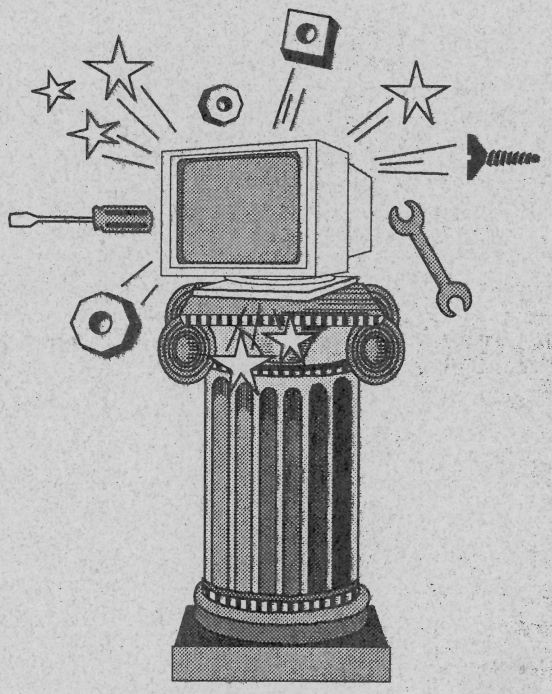
SARATE TABCUM JR.

SiliconGraphics
Computer Systems

Origin™ Maintenance Course

Student Workbook



Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

RESTRICTION ON USE:

This document is protected by copyright and contains information proprietary to Silicon Graphics, Inc. Any copying, adaptation, distribution, public performance, or public display of this document without the express written consent of Silicon Graphics, Inc., is strictly prohibited. The receipt or possession of this document does not convey the rights to reproduce or distribute its contents, or to manufacture, use, or sell anything that it may describe, in whole or in part, without the specific written consent of Silicon Graphics, Inc.

Copyright © 1997 Silicon Graphics, Inc. All rights reserved.

U.S. GOVERNMENT RESTRICTED RIGHTS LEGEND:

Use, duplication, or disclosure of the data and information contained in this document by the Government is subject to restrictions as set forth in FAR 52.227-19(c)(2) or subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and/or in similar or successor clauses in the FAR, or the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., P.O. Box 7311, Mountain View, CA 94039-7311.

PART NUMBER:

LBT139-1.0-MR-6.4-S-SD-W

TRADEMARKS:

SGI Trademarks

Silicon Graphics, the Silicon Graphics logo, CHALLENGE, and Onyx are registered trademarks, and Origin2000, Origin200, IRIX, SSMP, CrayLink, Xpress Link, Onyx2, InfiniteReality2, Origin, INDY, Cray, and PatchFinder are trademarks of Silicon Graphics, Inc. R10000 is a trademark of MIPs.

Other Trademarks

Ethernet is a registered trademark of Xerox Corp. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. Sony is a registered trademark of Sony Corporation

The contents of this publication are subject to change without notice.

CONTRIBUTIONS:

Course development by Sam Sengupta, Tom Hannahs, Chris Pionkowski, Paul Pedersen, Richard Houston, and Matt Marchese.

Technical review by Philip Montalban, Henry Ortiz, and William Yergin

Illustrations by Sam Sengupta, Tom Hannahs, Le Grand Varner, Paul Pedersen, Richard Houston, and various illustrators of the SGI Technical Publications Group.

Editing and Production by Tom Dye, Julie Knoeller, and Cindi Leiser.

