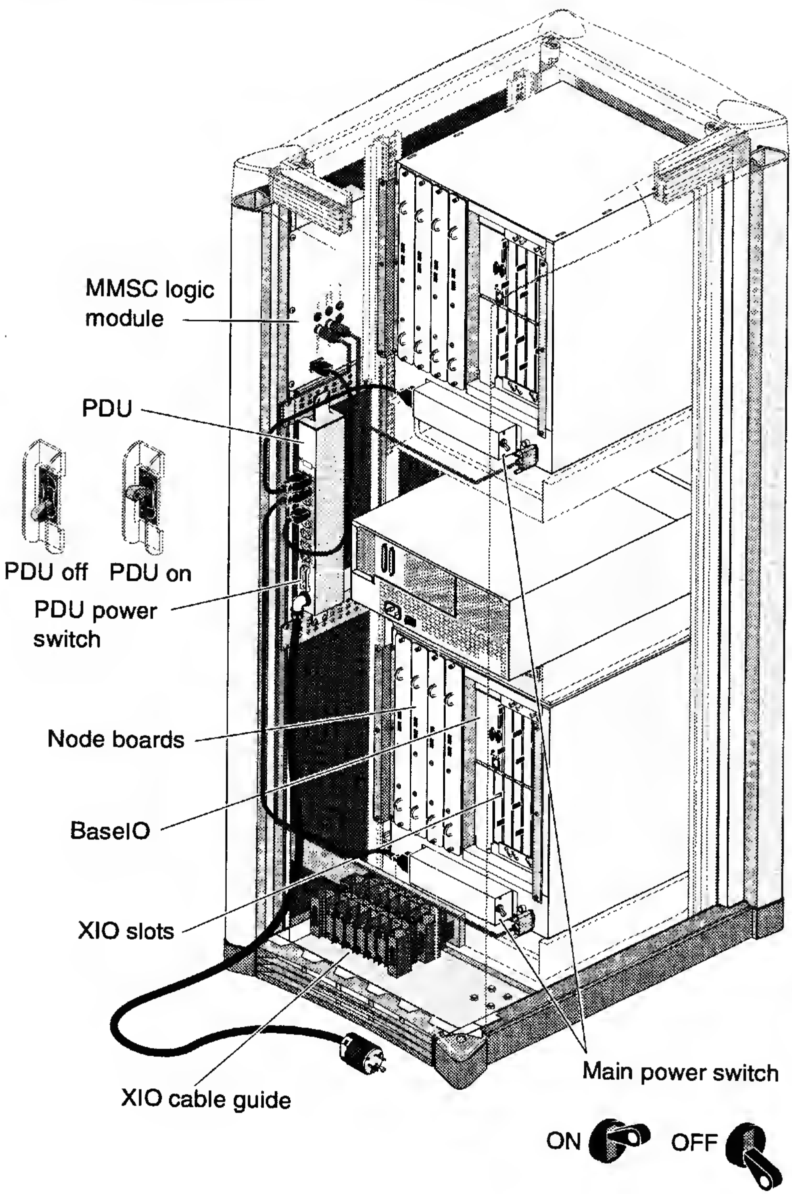


Figure 87. Origin 2000 Rack Exploded View (Part 1 of 2)

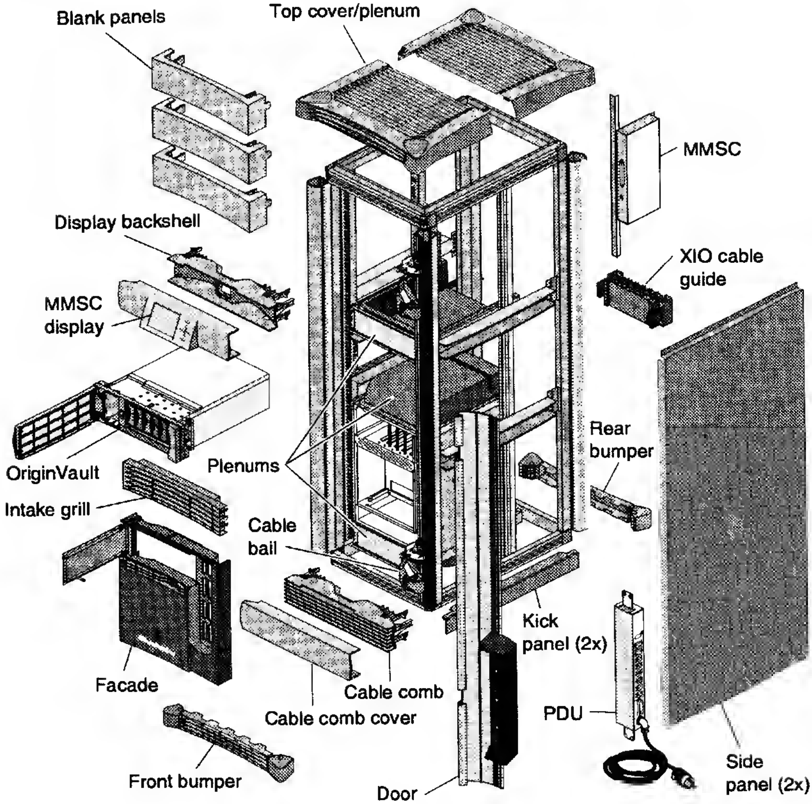


Figure 87. Origin 2000 Rack Exploded View (Part 2 of 2)