Contents: Student Workbook

Module 1: Introduction

1-2 FOIL: Welcome 1-3

1-3 FOIL: Description 1-5

1-4 FOIL: Equipment: 1-7

1-5 FOIL: Scope 1-9

1-6 FOIL: Prerequisites 1-11

1-7 FOIL: References 1-13

1-8 FOIL: Course Objectives 1-15

1-9 FOIL: Performance 1-17

Course Outline 1-19

Daily Schedule: Week 1 1-28

Daily Schedule: Week 2 1-29

Module 2: Product Line

2-2 FOIL: Module Objectives 2-3

2-3 FOIL: Introduction 2-5

2-4 FOIL: Product Description 2-7

2-5 FOIL: Enclosures 2-9

2-6 FOIL: Product Line 2-11

2-7 FOIL: Origin200 2-13

2-8 FOIL: Origin200 (continued) 2-15

2-9 FOIL: Origin200 (continued) 2-17

2-10 FOIL: Origin2000 Deskside 2-19

2-11 FOIL: Origin2000 Rack 2-21

2-12 FOIL: CRAY Origin2000 System 2-23

2-13 FOIL: Onyx2 Deskside 2-25

2-14 FOIL: Onyx2 Deskside (continued) 2-27

2-15 FOIL: Onyx2 Rack Systems 2-29

2-16 FOIL: Origin Rack Systems (continued) 2-31

2-17 FOIL: Origin Vault 2-33

Module 3: System Architecture

3-2 FOIL: Module Objectives 3-3

3-3 FOIL: Module Topics 3-5

3-4 FOIL: CHALLENGE vs. Origin2000 Systems 3-7

3-5 FOIL: CHALLENGE vs. Origin2000 Systems (continued) 3-9

3-6 FOIL: System Features 3-11

3-7 FOIL: System Features (continued) 3-13

3-8 FOIL: System Components 3-15

3-9 FOIL: System Components (continued) 3-17

3-10 FOIL: System Components (continued) 3-19

3-11 FOIL: System Components (continued) 3-21

3-12 FOIL: Key ASICs 3-23

3-13 FOIL: Crossbar Switches 3-25

3-14 FOIL: Number In a Can (NIC) 3-27

3-15 FOIL: Origin2000 Overview 3-29

3-16 FOIL: Node Board 3-31

3-17 FOIL: Node Board (continued) 3-33

3-18 FOIL: Midplane 3-35

3-19 FOIL: Midplane (continued) 3-37

3-20 FOIL: XIO Slots 3-39

3-21 FOIL: Router 3-41

3-22 FOIL: Router (continued) 3-43

3-23 FOIL: XIO Boards 3-45

3-24 FOIL: Block Diagram 3-47

3-25 FOIL: Connections 3-49

3-26 FOIL: Indirect Connections 3-51

3-27 FOIL: Data Integrity 3-53

3-28 FOIL: Data Integrity (continued) 3-55

3-29 FOIL: Memory Organization 3-57

3-30 FOIL: Cache Coherence 3-59

Module 4: System Components

4-2 FOIL: Module Objectives 4-3

4-3 FOIL: Module Topics 4-5

4-4 FOIL: Origin2000 Major Components 4-7

4-5 FOIL: SN0 Component Introduction 4-9

4-6 FOIL: Node Board 4-11

4-7 FOIL: Node Board Components 4-13

4-8 FOIL: Node Board Components (continued) 4-15

4-9 FOIL: HUB ASIC 4-17

4-10 FOIL: Processor Interface (PI) 4-19

4-11 FOIL: Network Interface (NI) 4-21

4-12 FOIL: XIO Interface (II) 4-23

4-13 FOIL: Memory/Directory Interface (MD) 4-25

4-14 FOIL: Memory Block Diagram 4-27

4-15 FOIL: Memory 4-29

4-16 FOIL: Memory Layout 4-31

4-17 FOIL: Memory Layout (continued) 4-33

4-18 FOIL: Memory DIMMs 4-35

4-19 FOIL: DIMM Location Chart 4-37

4-20 FOIL: DIMM Combination Examples 4-39

4-21 FOIL: DIMM Combination Examples (continued) 4-41

4-22 FOIL: DIMM Combination Examples (continued) 4-43

4-23 FOIL: Router Board Types 4-45

4-24 FOIL: Null Router 4-47

4-25 FOIL: Null Router (continued) 4-49

4-26 FOIL: Standard Router 4-51

4-27 FOIL: Standard Router (continued) 4-53

4-28 FOIL: Star Router 4-55

4-29 FOIL: Star Router (continued) 4-57

4-30 FOIL: Cray Router 4-59

4-31 FOIL: Midplane 4-61

4-32 FOIL: Board Numbering 4-63

4-33 FOIL: Power and Cooling 4-65

4-34 FOIL: Power and Cooling (continued) 4-67

Module 5: Hardware Configurations

5-2 FOIL: Objectives 5-3

5-3 FOIL: Module Topics 5-5

5-4 FOIL: CrayLink Interconnect 5-7

5-5 FOIL: Interconnect Fabric 5-9

5-6 FOIL: Midplane-to-Router Connections 5-11

5-7 FOIL: Configurations 5-13

5-8 FOIL: Configurations (continued) 5-15

5-9 FOIL: Configurations (continued) 5-17

5-10 FOIL: Configurations (continued) 5-19

5-11 FOIL: Configurations (continued) 5-21

5-12 FOIL: Configuration Table 5-23

5-13 FOIL: Configuration Table (continued) 5-25

Module 6: ESD: Board Handling & Removal

6-2 FOIL: Objectives 6-3

6-3 FOIL: Module Topics 6-5

6-4 FOIL: ESD 6-7

6-5 FOIL: General Parts Removal 6-9

6-6 FOIL: Lab Project 1: ESD: Board Handling & Removal 6-11

Module 7: CrayLink Cabling

7-2 FOIL: Module Objectives 7-3

7-3 FOIL: Rack Numbers 7-5

7-4 FOIL: Module Numbers 7-7

7-5 FOIL: Module Block Diagram (classroom systems) 7-9

7-6 FOIL: CrayLink Considerations 7-11

7-7 FOIL: Router LEDs 7-13

7-8 FOIL: Basic Classroom Rack Configuration 7-15

7-9 FOIL: Classroom Racks with XpressLink 7-17

7-10 FOIL: Route Table 7-19

7-11 FOIL: Block Diagram: 24 CPUs 7-21

7-12 FOIL: Hardware Inventory 7-23

7-13 FOIL: Configuration Data 7-25

7-14 FOIL: Useful Commands 7-27

7-15 FOIL: Lab Project 2: CrayLink Cabling 7-29

Module 8: PROM Monitor

8-2 FOIL: Module Objectives 8-3

8-3 FOIL: System Software 8-5

8-4 FOIL: IP27 Microcode 8-7

8-5 FOIL: IO6 Microcode 8-9

8-6 FOIL: PROM Monitor 8-11

8-7 FOIL: Command Monitor Commands 8-13

8-8 FOIL: PROM Variable Descriptions. 8-15

8-9 FOIL: Environmental Parameters 8-17

8-10 FOIL: Device Naming: Old Method 8-19

8-11 FOIL: Device Naming: New Method 8-21

8-12 FOIL: Lab Project 3: PROM Monitor 8-23

Module 9: System Controllers

9-2 FOIL: Module Objectives 9-3

9-3 FOIL: System Controllers 9-5

9-4 FOIL: MSC Features and Functions 9-7

9-5 FOIL: MSC Front Panel 9-9

9-6 FOIL: MSC Messages 9-11

9-7 FOIL: Origin MSC Interconnect Diagram 9-13

9-8 FOIL: MSC Power-on Sequence 9-15

9-9 FOIL: Origin2000 Voltages 9-17

9-10 FOIL: LEDs 9-19

9-11 FOIL: MSC Hardware Debug Switches 9-21

9-12 FOIL: Debug Switch Meaning 9-23

9-13 FOIL: Debug Switch Meaning (continued) 9-25

9-14 FOIL: The Debug Switches 9-27

9-15 FOIL: Physical Switches vs. Hardware Switches 9-29

9-16 FOIL: Toggle a Hardware Debug Switch 9-31

9-17 FOIL: MMSC Features 9-33

9-18 FOIL: MMSC Cable Configuration 9-35

9-19 FOIL: MMSC-to-MMSC 9-37

9-20 FOIL: MMSC Display Panel 9-39

9-21 FOIL: MMSC Special Screens 9-41

9-22 FOIL: ASCII Terminal Attributes 9-43

9-23 FOIL: Lab Project 4: System Controllers - MSC/MMSC 9-45

Module 10: Power-on Diagnostics

10-2 FOIL: Module Objectives 10-3

10-3 FOIL: Power-on Boot Process 10-5

10-4 FOIL: Firmware Flow Overview 10-7

10-5 FOIL: Power-on Boot Process 10-9

10-6 FOIL: Power-On Boot Process (continued) 10-11

10-7 FOIL: Power-on Boot Process (continued) 10-13

10-8 FOIL: Power-on Boot Process (continued) 10-15

10-9 FOIL: Power-on Boot Process (continued) 10-17

10-10 FOIL: Power-on Boot Process (continued) 10-19

10-11 FOIL: Node Board LEDs 10-21

10-12 FOIL: Node LED Error Codes 10-23

10-13 FOIL: Console Base IO Selection Algorithm 10-25

10-14 FOIL: Global Master Selection Algorithm 10-27

10-15 FOIL: Power-on Boot Analysis 10-29

10-16 FOIL: Power-on Boot Analysis (continued) 10-31

10-17 FOIL: Power-on Boot Analysis (continued) 10-33

10-18 FOIL: Power-on Boot Analysis (continued) 10-35

10-19 FOIL: Power-on Boot Analysis (continued) 10-37

10-20 FOIL: Power-on Boot Analysis (continued) 10-39

10-21 FOIL: Power-on Boot Analysis (continued) 10-41

10-22 FOIL: Power-on Boot Analysis (continued) 10-43

10-23 FOIL: Power-on Boot Analysis (continued) 10-45

10-24 FOIL: Power-on Boot Analysis (continued) 10-47

10-25 FOIL: POD Command Interpreter 10-49

10-26 FOIL: Entering POD 10-51

10-27 FOIL: POD Prompt 10-53

10-28 FOIL: Why POD 10-55

10-29 FOIL: Lab Project 5: POD 10-57

10-30 FOIL: Individual CPU Power-on Dumps. 10-59

Module 11: Fault Isolation: Origin2000

11-2 FOIL: Module Objectives 11-3

11-3 FOIL: Fault Isolation: Origin2000 11-5

Module 12: Graphics Introduction

12-2 FOIL: Module Objectives 12-3

12-3 FOIL: GFX Terminology 12-5

12-4 FOIL: GFX Terminology (continued) 12-7

12-5 FOIL: GFX Terminology (continued) 12-9

12-6 FOIL: Primary Colors 12-11

12-7 FOIL: Screen Pixel 12-13

12-8 FOIL: Secondary Colors 12-15

12-9 FOIL: Colors Exercise 12-17

12-10 FOIL: Screen Coordinates 12-19

12-11 FOIL: Pixel vs. FrameBuffer 12-21

12-12 FOIL: Image Buffers 12-23

12-13 FOIL: Motion Exercise 12-25

12-14 FOIL: 2D Drawings 12-27

12-15 FOIL: 2D Exercise 12-29

12-16 FOIL: 3-Dimensional Space 12-31

12-17 FOIL: Orthographic Space 12-33

12-18 FOIL: Perspective Space 12-35

12-19 FOIL: World Space Exercise 12-37

12-20 FOIL: Z Buffer 12-39

12-21 FOIL: Z Buffer Exercise 12-41

12-22 FOIL: Lighting 12-43

12-23 FOIL: Lighting Exercise 12-45

Module 13: The Onyx2 System

13-2 FOIL: Module Objectives 13-3

13-3 FOIL: Onyx2 Products 13-5

13-4 FOIL: Onyx2 Reality 13-7

13-5 FOIL: Onyx2 InfiniteReality Deskside 13-9

13-6 FOIL: Onyx2 InfiniteReality Rack 13-11

13-7 FOIL: GFX Components 13-13

13-8 FOIL: Deskside Slot Positions 13-15

13-9 FOIL: Rack Slot Positions 13-17

13-10 FOIL: Crosstown Board (Xtown) 13-19

13-11 FOIL: KonaTown Board (Ktown) 13-21

13-12 FOIL: Single-rack Configuration 13-23

13-13 FOIL: Dual-rack Configuration - Maximum CPUs 13-25

13-14 FOIL: Dual-rack Configuration - Three 4-RM Pipes 13-27

13-15 FOIL: Dual-rack Configuration - Maximum Pipes 13-29

13-16 FOIL: Crosstown Cable 13-31

13-17 FOIL: Onyx2 Board Set 13-33

13-18 FOIL: Onyx2 Board Illustrations 13-35

13-19 FOIL: IO6G 13-37

13-20 FOIL: IO6G Connections 13-39

13-21 FOIL: GFX Monitor Connections 13-41

13-22 FOIL: DG5-8 Connections 13-43

13-23 FOIL: DG5-GVO Connections 13-45

13-24 FOIL: Multi-seat Onyx2 Configurations 13-47

13-25 FOIL: Basic Block Diagram 13-49

13-26 FOIL: Onyx2 Architecture 13-51

13-27 FOIL: GE14 Functions 13-53

13-28 FOIL: GE14 Functional Block Diagram 13-55

13-29 FOIL: Processing Geometric Vertex Data 13-57

13-30 FOIL: RM Functions 13-59

13-31 FOIL: RM7 Block Diagram 13-61

13-32 FOIL: Edge Interpolation 13-63

13-33 FOIL: Fill the Triangle 13-65
13-34 FOIL: Texture Mapping 13-67
13-35 FOIL: Image Memory Processor 13-69
13-36 FOIL: DG5 Functions 13-71
13-37 FOIL: DG5 Functions (continued) 13-73
13-38 FOIL: DG5 Block Diagram 13-75
13-39 FOIL: Composite Video 13-77
13-40 FOIL: Dynamic Framebuffer Resizing 13-79
13-41 FOIL: Multiple Output Channels 13-81
13-42 FOIL: Software Tools 13-83
13-43 FOIL: System Boot Process 13-85
13-44 FOIL: Init Process 13-87
13-45 FOIL: /etc/inittab 13-89
13-46 FOIL: Starting the Graphics Subsystem 13-91
13-47 FOIL: Configuration State Variables 13-93
13-48 FOIL: Clogin Window 13-95
13-49 FOIL: X Display Manager (xdm) 13-97
13-50 FOIL: X Display Manager (continued) 13-99
13-51 FOIL: Login Process 13-101
13-52 FOIL: Diagnostics: irsaudit 13-103
13-53 FOIL: Troubleshooting Tips 13-105
13-54 FOIL: Video Output Format 13-107
13-55 FOIL: ircombine 13-109
13-56 FOIL: Initial ircombine window 13-111
13-57 FOIL: Edit globals window 13-113
13-58 FOIL: Select channel 0 13-115

13-59 FOIL: Select a video format 13-117

13-60 FOIL: A selected channel 13-119

13-61 FOIL: Edit attributes window 13-121

13-62 FOIL: Lab Project 6: irsaudit and ircombine 13-123

Module 14: Fault Isolation: Graphics

14-2 FOIL: Module Objectives 14-3

14-3 FOIL: Fault Isolation: Onyx2 Graphics System 14-5

Module 15: Component Locations & FRUs

15-2 FOIL: Objectives 15-3

15-3 FOIL: Module Topics 15-5

15-4 FOIL: General Parts Removal 15-7

15-5 FOIL: Lab Project 7: Component Locations and FRUs 15-9

Module 16: Origin200 System

16-2 FOIL: Objectives 16-3

16-3 FOIL: Module Topics 16-5

16-4 FOIL: System Overview 16-7

16-5 FOIL: Origin200 Profiles 16-9

16-6 FOIL: Rackmount Origin200 16-11

16-7 FOIL: Logic Carrier 16-13

16-8 FOIL: Main System Board 16-15

16-9 FOIL: Main System Board (continued) 16-17

16-10 FOIL: Memory Layout 16-19

16-11 FOIL: CPU Daughter Card 16-21

16-12 FOIL: PCI Backplane 16-23

16-13 FOIL: Module System Controller (MSC) 16-25

16-14 FOIL: Drive Bays 16-27

16-15 FOIL: Drive Bays (continued) 16-29

16-16 FOIL: Power Supply 16-31

16-17 FOIL: Installation 16-33

16-18 FOIL: Standalone Installation 16-35

16-19 FOIL: Dual-module Installation 16-37

16-20 FOIL: MSC DIP Switch Settings 16-39

16-21 FOIL: Rackmount Installation 16-41

16-22 FOIL: System Operation and LEDs 16-43

16-23 FOIL: Module System Controller 16-45

16-24 FOIL: Diagnostics 16-47

16-25 FOIL: LEDs 16-49

16-26 FOIL: FRUs 16-51

16-27 FOIL: Lab Project 8: Origin200 16-53

Module 17: Support Tools

17-2 FOIL: Module Objectives 17-3

17-3 FOIL: Module Topics 17-5

17-4 FOIL: Introduction to Support Tools 17-7

17-5 FOIL: Support Tools Plan 17-9

17-6 FOIL: Releases 17-11

17-7 FOIL: User Guides 17-13

17-8 FOIL: URLs 17-15

17-9 FOIL: Availmon 17-17

17-10 FOIL: Availmon Installation 17-19

17-11 FOIL: Registration 17-21

17-12 FOIL: Configuration 17-23

17-13 FOIL: Configuration (continued) 17-25

17-14 FOIL: Availmon Reports 17-27

17-15 FOIL: Viewing Reports 17-29

17-16 FOIL: Viewing Reports (continued) 17-31

17-17 FOIL: Availlog file 17-33

17-18 FOIL: Availlog file (continued) 17-35

17-19 FOIL: Availlog file (continued) 17-37

17-20 FOIL: Availlog file (continued) 17-39

17-21 FOIL: Availlog file (continued) 17-41

17-22 FOIL: Availlog file (continued) 17-43

17-23 FOIL: Availlog file (continued) 17-45

17-24 FOIL: Mailing Reports 17-47

17-25 FOIL: Availmon URLs 17-49

17-26 FOIL: SVP 17-51

17-27 FOIL: SVP Process 17-53

17-28 FOIL: SVP test Coverage 17-55

17-29 FOIL: SVP Installation 17-57

17-30 FOIL: SVP Installation (continued) 17-59

17-31 FOIL: Installing SVP from remote location 17-61

17-32 FOIL: Viewing Test Results 17-63

17-33 FOIL: SVP Output 17-65

17-34 FOIL: SVP Output (continued) 17-67

17-35 FOIL: SVP Output (continued) 17-69

17-36 FOIL: SVP Output (continued) 17-71

17-37 FOIL: SVP Output (continued) 17-73

17-38 FOIL: Configuration Data 17-75

17-39 FOIL: SVP Client 17-77

17-40 FOIL: e_mail_client 17-79

17-41 FOIL: RAT 17-81

17-42 FOIL: RAT Functions 17-83

17-43 FOIL: RAT Configuration 17-85

17-44 FOIL: RAT Configuration (configuration) 17-87

17-45 FOIL: FRU 17-89

17-46 FOIL: FRU (continued) 17-91

17-47 FOIL: Sysmon 17-93

17-48 FOIL: PatchFinder 17-95

17-49 FOIL: MDK 17-97

17-50 FOIL: MDK (continued) 17-99

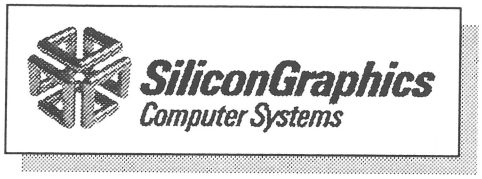
17-51 FOIL: Lab Project 9: Support Tools 17-101

Module 18: Fault Isolation: Entire System

18-2 FOIL: Module Objectives 18-3

18-3 FOIL: Fault Isolation: Entire System 18-5

Index: Student Workbook Index-1

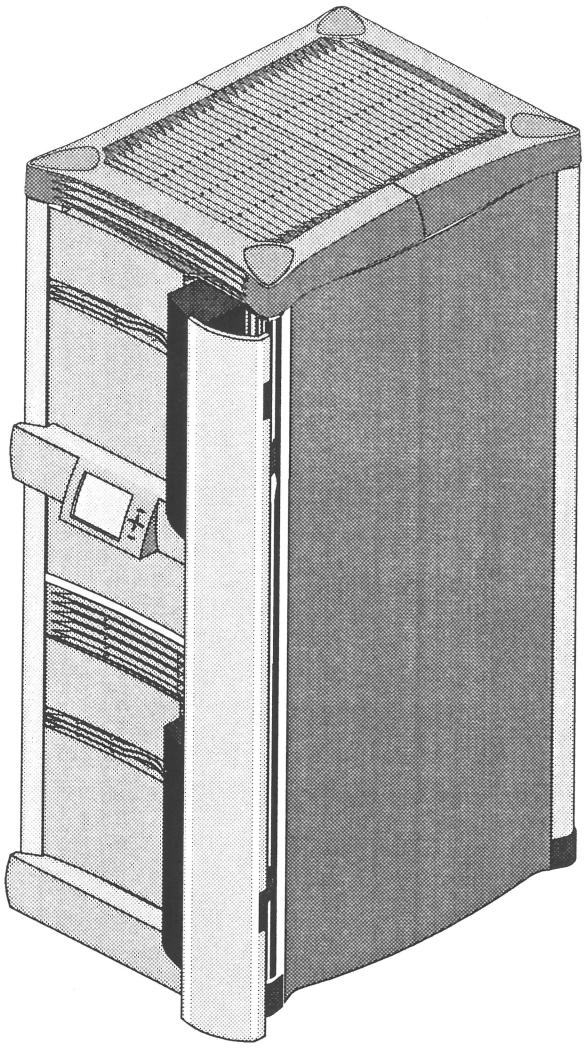


Module 1: Introduction

Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

1-2 FOIL: Welcome

Origin™ Maintenance Course



- Course is 10 days in length.
- Class begins at 8:00 a.m.
- Course consists of:
  - Lecture.
  - Lab projects.
  - Fault isolation lab.

Student Notes

1-3 FOIL: Description

- The goal of this course is to teach the skills necessary to install, configure, and perform first line maintenance on a Silicon Graphics®, Inc. Origin2000™ Rackmount system that contains an Onyx2™ Graphics Subsystem insert module.

- These skills include an understanding of basic data flow; how to locate, load, and run any available diagnostic or system support utility; locate required information from the various supporting technical manuals or Web-based information; and finally, how to troubleshoot to a Field Replaceable Unit (FRU).

- Along with the course ***Student Workbook*** and ***Lab Book***, the students will be supplied with several technical manuals that supply required information for installing and servicing the system.

Student Notes

1-4 FOIL: Equipment:

- The classroom lab contains three 2-rack systems.
- Each 2-rack system has one rack that contains two Origin2000 modules, two Base IO boards, and four IP27 node boards (two in each module).
- The second rack contains one Origin2000 module with two IP27 node boards, and one Onyx2 (KCAR) graphics insert module.
- Each system also contains various peripheral units.

Student Notes

1-5 FOIL: Scope

- Each student is required to complete several lab projects and answer the lab project review questions. The lab project review questions help reinforce the students' knowledge and learning.
- The lab projects and fault isolation labs are used to determine whether course objectives are being met.
- The theory-of-operation is taught using functional block diagrams. This theory provides students with a basic understanding of how the major functional logic areas and system operate.
- The course is 10 days long and includes the following modules:
  - Introduction.
  - Product Line.
  - System Architecture.
  - System Components.
  - Hardware Configurations.
  - ESD: Board Handling and Removal.
  - CrayLink™ Cabling.
  - PROM Monitor.
  - System Controllers.
  - Power-on Diagnostics.
  - Fault Isolation: Origin2000.
  - Graphics Introduction.
  - Onyx2 Graphics.
  - Fault Isolation: Graphics.
  - Component Locations and FRUs.
  - Origin200™ System.
  - Support Tools.
  - Fault Isolation: Entire System.

Student Notes

1-6 FOIL: Prerequisites

Successful completion of the following three courses from the SSE Curriculum:

- **SSE New Hire** course (Phase I).
- **SSE System Administration** course (Phase II).
- **SSE Network Administration** (Phase III).

Student Notes

1-7 FOIL: References

- The student workbook and lab book.
- Course instructors.
- The Origin2000 technical manuals.

Student Notes

1-8 FOIL: Course Objectives

Upon completion of this course, the student should be able to:

- Correctly install and cable the Origin2000 modules.
- Correctly install and cable the Origin200 towers.
- Trace the flow of data through the CPU and Graphics subsystems and correctly describe the purpose of each functional block.
- Correctly run available diagnostics and system support utilities.
- Working with a lab partner, isolate instructor-induced failures within 1 hour of the failure.
- Locate required information from the course material, technical reference manuals, on-line reference documents, and WEB pages.

The course objectives will be met if the student successfully completes the required lab projects and demonstrates proficiency in the required skills.

Student Notes

1-9 FOIL: Performance

The prime objective is to ensure that each student learns the basic skills that are necessary to install, configure, and diagnose induced failures within a reasonable time period.

Lab projects, review questions, and fault-isolation problems are provided to evaluate the student progress in meeting the course objectives.

Course Outline

Description

The goal of this course is to produce a person who is proficient in the skills necessary to install, configure, and perform first line maintenance on a Silicon Graphics Origin2000 Rackmount system containing the Onyx2 Graphics Subsystem insert module.

These skills will include: An understanding of basic data flow; how to locate, load, and run any available diagnostic or system support utility; locate required information from the various supporting technical manuals or web-based information; and finally, how to troubleshoot to a Field Replaceable Unit (FRU).

Along with the course *Student Workbook* and *Lab Book*, the students will be supplied with several technical manuals that supply required information for installing and servicing the system.

Equipment:

The classroom lab systems are comprised of two racks with the following configurations: three Origin2000 modules, each with two IP27 node boards, two of the modules contain Base IO boards, an Onyx2 (KCAR) Graphics Insert module, and various peripheral units.

Scope

- Each student is required to complete several lab projects and answer the lab project review questions. The lab project review questions help reinforce the students' knowledge and learning.
- The lab projects and fault isolation labs are used to determine whether course objectives are being met.
- The theory-of-operation is taught using functional block diagrams. This theory provides students with a basic understanding of how the major functional logic areas and system operate.
- The course is 10 days long and includes the following modules:
  - Module 1: Introduction.
  - Module 2: Product Line.
  - Module 3: System Architecture.
  - Module 4: System Components.
  - Module 5: Hardware Configurations.

- Module 6: ESD: Board Handling and Removal.
- Module 7: CrayLink Cabling.
- Module 8: PROM Monitor.
- Module 9: System Controllers.
- Module 10: Power-on Diagnostics.
- Module 11: Fault Isolation: Origin2000.
- Module 12: Graphics Introduction.
- Module 13: Onyx2 Graphics.
- Module 14: Fault Isolation: Graphics.
- Module 15: Component Locations and FRUs.
- Module 16: Origin200™ System.
- Module 17: Support Tools.
- Module 18: Fault Isolation: Entire System.

Prerequisites

Successful completion of the following three courses from the SSE Curriculum:

- SSE New Hire course (Phase I).
- SSE System Administration course (Phase II).
- SSE Network Administration (Phase III).

References

- The student workbook and lab book.
- Course Instructors.
- The Origin2000 technical manuals.

Course Objectives

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

- Correctly install and cable the Origin2000 modules.
- Correctly install and cable the Origin200 towers.
- Trace the flow of data through the CPU and Graphics subsystems and correctly describe the purpose of each functional block of the block diagram.
- Correctly run available diagnostics and system support utilities.

- Working with their lab partner, isolate instructor induced failures within one hour of the failure.
- Locate required information from the course material, technical reference manuals, on-line reference documents, and WEB pages.

The course objectives will be met if the student successfully completes the required lab projects and demonstrates proficiency in the required skills.

Module Objectives

Each module focuses on specific areas of the system. Using any available documentation and/or tools, the student will be able to meet or perform the stated objectives for each of the following modules:

Module 1: Introduction

No stated objectives for this module.

Module 2: Product Line

Upon completion of this module, using available documentation, you will be able to:

- List the three types of product enclosures.
- List the server and graphics products.
- Describe any general restrictions that may apply to the inter-cabling of products.

Module 3: System Architecture

Upon completion of this module, using available documentation, you will be able to:

- Describe Distributed Shared Memory.
- Explain why the Origin2000 memory bandwidth increases when the node count is increased.
- Explain how cache coherence is maintained in the Origin2000.
- Name the five major components on the node board.
- Explain the function of the router board.
- Locate the XBOW ASIC and describe its function.
- Trace the data flow from a processor to an I/O slot.
- Trace the data flow from one node's local memory to the remote memory on another node.

Module 4: System Components

Upon completion of this module, using available documentation, you will be able to:

- Name and briefly describe the function of each of the eight major components that make up the Origin2000 module.
- Name and briefly describe the function of each of the three major components on the node board.
- List the four ports on the Hub ASIC and what they are connected to.
- Describe the memory banking used on the node board.
- Name the component used to maintain cache coherence in the Origin2000 memory subsystem.
- Be able to determine which memory DIMMs and which directory DIMMs may be used in an expanded directory.
- Name the four router board types and their differences.
- List the four functions of the midplane.

Module 5: Hardware Configurations

Upon completion of this module, using available documentation, you will be able to:

- Name the three ASICs that are used in the system interconnect fabric used in the Origin2000 system, and their function.
- List the node board-to-router board connections.
- Name which type of router is used in the following configurations:
  - 2-node.
  - 4-node.
  - 8-node.
  - 16-node.
  - 64-node.

Module 6: ESD: Board Handling and Skin Removal

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

- Take the proper precautions to guard against ESD damage to equipment.
- Prevent damage to the exterior skins of the equipment.
- Handle the system components without damaging them.

Module 7: CrayLink Cabling

Upon completion of this module, using available documentation and the classroom Origin2000 rack-cluster system, you will be able to:

- Correctly explain how Racks and Modules are numbered so that all installed rack systems are numbered the same.
- Given an existing, correctly-cabled Origin rack system, use the router LEDs to identify a point (in the cabling structure) where a problem in router communication has been detected.
- When installing CrayLink cables, use the router LEDs or power-on boot process to identify configurations that are not valid.
- Correctly explain how the number of router hops (traversals) are determined by the placement of the CrayLink cables and slot positions of the node boards.

Module 8: PROM Monitor

Upon completion of this module, using available documentation and the classroom rack and deskside system, you will be able to:

- Explain what each PROM monitor command is used for.
- Explain what each PROM monitor environment variable is used for.
- Using the ***hinv*** command, display the following: hardware configuration, board serial numbers, and board laser NIC numbers.
- Correctly disable and then re-enable the following: an individual CPU, a memory bank, a PCI bus of the Base IO board.
- Flash the IP27 and Base IO microcode.
- Transfer control of the console from the original console Base IO board to a second Base IO board in the system.
- Boot the system from alternate sources.
- Verify that the networking hardware is functioning.
- Correctly set the baud rate of the Base IO serial port.

Module 9: System Controllers

Upon completion of this module, using available documentation and the classroom rack and deskside system, you will be able to:

- List the features and functions of the Module System Controller (MSC) and Multi-Module System Controller (MMSC).

- Correctly attach an ASCII terminal to the MSC Diagnostic port and perform specific MSC commands that control the following: power-on/ power-off, fan speed, voltage margins, control of power-on diagnostics, and various other status-collecting commands.
- Correctly cable an ASCII terminal, three Origin2000 modules, and an INDY™ system to the rack systems MMSCs, allowing control of the complete system to be handled by the INDY or the ASCII terminal.
- Flash (upgrade) the MMSC microcode.
- Locate and correctly set specified MSC hardware debug switches.
- Demonstrate that individual MSC hardware debug switches can be remotely set on or set off.
- Using the MSC key-switch, correctly put the MSC into any of the following modes: standby, on, or diagnostic.
- Cause a Non-Maskable Interrupt to occur.
- Reset the system through the MSC.

Module 10: Power-on Diagnostics

Upon completion of this module, using available documentation and the classroom rack and deskside system, you will be able to:

- Correctly identify what debug switch would be used to cause the stated function of each switch.
- Given a dump of the power-on boot process, correctly identify where important functions occur, as identified during the boot process lecture.
- Set a specific MSC Hardware Debug switch to cause the associated function to occur.
- Attach an ASCII terminal to the MSC diagnostic port and use the microcode-based utility called POD to perform the following tasks:
  - Nullify the MSC password.
  - Nullify the Command Monitor password.
  - Flash (upgrade) the microcode of an IP27 board.
  - Cause a hardware error dump to occur and run the FRU analyzer.
  - Test local memory.
  - Using a memory error report, identify the failing DIMM pair.

Module 11: Fault Isolation: Origin2000

The objective of this module allows each student time to apply the knowledge learned during modules 1 through 10, and reinforce that

learning by troubleshooting instructor-induced faults within the Origin2000 rack system.

- Each student must demonstrate that a majority of the instructor-induced faults can be identified using any available documentation or tool.
- Each student must demonstrate that diagnostics and special troubleshooting procedures can be used to locate induced failures.

Module 12: Graphics Introduction

Upon completion of this module, using available documentation, you will be able to:

- Name the 3 primary and secondary colors.
- Correctly explain what a pixel is and how many pixels define the resolution of the standard and optional graphics monitors.
- Correctly explain what image memory is and identify what functions the individual memory segments (bit planes) perform, or the type of data that is stored in each memory segment.
- Name the 2 color modes that a programmer can specify.
- State the difference between 2D and 3D graphics.
- Explain what 2 functions the Z-buffer performs.
- Discuss how lighting creates realistic 3D models.

Module 13: Onyx2 Graphics

Upon completion of this module, using available documentation and the classroom Onyx2 systems, you will be able to:

- Identify all the boards and cables required to install any of the various Onyx2 configurations.
- Match the functions performed to the individual Onyx2 boards.
- Correctly state all the special rules or warnings for installing the different configurations of the Onyx2 system.
- Correctly identify all the graphics configuration start-up processes that are called by the init process to initialize the graphics subsystem and state the order that the start-up processes are run.
- Manually start and stop the graphics system.
- Run the irsaudit diagnostic program.
- Configure the Onyx2 system using the ircombine utility.

Module 14: Fault Isolation: Graphics

The objective of this module allows each student time to apply the knowledge learned during modules 12 and 13, and reinforce that learning by troubleshooting instructor-induced faults within the Onyx2 graphics system.

- Each student must demonstrate that a majority of the instructor-induced faults can be identified using any available documentation or tool.
- Each student must demonstrate that diagnostics and special troubleshooting procedures can be used to locate induced failures.

Module 15: Component Locations and FRUs

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

- Take the proper precautions to guard against ESD damage to equipment.
- Prevent damage to the exterior skins of the equipment.
- Handle the system components without damaging them.

Module 16: Origin200 System

Upon completion of this module, using available documentation and the classroom Origin200 systems, you will be able to:

- Identify the system logic carrier and its major subassemblies.
- Describe memory DIMM layout.
- Identify the various control panel switches.
- Describe how to configure a slave module.
- Interpret the MSC status LED display.
- List the system field replaceable units (FRUs).

Module 17: Support Tools

Upon completion of this module, using available documentation, you will be able to:

- Interpret the reports generated by Availmon.
- Query the Silicon Graphics database for Availmon information.
- Interpret the reports generated by SVP.
- Understand how to configure RAT on the Origin system.

- Interpret the reports generated by the FRU analyzer.
- Understand how to run and interpret the results from SVP.
- Use PatchFinder™ to view and download patches.
- Use Sysmon to monitor system activity.
- Understand the function of MDK.

Module 18: Fault Isolation: Entire System

The objective of this module allows each student time to apply the knowledge learned during the course, and reinforce that learning by troubleshooting instructor-induced faults within the complete system.

- Each student must demonstrate that a majority of the instructor-induced faults can be identified using any available documentation or tool.
- Each student must demonstrate that diagnostics and special troubleshooting procedures can be used to locate induced failures.

Performance

The prime objective is to ensure that each student learns the basic skills that are necessary to install, configure, and diagnose induced failures within a reasonable time period.

Lab projects, review questions, and fault-isolation problems are provided to evaluate the student progress in meeting the course objectives.

Daily Schedule

This schedule is only a guide. The actual schedule may vary from class to class due to class size and overall experience level of the students attending the class.

See the next two pages for the daily schedule.

Daily Schedule: Week 1

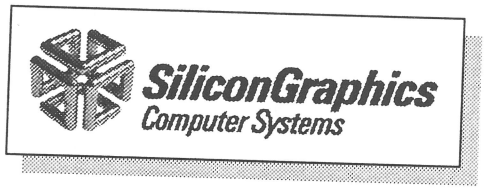
Day	Module	Lab Project	Description	Activity		Hours
				Lecture	Lab	
1	1		Introduction	X		1.0
1	2		Product Line	X		1.5
1	3		System Architecture	X		1.5
1			Lunch			1.0
1	4		System Components	X		1.0
1	5		Hardware Configurations	X		1.0
1	6		ESD: Board Handling and Skin Removal	X		0.5
1		1	ESD: Board Handling and Skin Removal		X	1.5
2	7		CrayLink Cabling	X		2.2
2		2	CrayLink Cabling		X	1.8
2			Lunch			1.0
2	8		PROM Monitor	X		2.0
2		3	PROM Monitor		X	2.0
3	9		System Controllers	X		4.0
3			Lunch			1.0
3		4	System Controllers		X	4.0
4	10		Power-on Diagnostics	X		4.0
4			Lunch			1.0
4	10		Power-on Diagnostics	X		1.5
4		5	POD		X	2.5
5	11		Fault Isolation: Origin2000		X	4.0
5			Lunch			1.0
5	11		Fault Isolation: Origin2000		X	4.0
			Total hours per activity	**20.2**	**19.8**	

Daily Schedule: Week 2

Day	Module	Lab Project	Description	Activity		Hours
				Lecture	Lab	
6	11		Fault Isolation: Origin2000		X	4.0
6			Lunch			1.0
6	12		Graphics Introduction	X		4.0
7	13		Onyx2 Graphics	X		4.0
7			Lunch			1.0
7	13		Onyx2 Graphics	X		1.5
7		6	IRSAUDIT and IRCOMBINE		X	2.5
8	14		Fault Isolation: Graphics		X	4.0
8			Lunch			1.0
8	15		Component Locations and FRUs	X		0.5
8		7	Component Locations and FRUs		X	3.5
9	16		Origin200	X		1.5
9		8	Origin200		X	2.5
9			Lunch			1.0
9	17		Support Tools	X		2.0
9		9	Support Tools		X	2.0
10	18		Fault Isolation: Entire System		X	4.0
10			Lunch			1.0
10	18		Fault Isolation: Entire System		X	4.0
			Total hours per activity	**13.5**	**26.6**	

Total hours lecture: 33.7 (42%)

Total hours lab: 46.3 (58%)



Module 2: Product Line

Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

2-2 FOIL: Module Objectives

- List the three types of product enclosures.
- List the server and graphics products.
- Describe any general restrictions that may apply to the intercabling of products.

Module Topics

- Product description.
- Enclosure types.
- Server products.
- Graphics products.