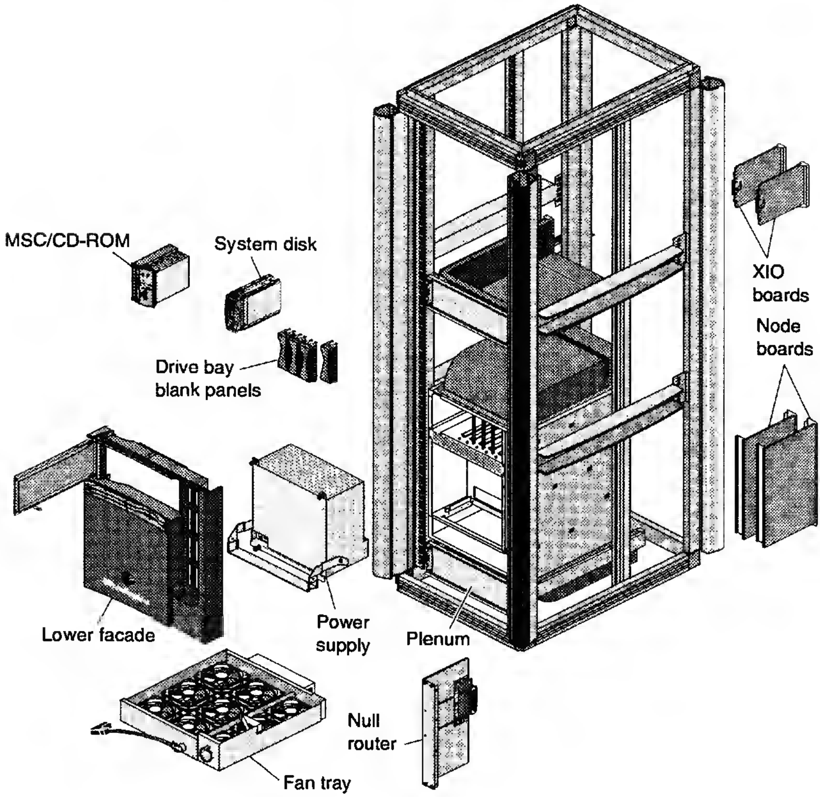


Figure 88. Origin 2000 Module Chassis

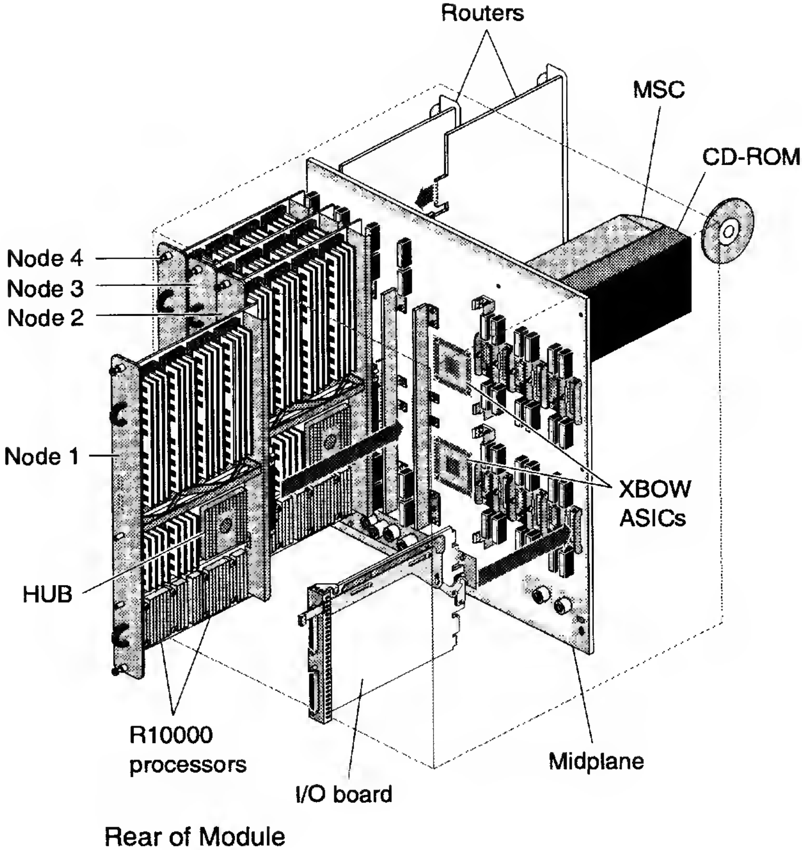


Figure 89. Origin 2000 Midplane Components (Front View)

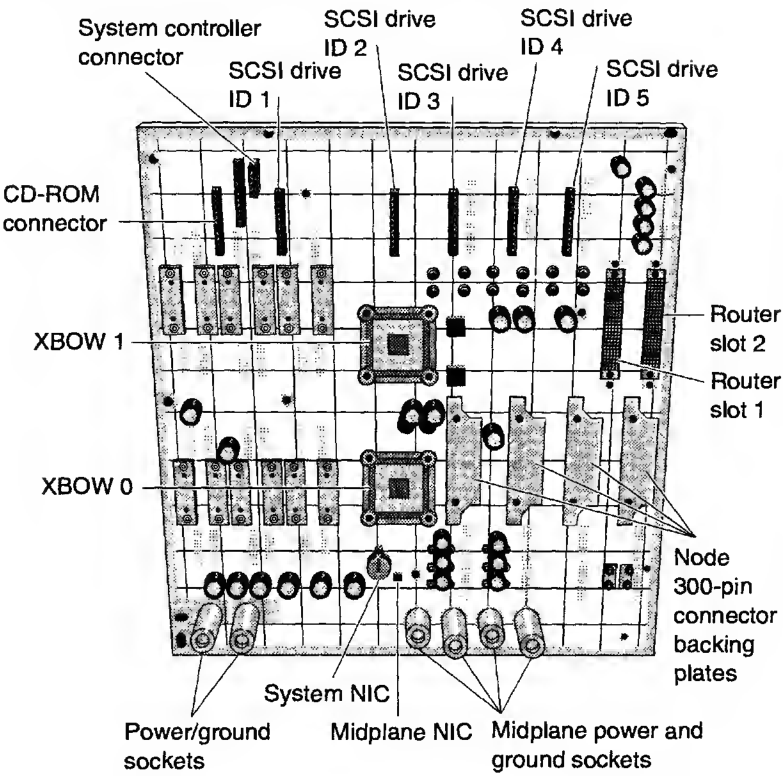


Figure 90. Midplane Components (Rear View)