Student Notes

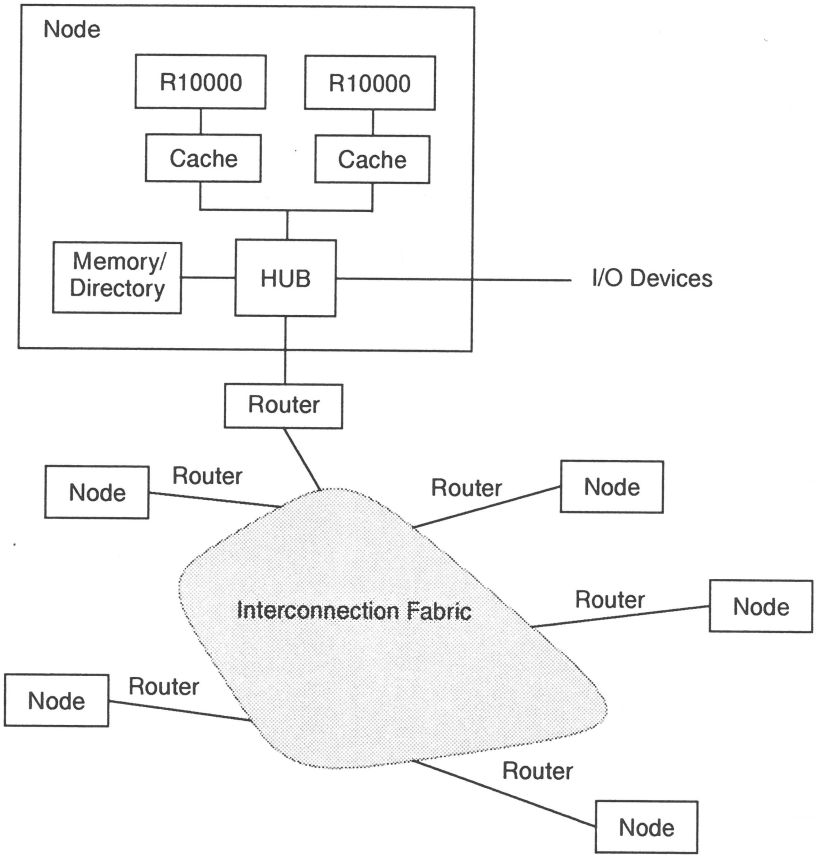
2-3 FOIL: Introduction

- The **Origin2000™** system product line is a scalable high-performance server that is available in either a compact deskside module or a standard 19-inch rack.
- An **entry-level server** (Origin200™) is also available in a deskside tower enclosure.
- There are new graphics systems called **Onyx2™** systems that use Origin2000 nodes for compute engines.

Student Notes

2-4 FOIL: Product Description

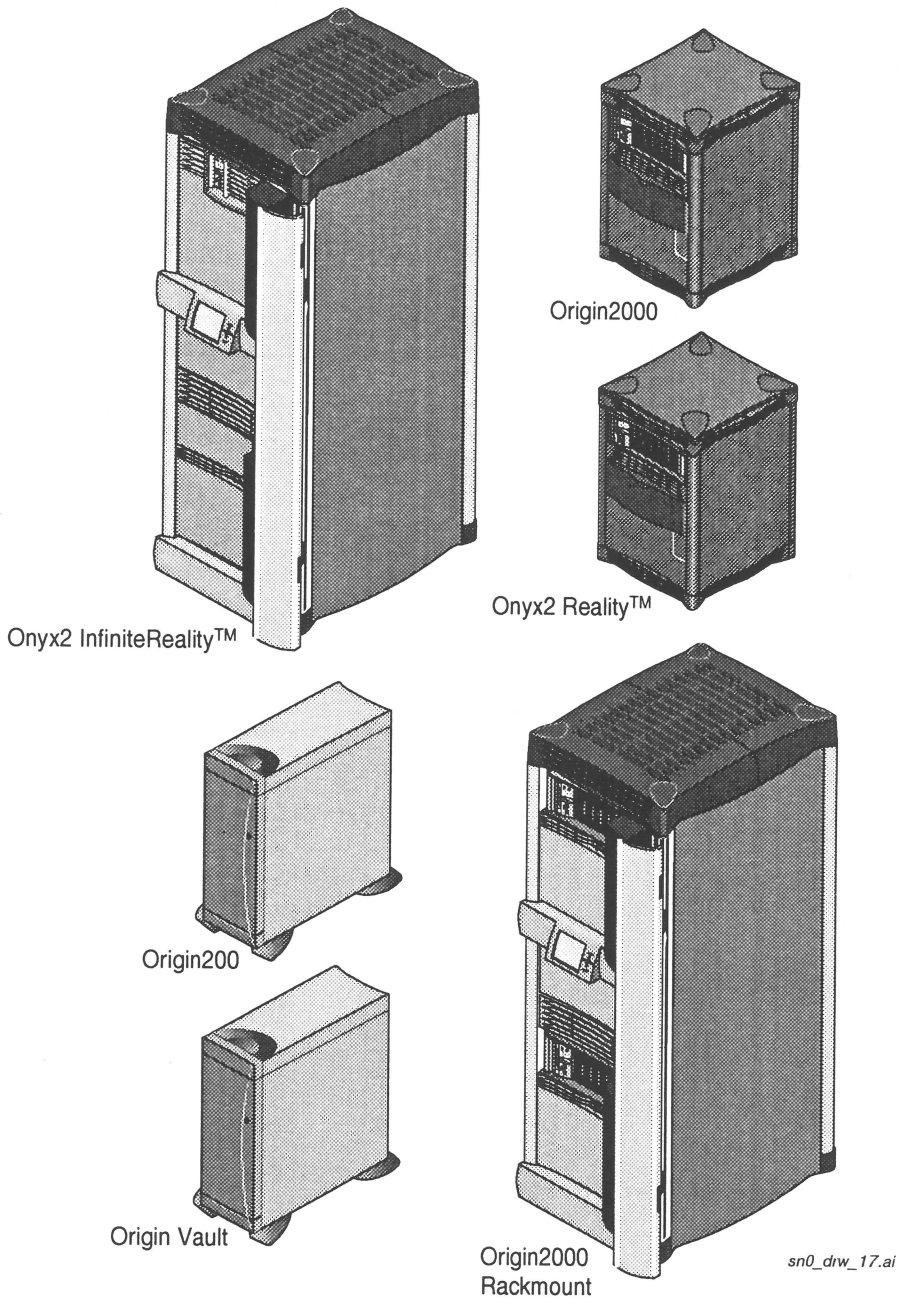
- The Origin2000 product consists of a number of **processing nodes** that are linked together by an **interconnection fabric.**
- Each processing node contains:
  - Up to two **R10000™** processors.
  - A portion of shared memory.
  - A directory for cache coherence.
  - **Two interfaces**: one that connects to I/O devices and another that links system nodes through the interconnection fabric.



Student Notes

2-5 FOIL: Enclosures

- The Origin2000 product line includes three enclosure types
  - **Tower**
  - **Deskside**
  - **Rackmount**



Student Notes

2-6 FOIL: Product Line

Origin Servers

Origin™ Servers range from an entry tower Origin200 system, which is capable of a maximim of 4 CPUs, to the Origin2000 system, which expands from a deskside system to a 128-CPU single shared-memory system.

The **Origin2000** system is available in **three models**:

- A **deskside** system can hold up to 8 CPUs
- An Origin2000 **rack** system can expand to 64 CPUs
- A **CRAY® Origin2000** system can expand from 65 to 128 CPUs

The unique aspect of these three Origin2000 models is that any system, even a deskside system, **can be expanded to up to 128 CPUs** without swapping or trading any components.

- To expand a deskside to a rack, simply remove the skins from the deskside module and mount it in a rack.
- To expand an Origin2000 rack system to a CRAY Origin2000 system, add the **Cray Router**; the existing hardware can be expanded to 128 CPUs.The scalable node product line consists of server-only systems and graphics systems.

Onyx2 Graphics Systems

Onyx2 systems are packaged in both compact deskside units for optimal price/ performance and larger rack units for maximum performance and expansion.

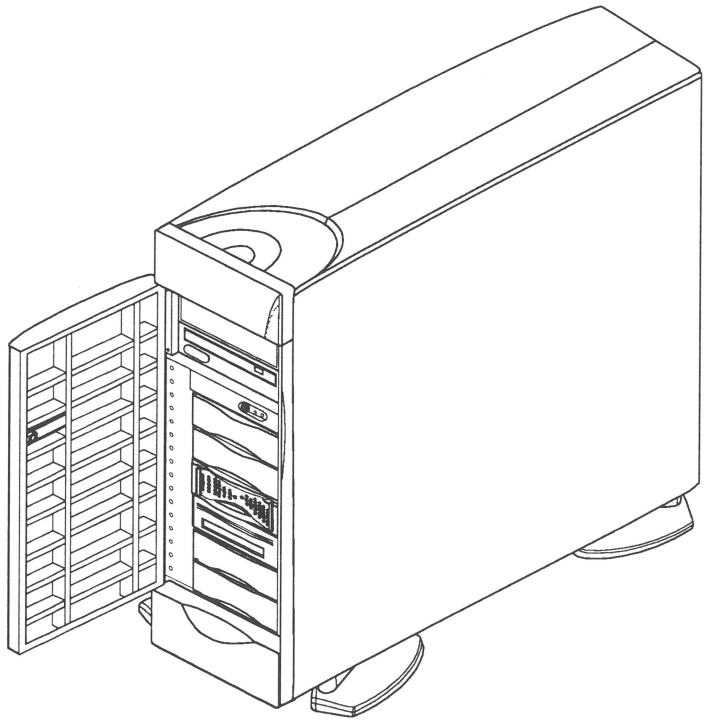
- Onyx2 Reality™ deskside
- Onyx2 InfiniteReality™ deskside and rack system
- Onyx2 InfiniteReality dual-rack system
- Onyx2 RealityMonste™r multi-rack system

Student Notes

2-7 FOIL: Origin200

The **Origin200 system** is a server-only entry-level system that is packaged in a tower enclosure. Each enclosure may contain up to two CPUs and up to three PCI I/O devices. Up to two Origin200 towers may be connected.

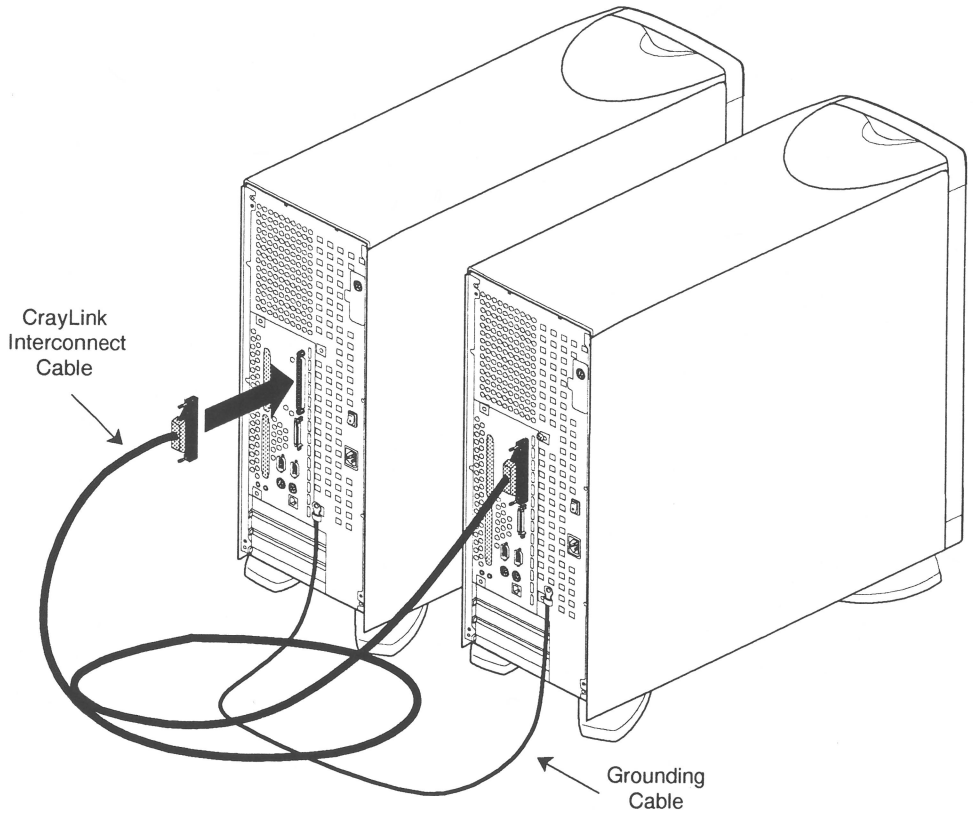
- **One** or **two** CPUs with **1 MB of secondary cache** each.
- **32 MB** to **2 GB** of main memory.
- Three **PCI** expansion slots.
- **One 5.25 inch** or **two 3.5 inch** internal disk drives.
- Up to **six 3.5 inch** disk drives.



Student Notes

2-8 FOIL: Origin200 (continued)

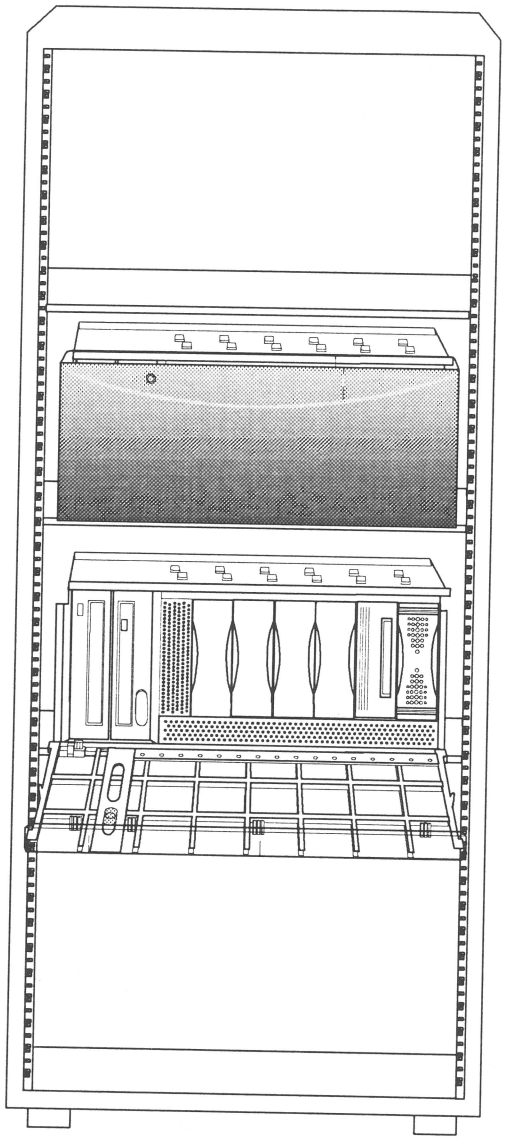
Two Origin200 computers are shown below with a grounding cable and a **CrayLink™** cable.



Student Notes

2-9 FOIL: Origin200 (continued)

The Origin200 system may be mounted in a rackmount enclosure.



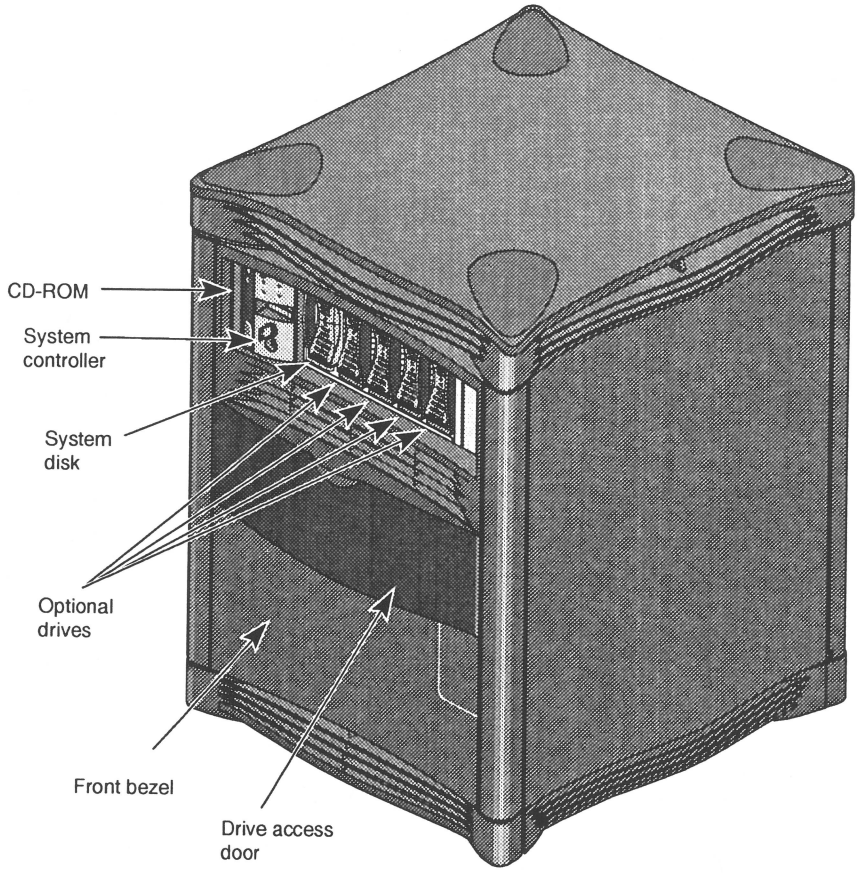
Student Notes

2-10 FOIL: Origin2000 Deskside

The Origin2000 system is a larger version of Origin200 system, which is a server-only system. The Origin2000 system may be packaged in a deskside enclosure.

Each enclosure may contain up to 8 CPUs and up to 12 I/O devices. An Origin2000 system in the deskside module CANNOT be connected to another deskside module.

The enclosure inside the skins is called a **module**, or sometimes, an **8P/12 module**.

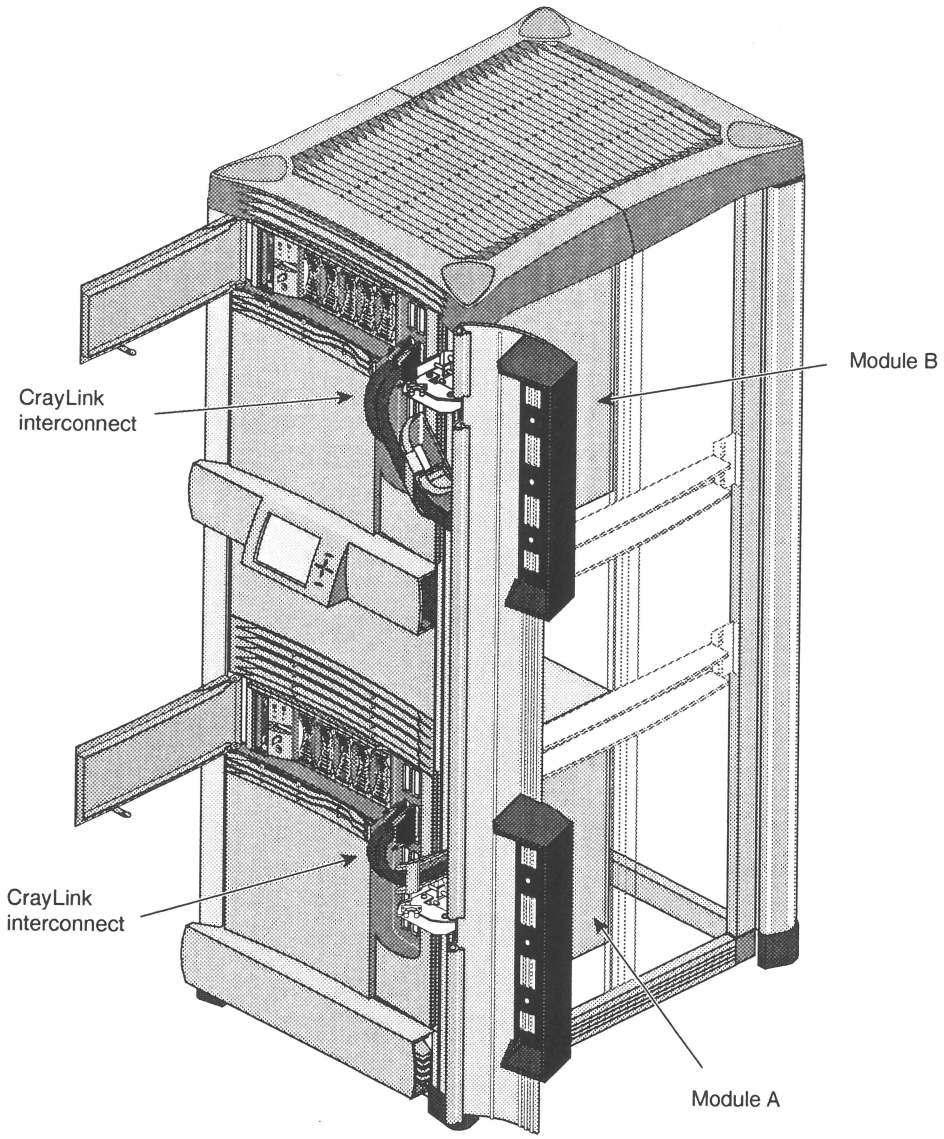


- Up to **eight CPUs** with **1** or **4 MB** of cache each.
- Up to **16 GB** of main memory.
- **One 5.25 inch** disk drive.
- Up to **five 3.5 inch** drives.
- Twelve **I/O** slots.
- CD-ROM reader.

Student Notes

2-11 FOIL: Origin2000 Rack

- The Origin2000 system may also be mounted in a **standard 19-inch** rack.
- Up to **two** modules may be mounted in a rack.
- The two modules in the rack (shown in the illustration below) may be connected to one another and to modules in another rack.
- Each module may contain up to 8 CPUs. This rack, therefore, may contain up to 16 CPUs.
- The largest **multi-rack** configuration presently is **128 CPUs**.



Student Notes

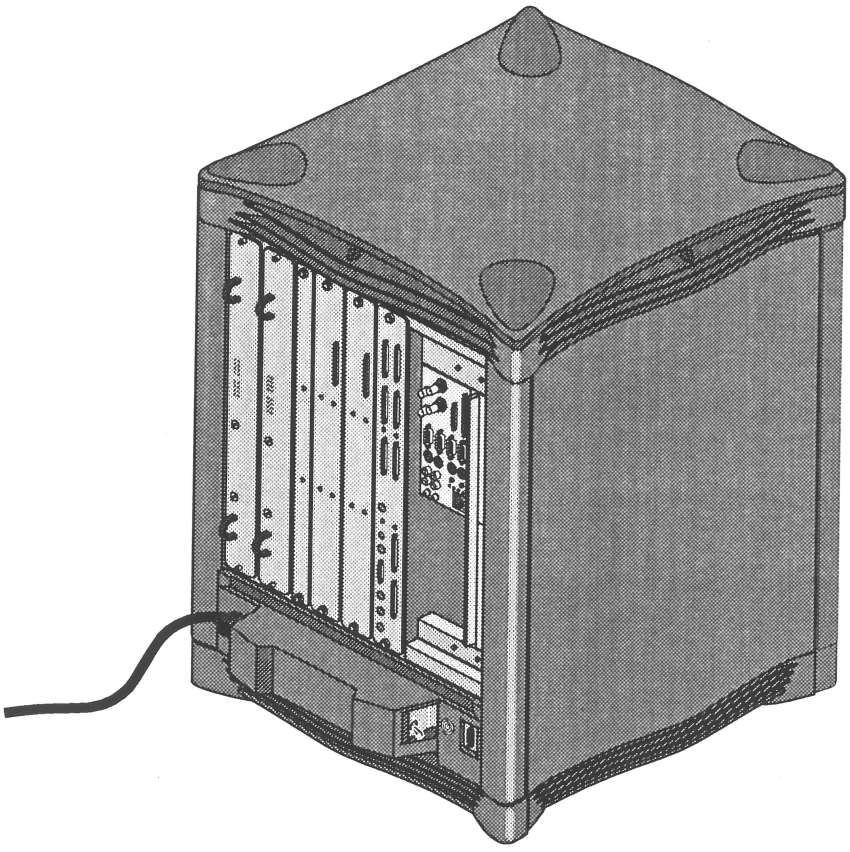
2-12 FOIL: CRAY Origin2000 System

- Origin2000 multi-rack systems may be interconnected up to 64 CPUs (4 racks)
- The **CRAY Origin2000 system** is a multi-rack system that may interconnect up to 9 racks (**8 server racks** and **1 Cray Router rack**) for 128 CPUs.

Student Notes

2-13 FOIL: Onyx2 Deskside

- **Onyx2** is the marketing code for the **InfiniteReality2 graphics subsystem.**
- The **Onyx2** system comes in **two deskside models.**
  - **Onyx2 Reality.**
  - **Onyx2 InfiniteReality.**
- The **Onyx2** system contains up to **two** compute nodes (4 CPUs) and one graphics head.
- **Onyx2 deskside** system is packaged in a module that may be mounted either in a **deskside** enclosure or in a **rackmount** enclosure.
- An **Onyx2** deskside system **cannot** be connected to another deskside module.



Student Notes

2-14 FOIL: Onyx2 Deskside (continued)

Onyx2 Reality system components:

- Up to **four CPUs**, each with **1 MB** of **cache.**
- **64 MB** to **4 GB** of memory.
- Single graphics pipeline with **one** or **two** RM8 (Raster Manager) boards.
- Frame buffer capacity of **40 MB.**
- Four **XIO** slots.
- One **PCI** expansion slot.
- **19 inch** monitor (**1280 x 1024** pixels).

The Onyx2 InfiniteReality deskside model is a graphics system with up to 4 CPUs, each with 4 MB of cache and one graphics head.

- Up to **4 CPUs**, each with **4 MB** of **cache.**
- **64 MB** to **4 GB** of memory.
- Single graphics pipeline with **one** or **two** RM7 (Raster Manager) boards.
- Frame buffer capacity of **80 MB**.
- Four **XIO** slots.
- One **PCI** slot.
- Super-wide ultra-high **24 inch** monitor (**1920 x 1200** pixels).

Student Notes

2-15 FOIL: Onyx2 Rack Systems

The Onyx2 InfiniteReality rack system may have up to 8 **CPUs** with **4 MB** of **cache** and **two graphics heads**.

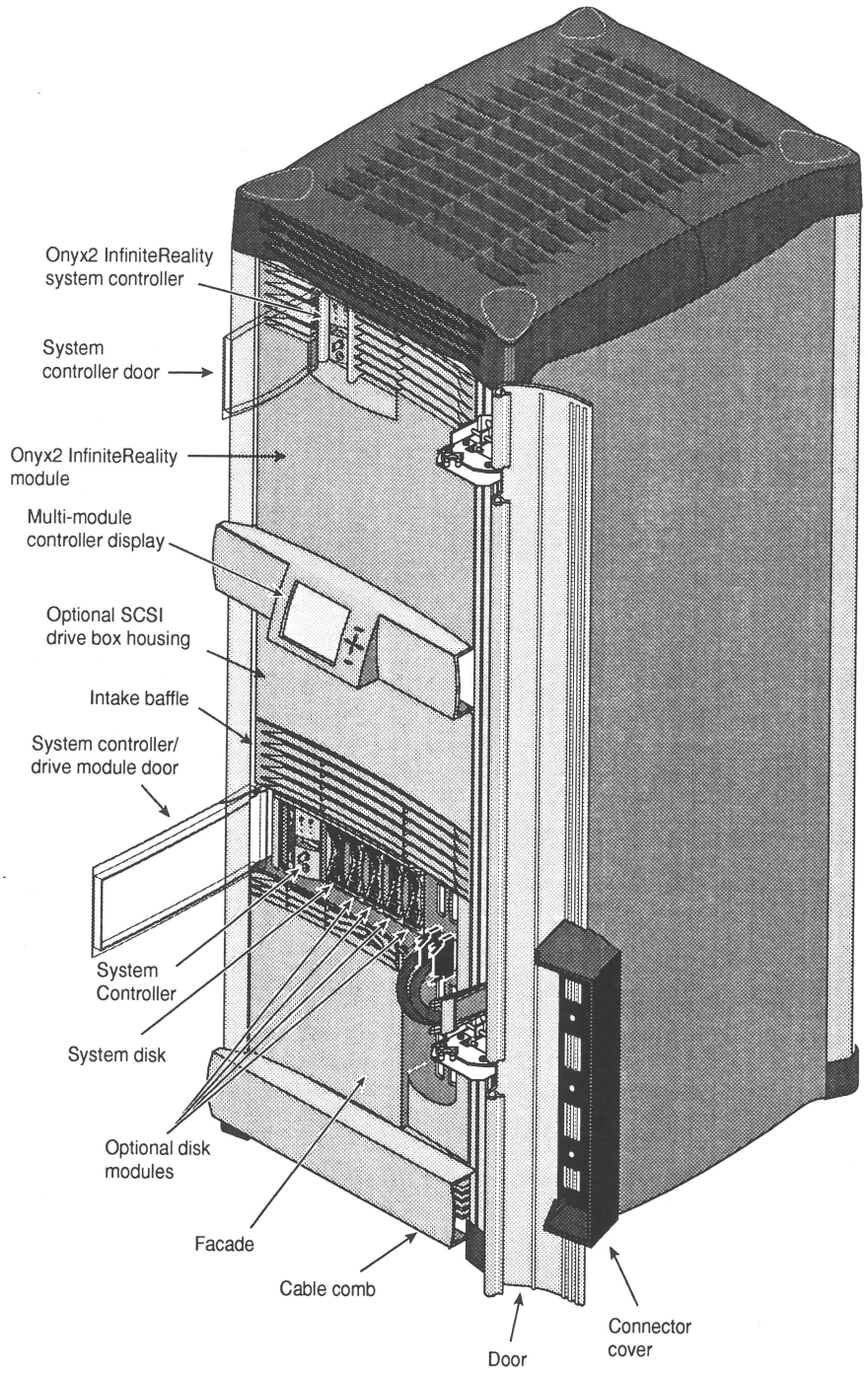
- Up to **eight CPUs** each with **4 MB** of **cache.**
- **64 MB** to **8 GB** of memory.
- Two graphics pipelines (heads).
  - A two RM7 pipe.
  - A four RM7 pipe.
- Super-wide ultra-high **24 inch** monitor (**1920 x 1200** pixels).