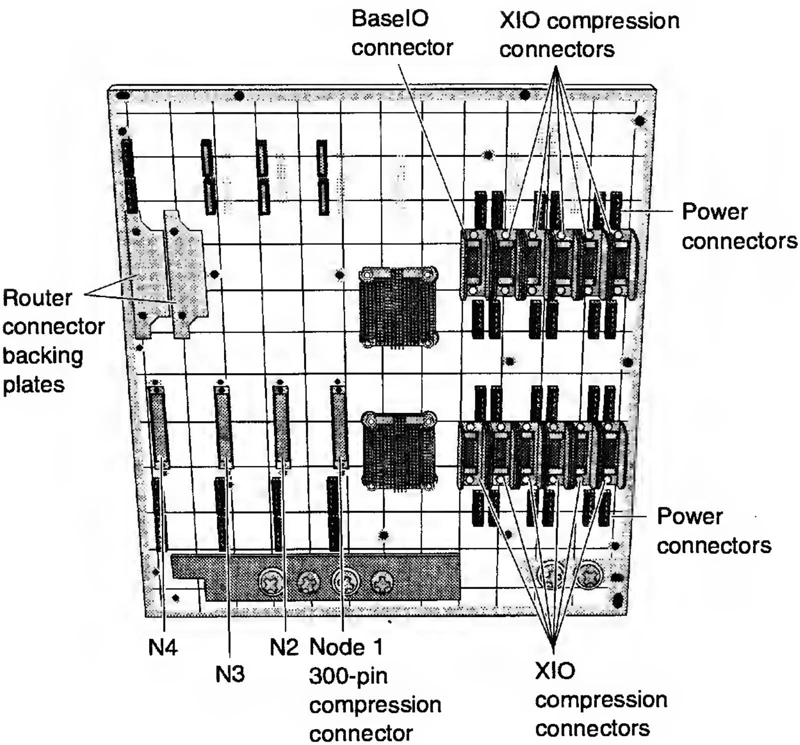


Figure 91. Node Board Bulkhead

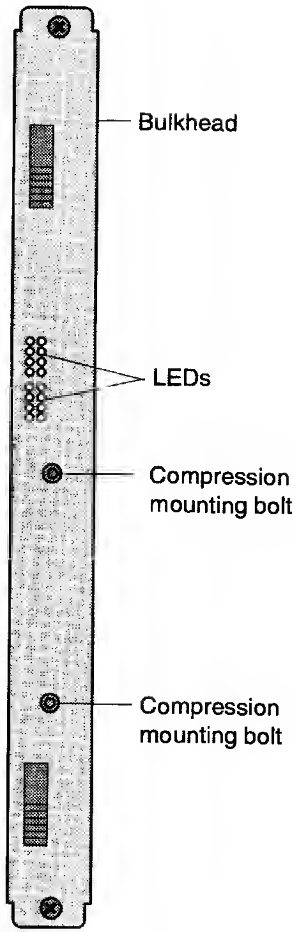


Figure 92. Node Board Components

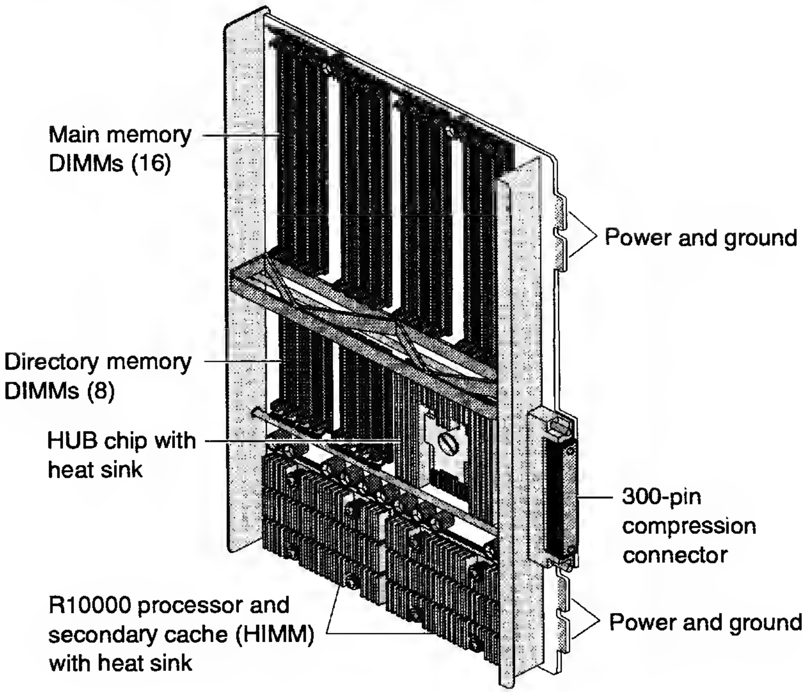


Figure 93. BaseIO Board Connectors

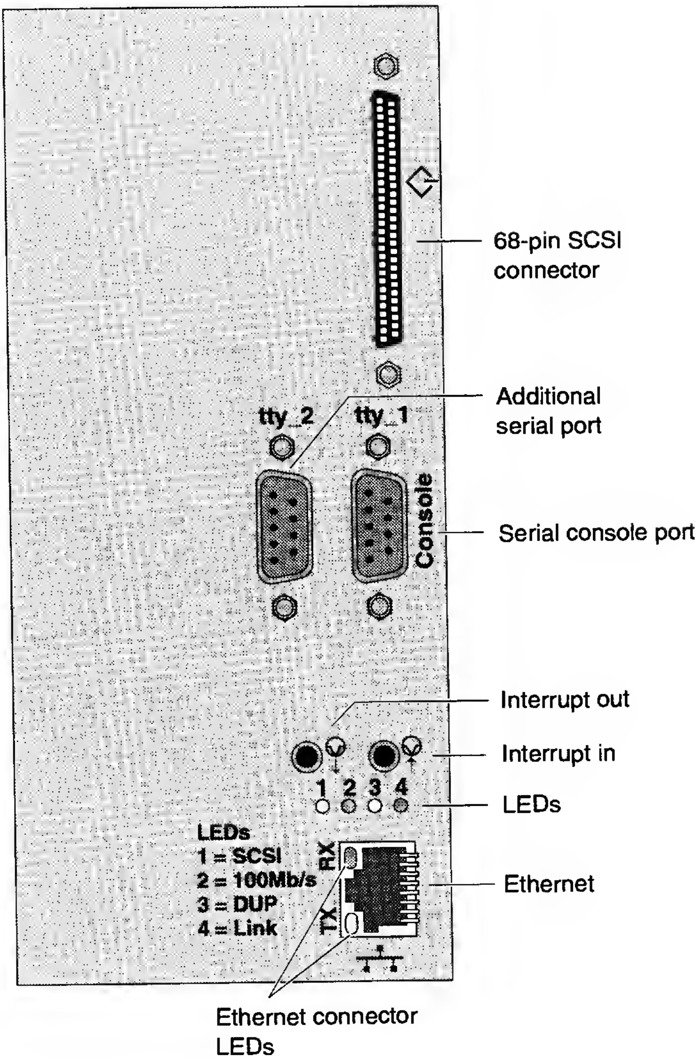


Figure 94. MSC Display

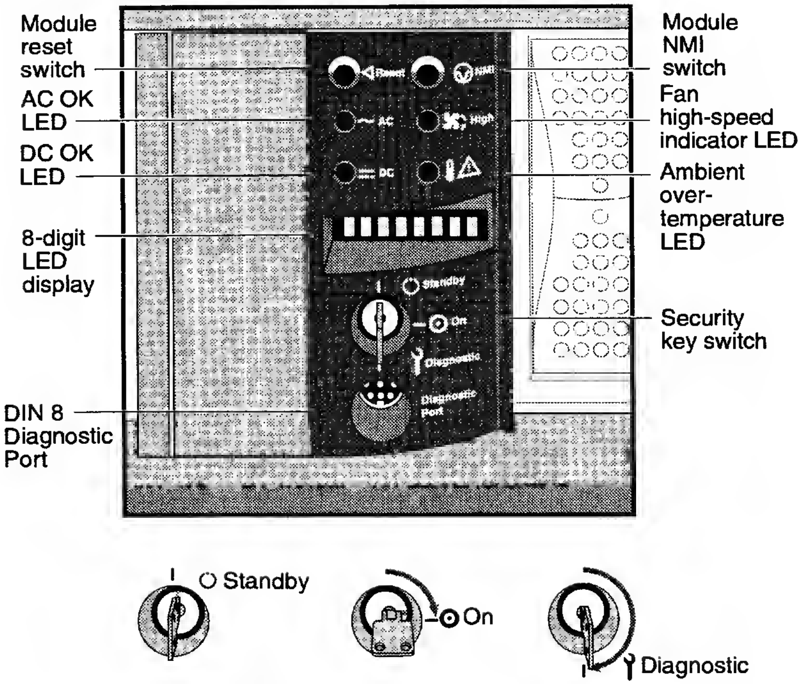
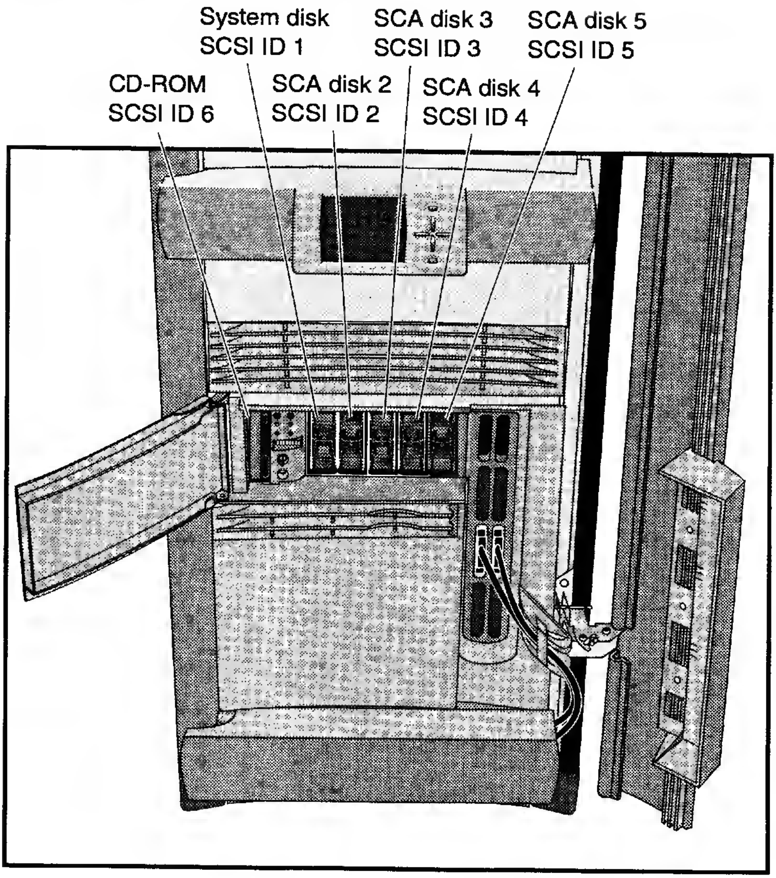