The Onyx2 InfiniteReality multi-rack system includes up to **24 CPUs** and up to **six graphics heads**.

- Up to **24 CPUs** each with **4 MB** of **cache.**
- **64 MB** to **12 GB** of memory.
- Up to **six graphics pipelines**:
  - Three with **one** or **two** 80-MB frame buffer Raster Managers.
  - Three with **one**, **two**, or **four** 80-MB frame buffer Raster Managers.
- Super-wide ultra-high **24 inch** monitor (**1920 x 1200** pixels).

Student Notes

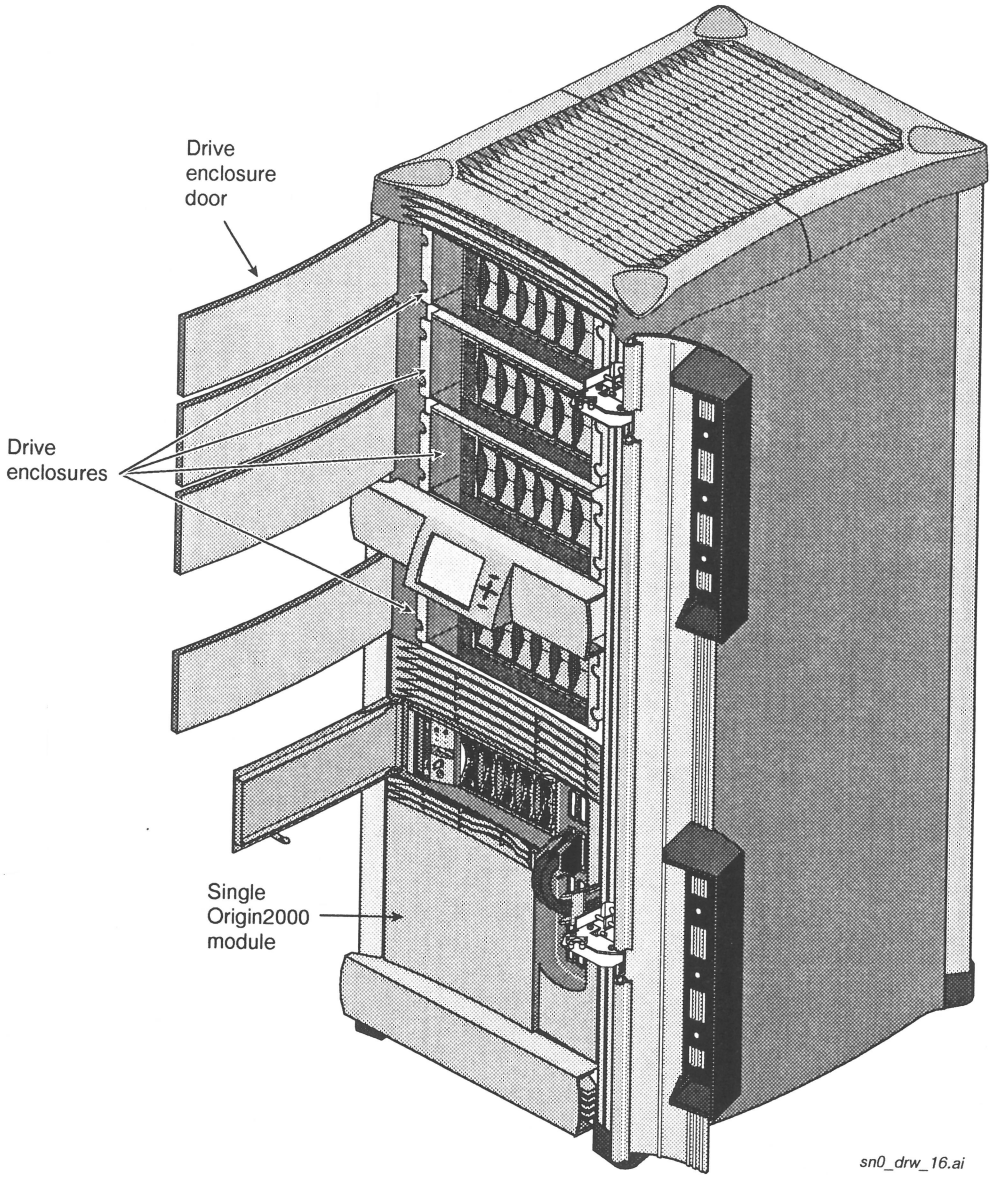
2-16 FOIL: Origin Rack Systems (continued)

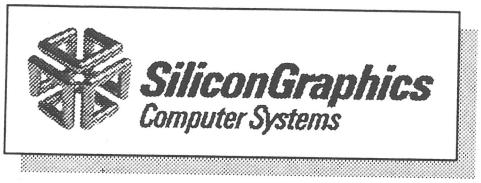


Student Notes

2-17 FOIL: Origin Vault

- **The Origin Vault** is a disk enclosure that may be placed either **vertically** on the floor in a **tower enclosure**, or **horizontally** in a **rackmount enclosure**.
- Up to **six** 3.5 inch SCSI disk drives and **two** 5.25 inch SCSI disk drives may be mounted in an Origin Vault enclosure.





Module 3: System Architecture

Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

3-2 FOIL: Module Objectives

- Describe Distributed Shared Memory.
- Explain why the Origin2000 memory bandwidth increases when the node count is increased.
- Explain how cache coherence is maintained in the Origin2000 system.
- Name the five major components on the node board.
- Explain the function of the router board.
- Locate the XBOW ASIC and describe its function.
- Trace the data flow from a processor to an I/O slot.
- Trace the data flow from one node's local memory to the remote memory on another node.

Student Notes

3-3 FOIL: Module Topics

- Design philosophy differences between CHALLENGE™ and Origin2000™ systems.
- System features.
- System components.
- Key ASICs.
- Crossbar switches.
- NICs (number-in-can).
- Origin2000 overview.
- System block diagram.
- Data integrity.
  - Link-level protocol.
  - SECDED (error correction).
- Memory organization.

Student Notes

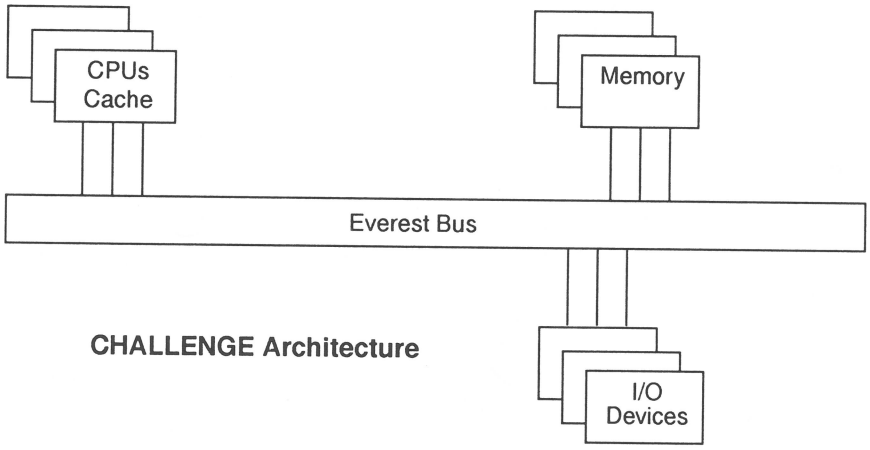
3-4 FOIL: CHALLENGE vs. Origin2000 Systems

The major differences between the CHALLENGE system and the Origin2000 system are how the memory is laid out and how data and control signals are passed from node to node (processor to memory).

- CHALLENGE **memory** is on **memory boards.**
- CHALLENGE **processor** and **cache** are on their **own boards.**

  Address and data must travel on the bus from the processor board to a memory board because all memory is remote from the processor.
- I/O data and control also travels on the bus.
- **Only one processor can access the bus during any given clock period**; that is, this one resource has to be time sliced between as many as 36 CPUs and memory and I/O.

  As activity increases, throughput for each CPU decreases because it is **timesharing** the bus.



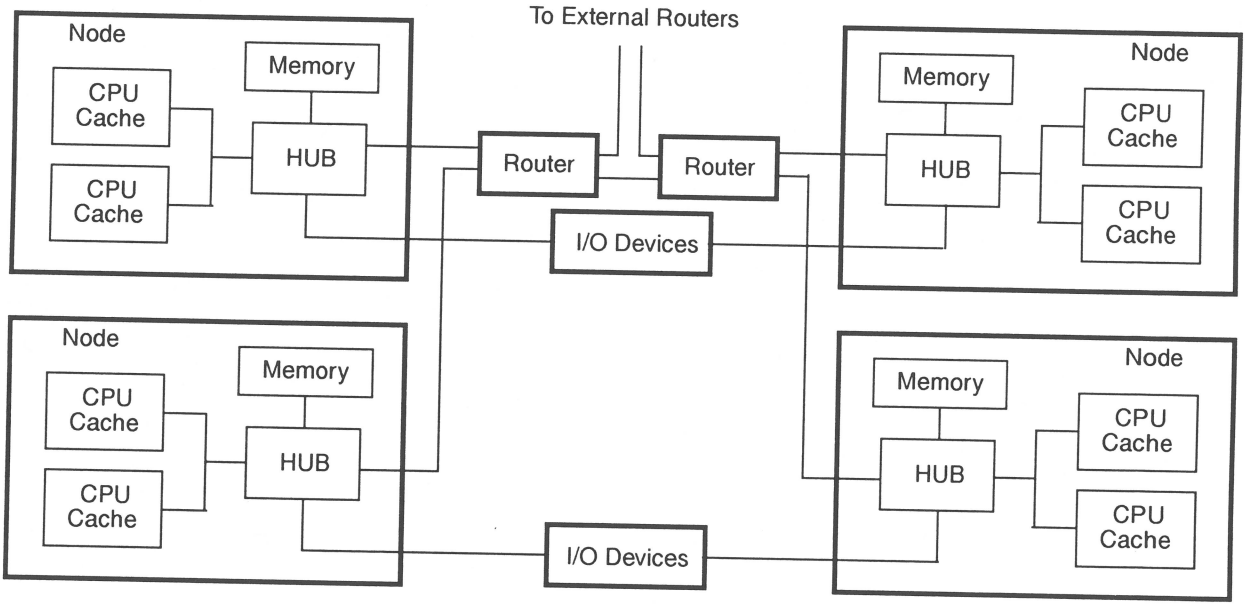
CHALLENGE Architecture

Student Notes

3-5 FOIL: CHALLENGE vs. Origin2000 Systems (continued)

The Origin2000 system, however, uses **point-to-point connections** to pass data, control and addresses, and I/O data. Because of this, the bandwidth scales **linearly** as the system grows.

- Supports a large number of processors (currently 128)
- Lower entry costs because you do not have to buy a bus that you may never fully utilize.
- Expandable I/O connectivity.
- Higher total memory bandwidth.



Origin2000 Module Architecture

Student Notes

3-6 FOIL: System Features

The following characteristics make the Origin2000 architecture different from previous system architectures.

Scalable

The Origin2000 system is easily scaled by **linking nodes** together over an **interconnection fabric**. System bandwidth scales linearly with an increase in the number of processors and the associated interconnection fabric.

Modular

Standard processing nodes compose the system.

- Each node contains:
  - Up to **two** R10000 processors with **32 KB** each of integral **instruction** and **data cache**.
  - **1** or **4 MB** of **secondary cache**.
  - **64 MB** to **4 GB** of **memory**, protected by Single Error Correction/Double Error Detection (**SECDED**).
  - A **directory** for **cache coherence**.
  - An **I/O interface**.
  - A **system interconnection** port.
- Standard nodes are housed in modules.
- The system may be expanded by adding nodes to modules or by adding modules to the system.

Connectivity

The Origin2000 system uses an **interconnection fabric** to link system nodes to **internal crossbars** within the system ASICs (HUB, Router, XBOW).

- The interconnection fabric is a web of dynamically allocated **switch-connected links** that attach nodes to one another.
- **Crossbars** are part of the interconnection fabric and are located inside several of the ASICs.

Student Notes

3-7 FOIL: System Features (continued)

Distributed shared memory and distributed shared I/O

Origin2000 memory is physically dispersed **throughout** the system for faster processor access.