Figure 95. Origin 2000 and Onyx2 SCSI ID Locations



Onyx2 Deskside Systems

Figure 96. Onyx2 Deskside Systems Rear View

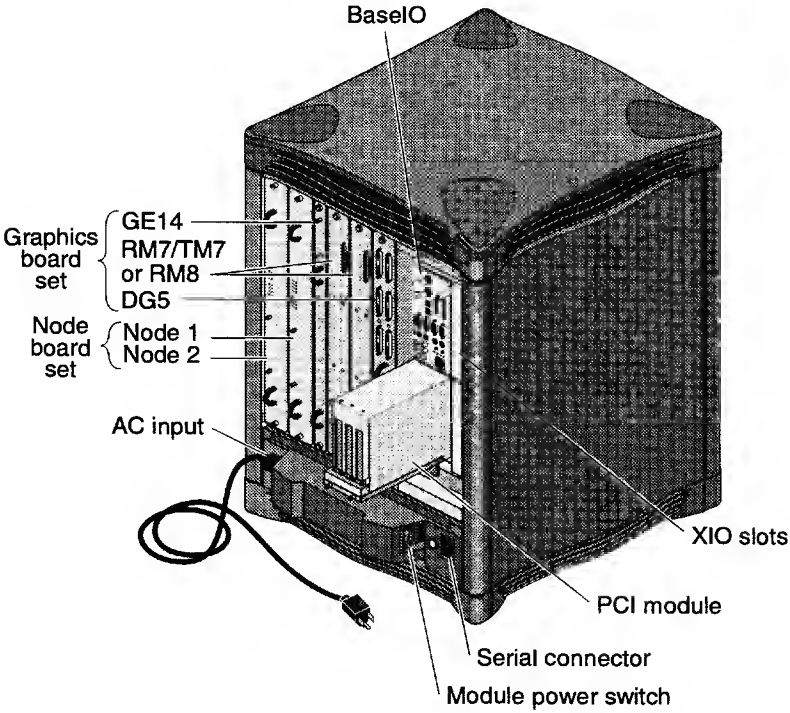


Figure 97. Onyx2 Deskside Exploded View

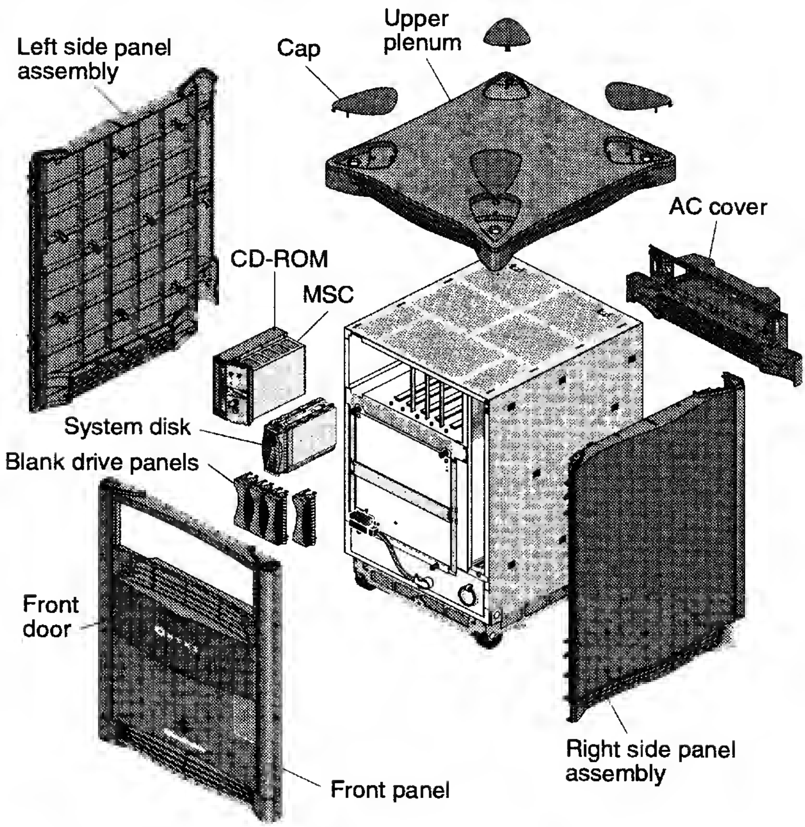
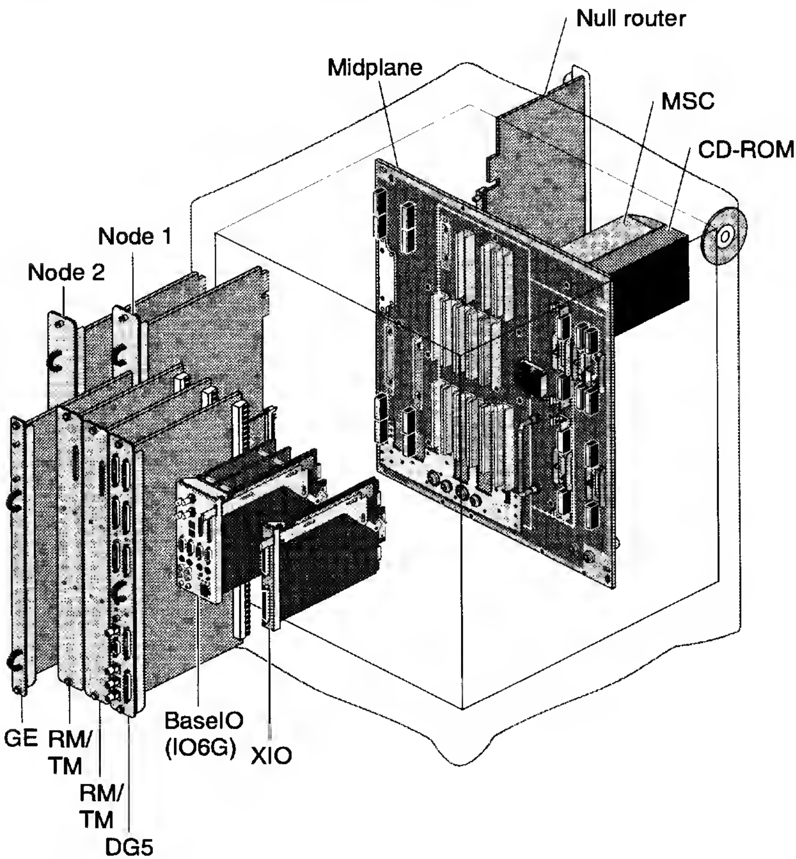


Figure 98. Onyx2 Deskside System Boards and Midplane Exploded View



Onyx2 Rack Systems

Figure 99. Onyx2 Graphics Boards

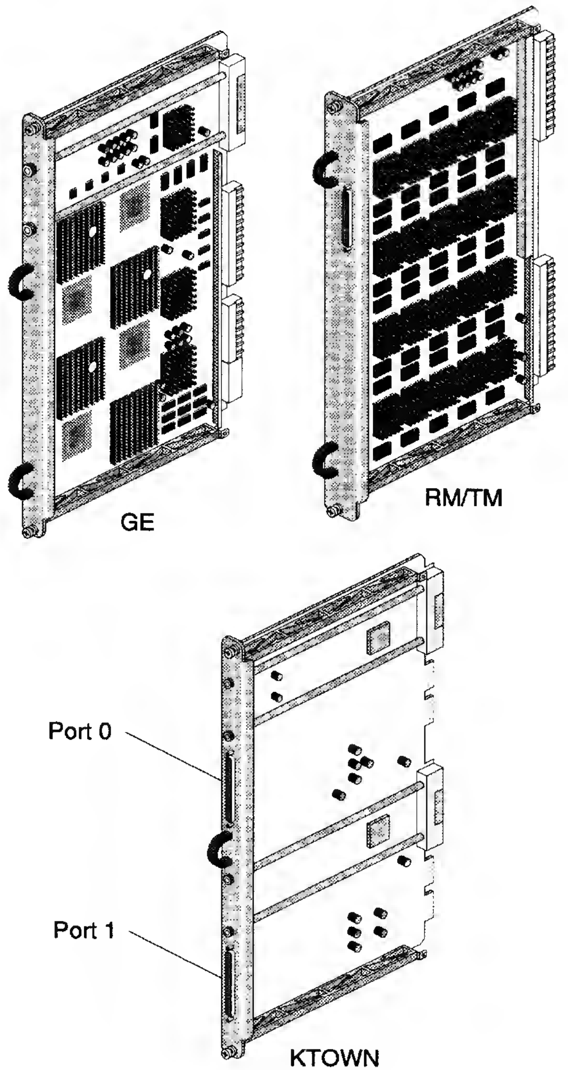


Figure 100. Onyx2 Rack System Front View

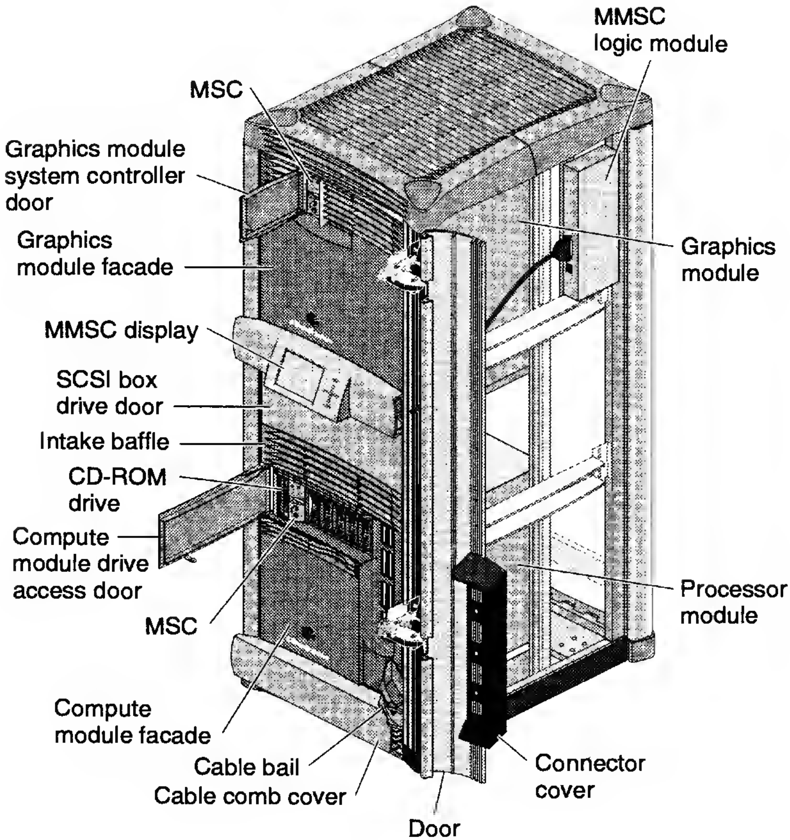


Figure 101. Onyx2 Rack System Rear View

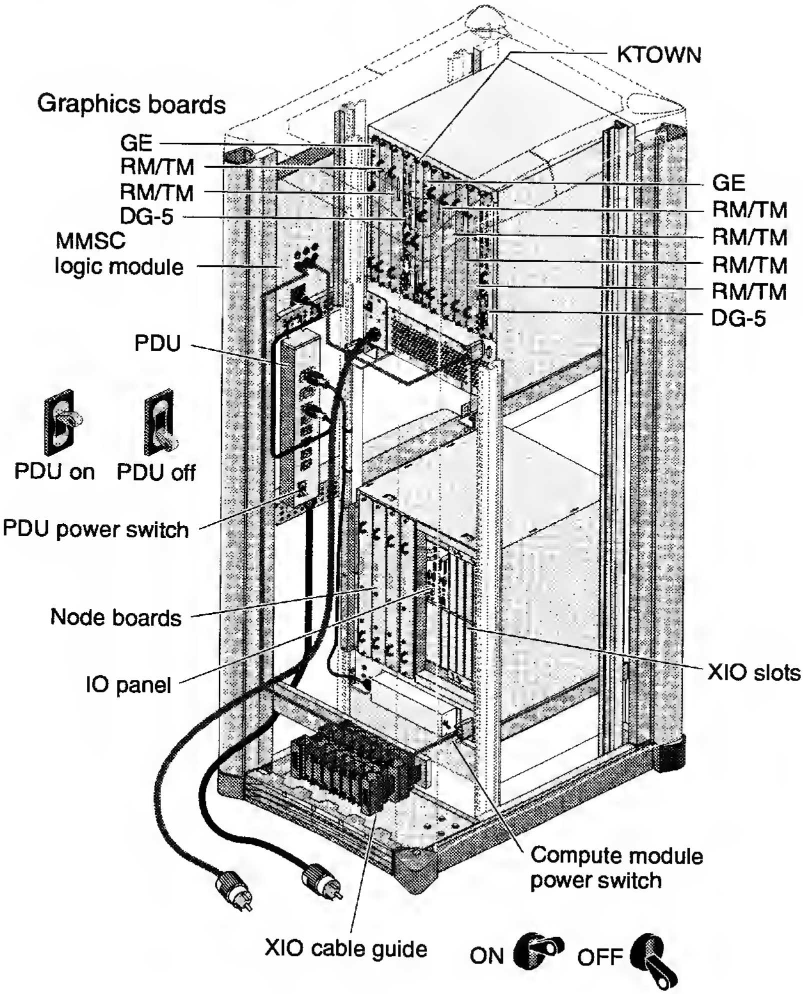


Figure 102. Onyx2 Rack Exploded View

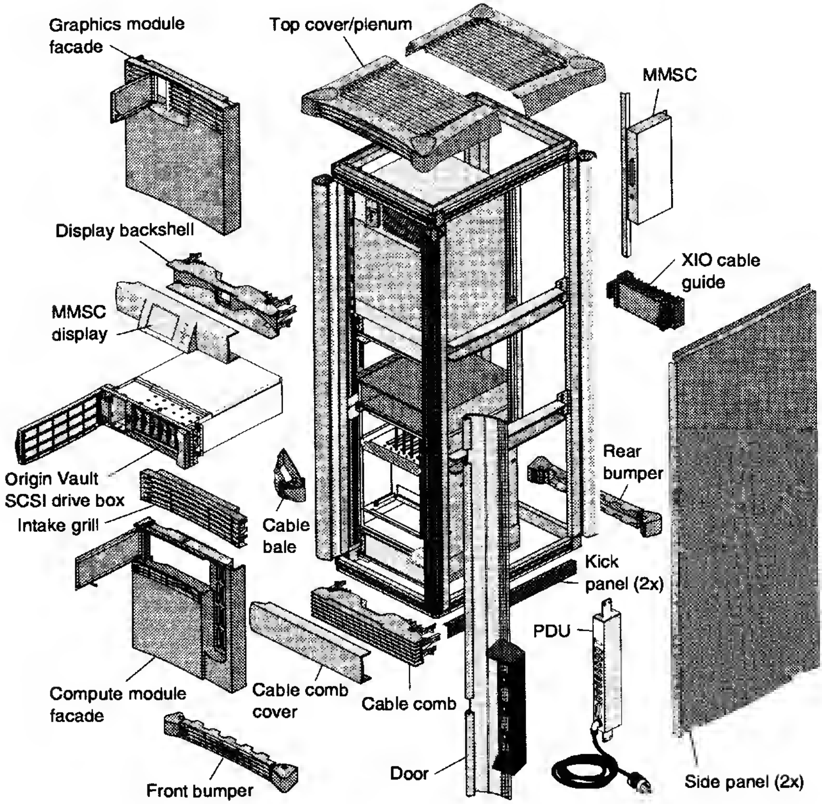