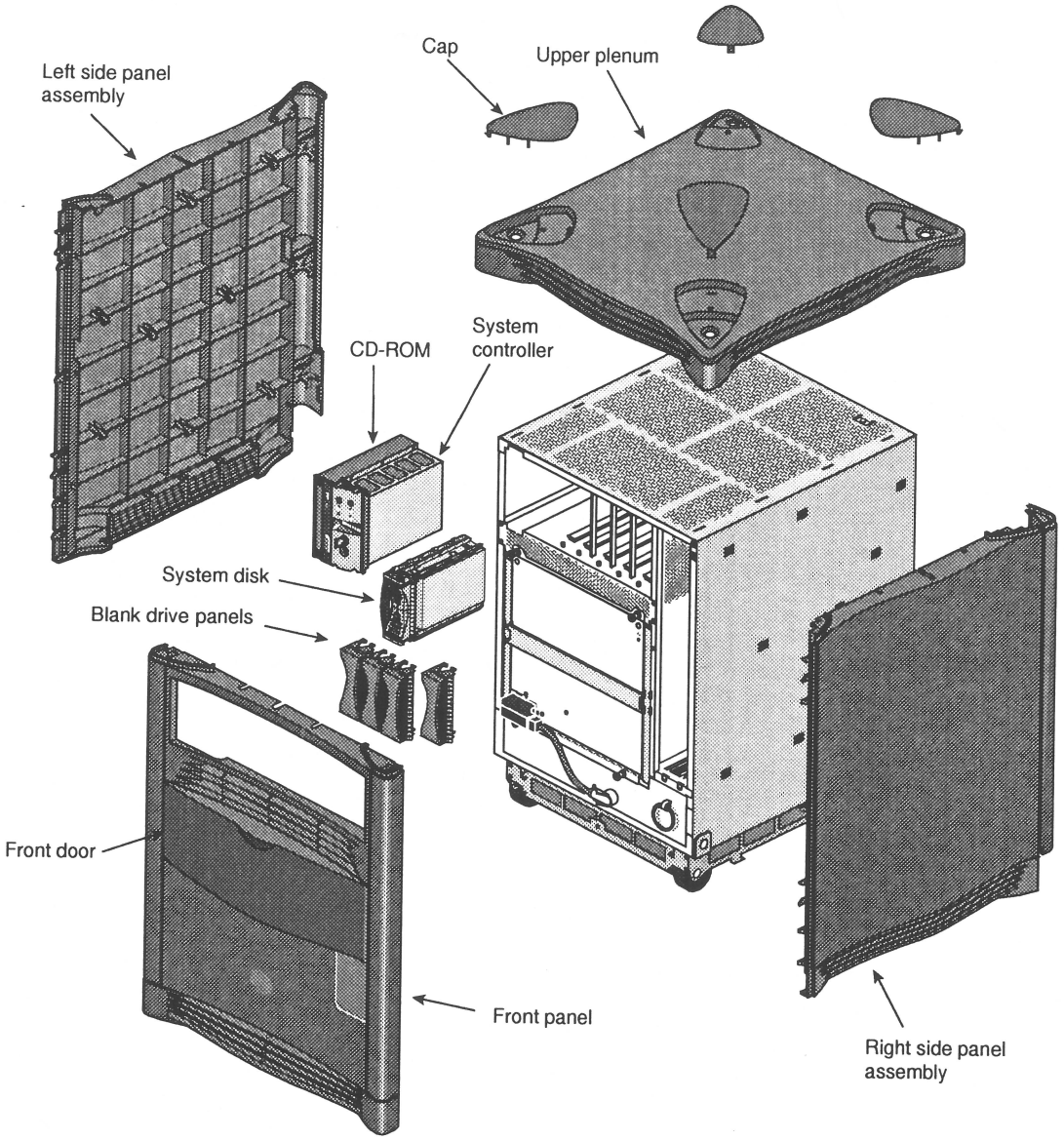
- Main memory is universally accessible and **shared between all processors** in the system.
- **Page migration** hardware moves data into memory closer to a processor that frequently uses it, which reduces latency.
- **Cache coherence** is maintained by using **directories** and a **directory-based cache coherence protocol**.
  - The Origin2000 system uses **caches** to reduce memory latency.
  - **Cache coherence** is maintained by a hardware directory that is distributed among the nodes.
  - **Cache coherence** is applied across the **entire system** and all memory.
  - A directory protocol relies on **point-to-point** messages that are sent only to those **CPUs that actually use the cache line**.
- Similarly, **I/O devices are distributed among the nodes**.
- Any node in any module can connect to any I/O device in the system.

Student Notes

3-8 FOIL: System Components

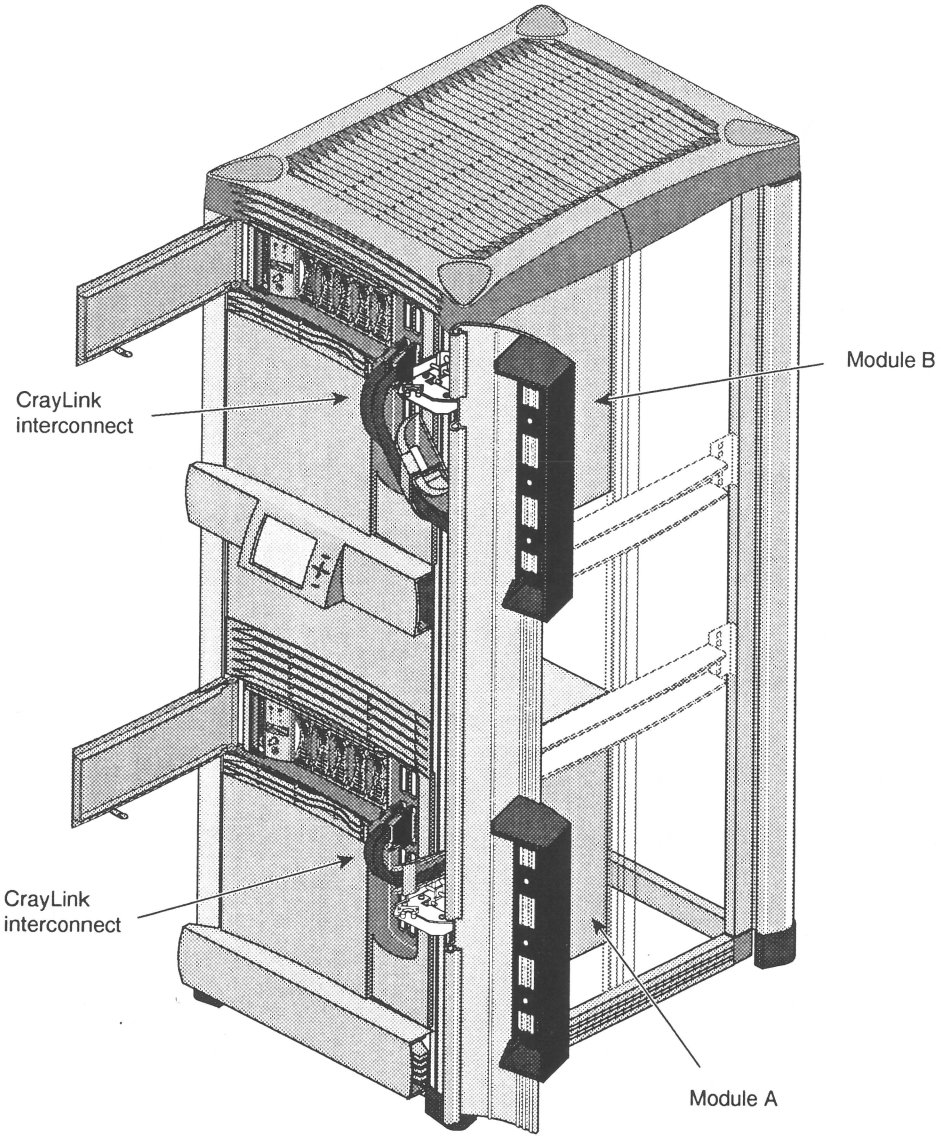
The basic building block for the Origin2000 system is the **module**. Each module may contain as many as 8 processors and 12 I/O devices. This module is sometimes called the **8P/12 module**.

- A module may be mounted in a deskside enclosure, or up to 2 modules may be mounted in a rack. A rack may contain up to 16 processors.
- **Module-to-module connection** intra- or inter-rack is **made with cables**.
- As many as 8 racks may be connected for a 128-processor system.



Student Notes

3-9 FOIL: System Components (continued)



Student Notes

3-10 FOIL: System Components (continued)

The components that compose a system are as follows:

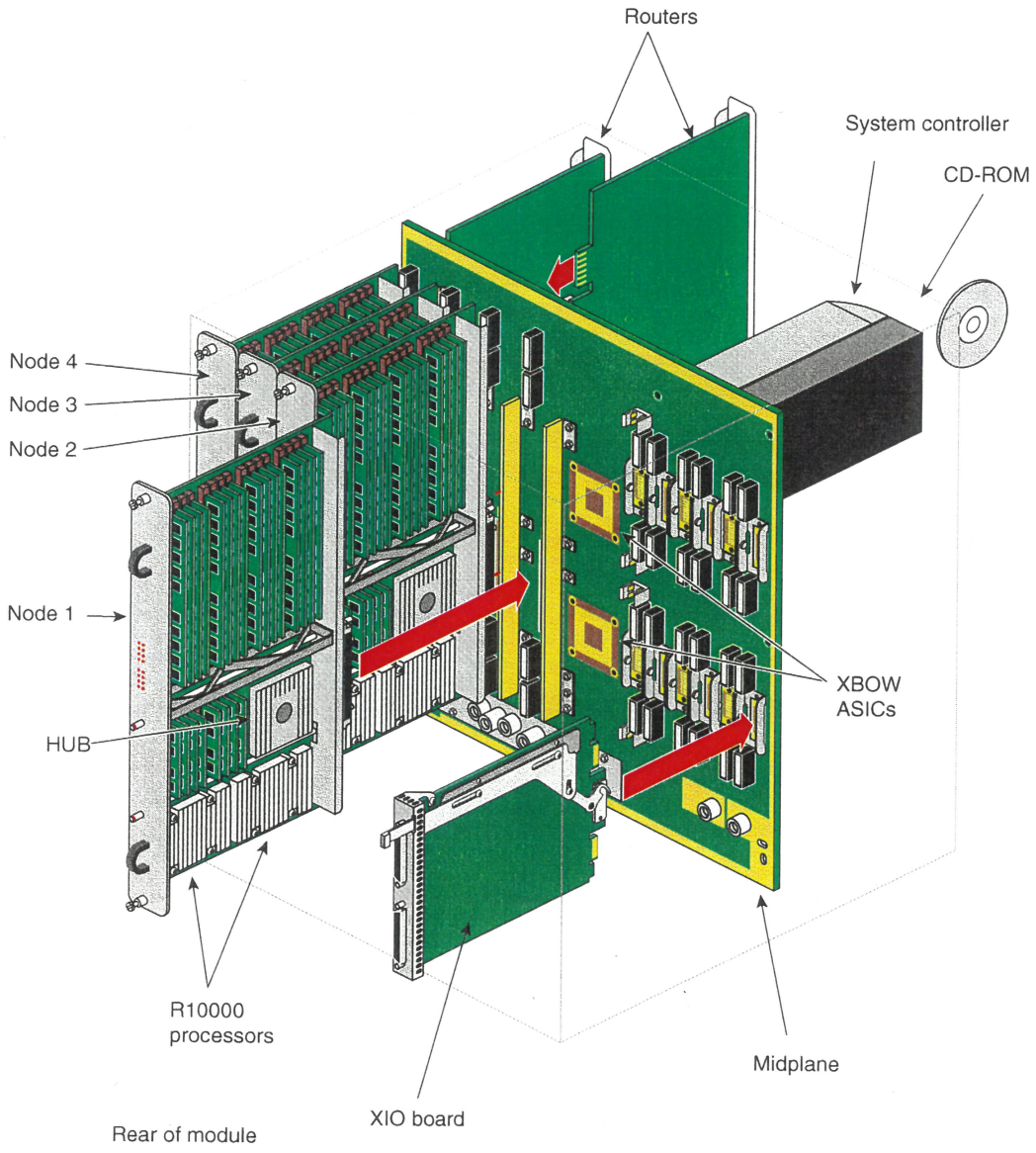
- One to 4 node boards.
  - 1 or 2 processors per node.
  - Primary and secondary cache.
  - Main and directory memory.
  - HUB ASIC to connect to other nodes, routers, and to I/O.
- 1 or 2 router boards.
  - Connects to nodes or other routers.
  - Router ASIC.
- One midplane board.
  - Boards plug into this board on both sides.
  - Connects node to router.
  - Connects node to Crosstalk I/O boards.
  - Converts node's STL to Crosstalk protocol.
  - Distributes power and clock to boards in the module.
  - Two XBOW ASICs.
- Up to 12 I/O Crosstalk boards.
  - Converts the midplane's Crosstalk protocol to PCI protocol.
  - Connects to an external device.
  - IOC3 and Bridge ASICs.
- Up to 5 SCSI disks.
- An optional CD-ROM reader.
- One module system controller (MSC).
  - Provides the minimum functionality for safe operation of a single enclosure.
  - Plugs into the midplane and all switches and displays are accessible from the front of the module.
  - Provides additional functionality related to maintenance and serviceability.

Student Notes

3-11 FOIL: System Components (continued)

- Each rackmount system also contains a multi-module system controller (MMSC).
- Each rackmount system may contain an Origin Vault disk enclosure.

  May contain up to six 3.5-inch SCSI drives and two 5.25-inch SCSI disks.
- Cables to connect nodes intra- and inter-module.
- NICs (number-in-can).



Student Notes

3-12 FOIL: Key ASICs

There are several key **application-specific integrated circuits (ASICs)** that control the flow of address, data, and control signals in the Origin2000 system.

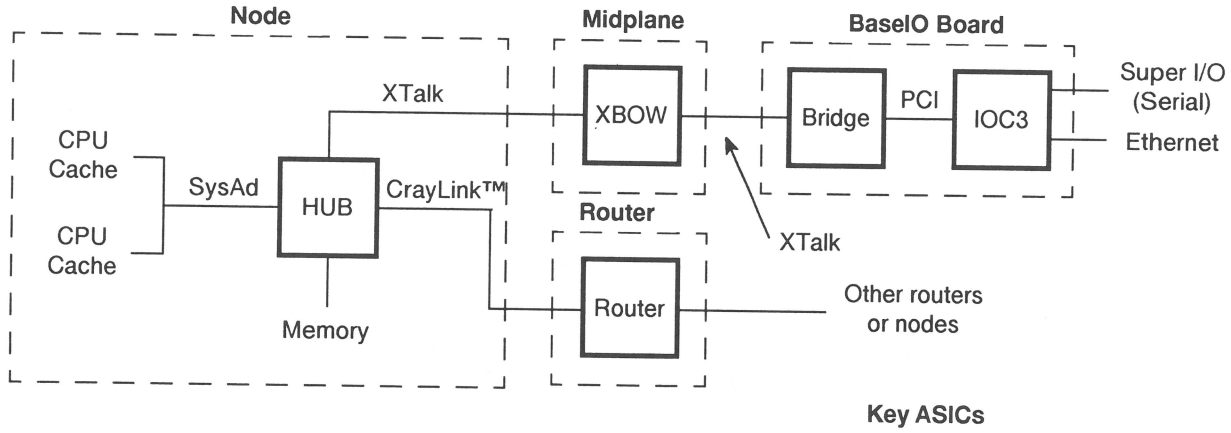
- **HUB**
  - The HUB ASIC allows the CPUs on the node to read and write both local and remote memory.
  - The HUB ASIC also allows the node to write and read any I/O device in the system.
- **XBOW**

  The two XBOW ASICs on the midplane connect to the HUB ASICs in the module.
- **Router**

  The router connects the HUB ASIC of one node to that of another node or to other Router ASICs.
- **Bridge**

  The Bridge ASIC is mounted on an XIO board and translates Crosstalk to PCI and vice versa.
- **IOC3**

  The IOC3 ASIC connects to Super Serial I/O and Ethernet ports on the Base I/O board.