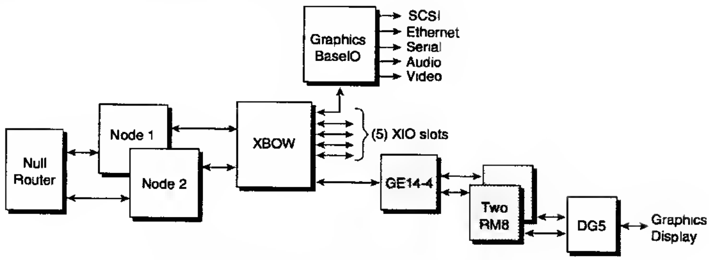
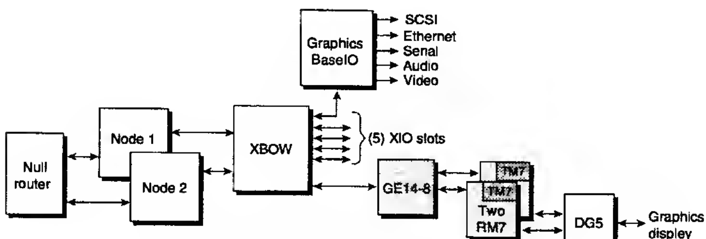


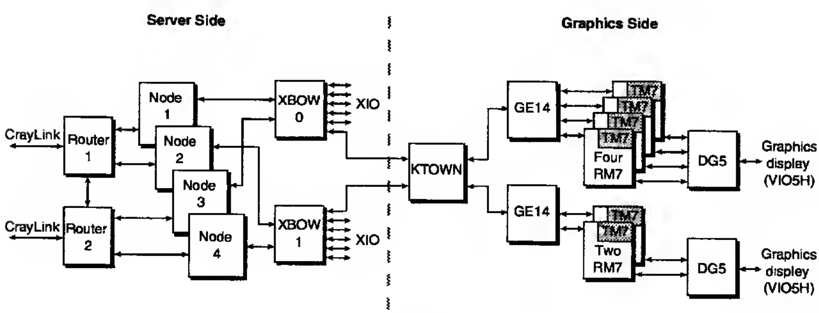
Figure 103. Onyx2 Systems Block Diagrams



Onyx2 Reality deskside block diagram



Onyx2 InfiniteReality deskside block diagram



Onyx2 rack block diagram

Figure 104. Onyx2 Midplane Front View

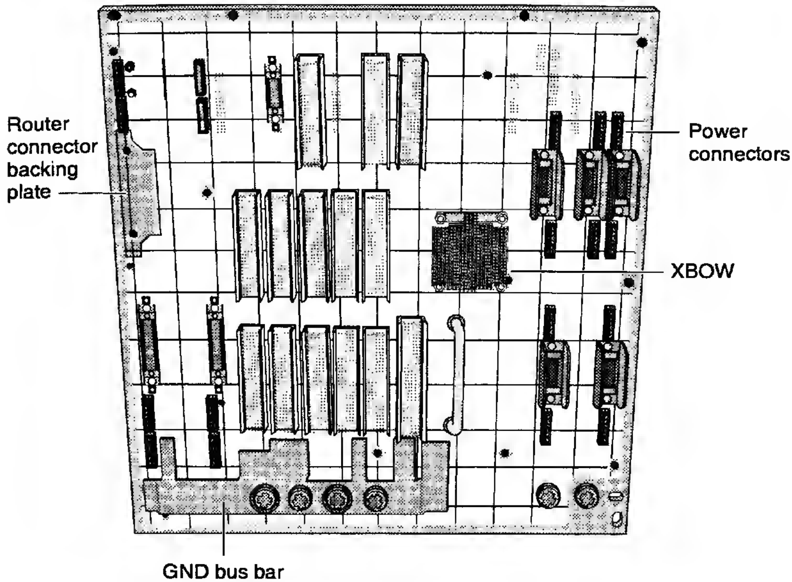
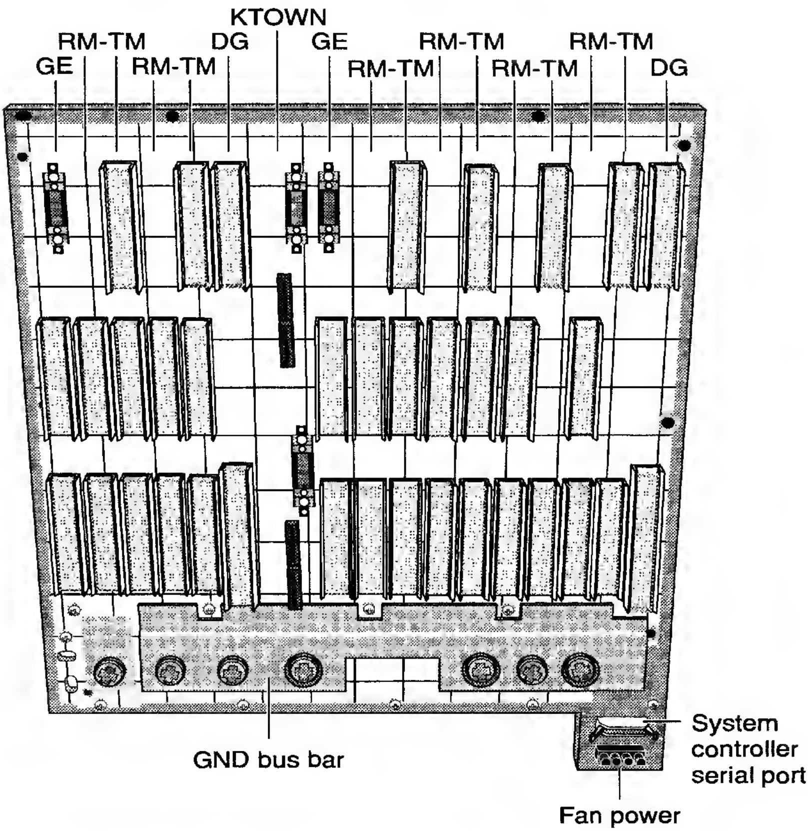


Figure 105. Onyx2 Midplane Rear View



MetaRouter

Figure 106. MetaRouter Cube Designations

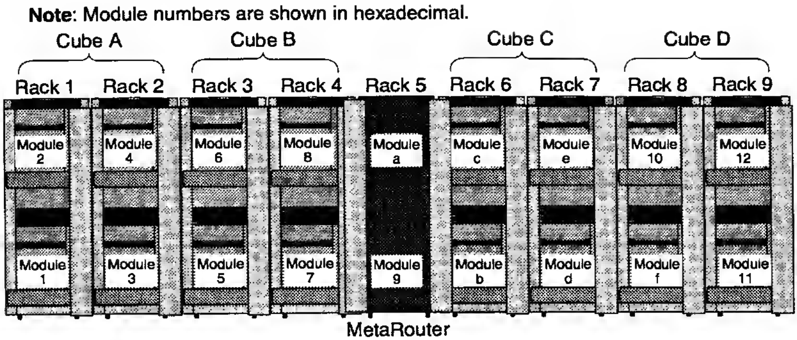


Figure 107. MetaRouter Midplane

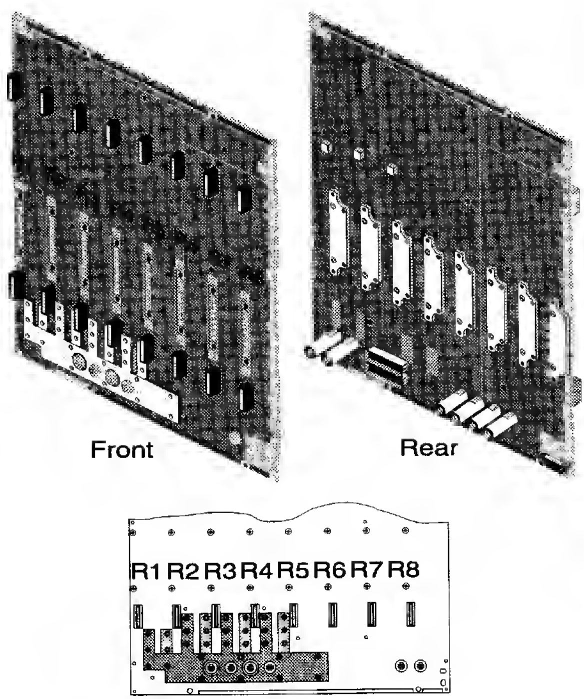
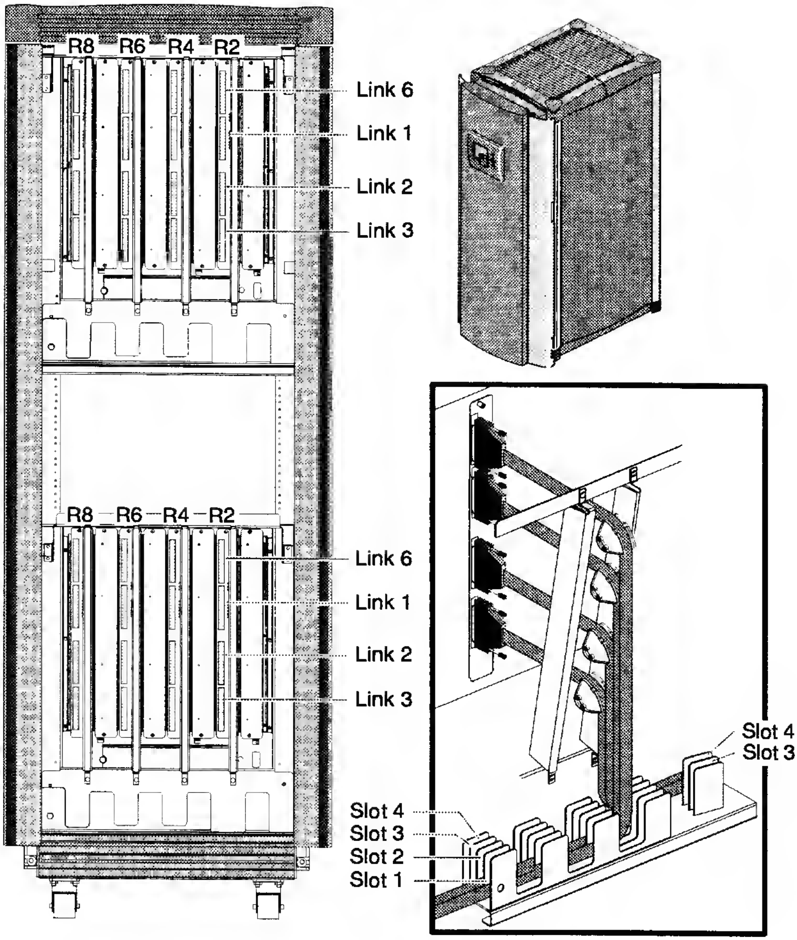


Figure 108. MetaRouter Slot Numbering



Index

Numerics

A

B

C

D

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Reader Comment Form

Harware Quick-reference Booklet
(Origin™ and Onyx2™ Series Systems **HMQ-380-C**

Your feedback on this publication will help us provide better documentation in the future. Please take a moment to answer the few questions below.

For what purpose did you primarily use this document?

_______Troubleshooting
_______Tutorial or introduction
_______Reference information
_______Classroom use
_______Other - please explain

Using a scale from 1 (poor) to 10 (excellent), please rate this document on the following criteria and explain your ratings:

_______Accuracy
_______Organization
_______Readability
_______Physical qualities (binding, printing, page layout)
_______Amount of diagrams and photos
_______Quality of diagrams and photos

Completeness (Check one and explain your answer)

_______Too much information
_______Too little information
_______Correct amount

You may write additional comments in the space below. Mail your comments to the address below, fax them to us at +1 715 726 4353, or e-mail them to us at *spt@sgi.com*. When possible, please give specific page and paragraph references. We will respond to your comments in writing within 48 hours.

____
____
____
____

sgi™

Attn: Service Publications and Training
890 Industrial Boulevard
P.O. Box 4000
Chippewa Falls, WI 54729-0078
USA