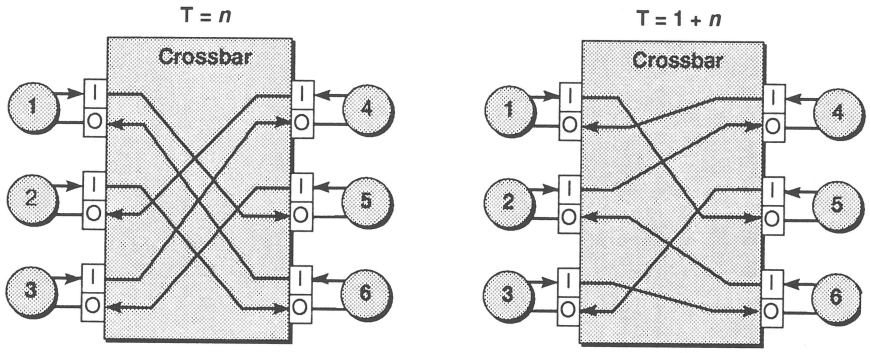
Student Notes

3-13 FOIL: Crossbar Switches

- Several of the ASICs are intelligent crossbar switches.

  A **crossbar** switch is able to switch **input-to-output** connections each **clock period**, if necessary.

- The diagram below shows that input 2, for example, is connected to output 6 in the first clock period, and to output 4 in the next clock period.



Student Notes

3-14 FOIL: Number In a Can (NIC)

All boards in the system, with the exception of graphics boards, have a unique serial number (Globally Unique IDentifier - GUID) that is contained in a chip that is soldered to the board. In addition, two boards have a second NIC that is removable.

- Boards with one NIC:
  - Node board.

    In the node board only, the serial number is read from the NIC and stored in a register on the network interface side of the HUB ASIC. This number can be read from other nodes.
  - Router board.
  - Media I/O board (MIO).
  - All XIO boards.
- Midplane board has two NICs.

  Board serial number.

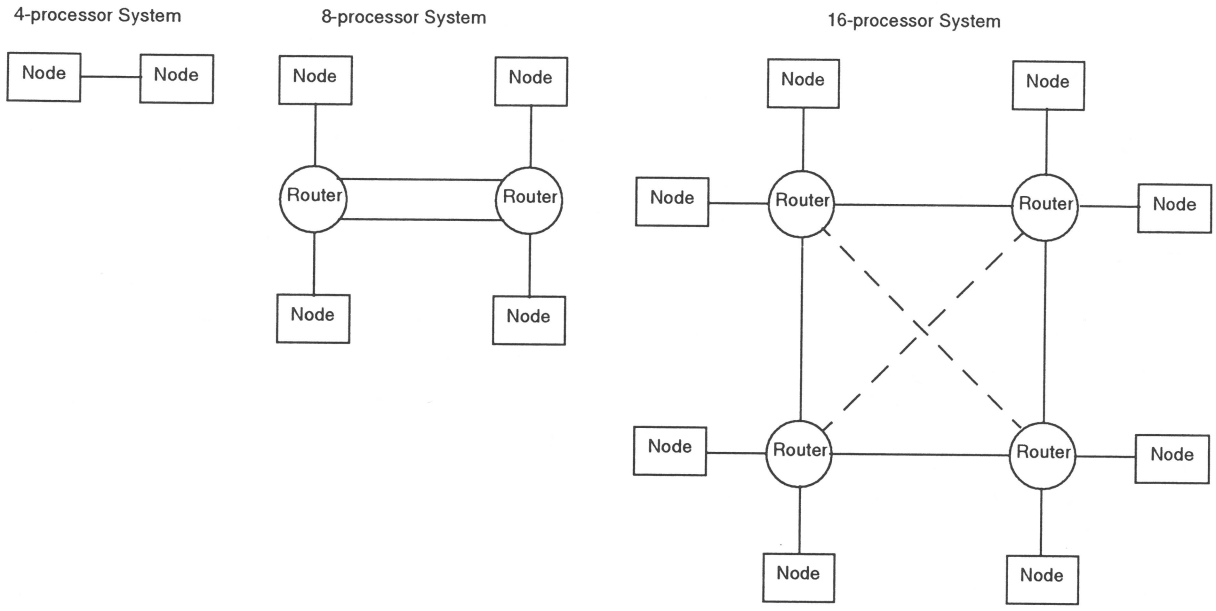
  Removable NIC holds the module serial number.
- IO6 Board.
  - Board serial number.
  - Removable Ethernet address.

Student Notes

3-15 FOIL: Origin2000 Overview

The new design philosophy for the Origin system is known as the **S2MP** architecture. S2MP means **scalable shared memory processing**, the industry's first 64-bit cache-coherent non-uniform memory access (cc-NUMA) architecture.

The basic technology that enables the Origin system is Distributed Shared Memory (DSM), which replaces the central memory system bus in an Everest system with a general interconnect network called CrayLink, whose bandwidth scales according to the number of processors.



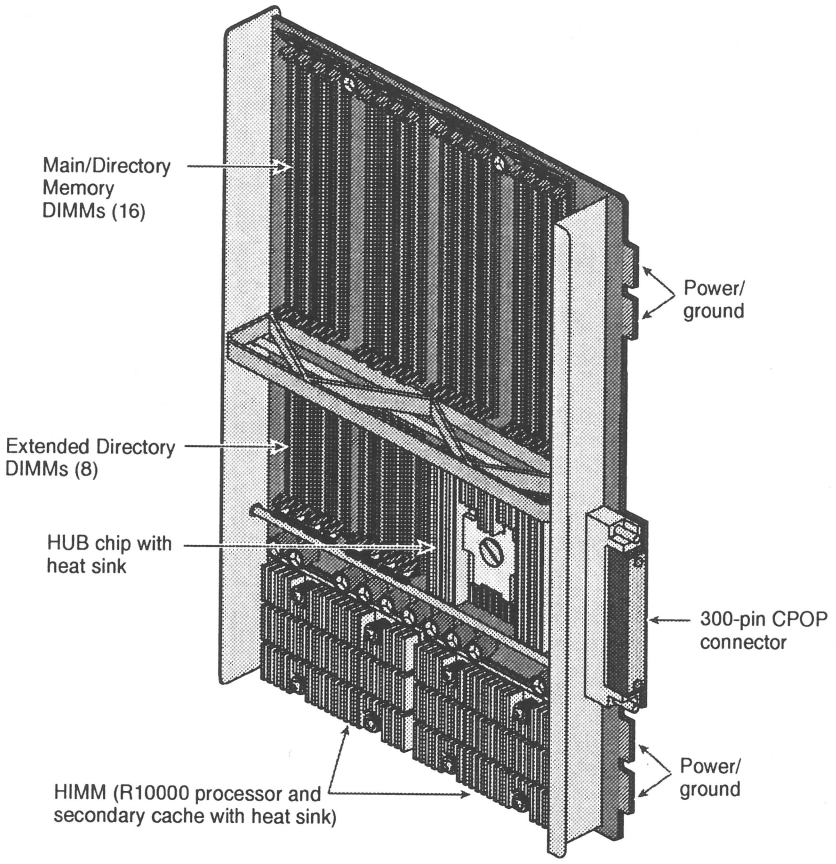
- Each node can support up to 4 GByte of memory
- This memory can be accessed by all nodes in the system.
- Cache coherency is maintained with a directory-based protocol.
- The nodes are interconnected by CrayLink cables via routers.

Student Notes

3-16 FOIL: Node Board

The node is the basic building block of the Origin server computer system. It contains:

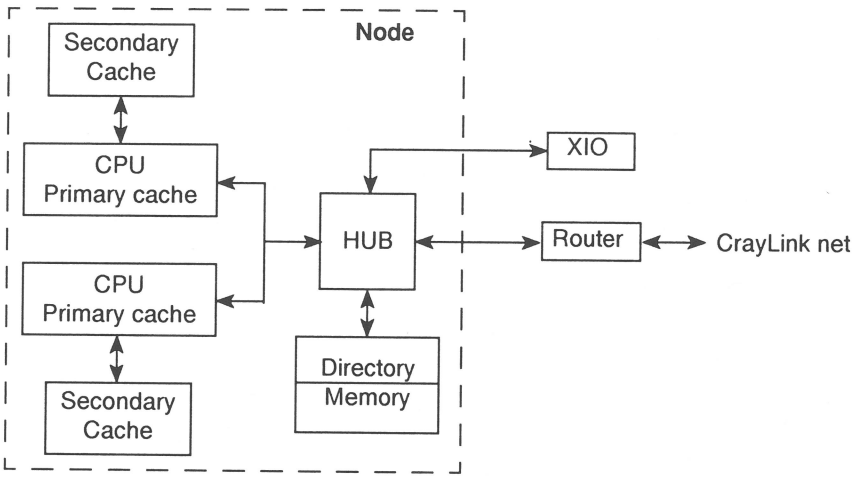
- Up to two horizontal memory modules (**HIMMs**), each with:
  - An **R10000** processor with integral 32 Kbytes of **primary instruction** and **data cache.**
  - **1** or **4** Mbytes of **secondary cache.**
- Slots for up to **8 banks** of memory mounted on dual inline memory modules (**DIMMs**). The DIMMs contain both main memory and directory memory.
- Slots for **extended directory memory** for systems having more than **32 processors.**



Student Notes

3-17 FOIL: Node Board (continued)

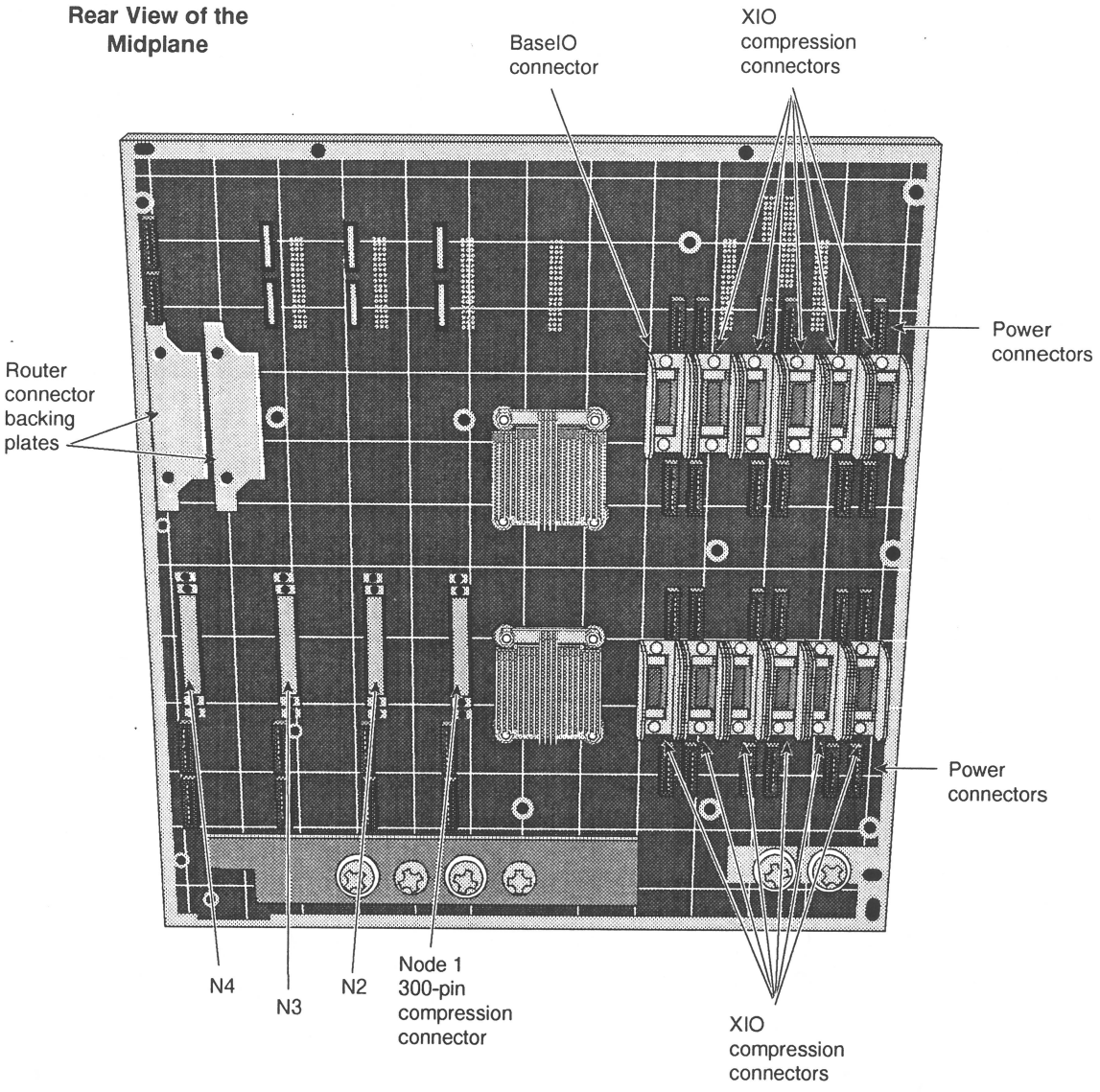
- A **HUB ASIC**, which is the node controller.
  - It passes memory data from local memory to the CPUs on the node, and to remote memory on another node possibly via a router over the CrayLink net.
  - It also provides a connection to the XIO boards in the module, or to XIO boards in another module.



Student Notes

3-18 FOIL: Midplane

The midplane is located in the center of the module. Router boards, disk units, and the system controller plug into the front of the midplane. Node boards and XIO boards plug into the rear of the midplane.

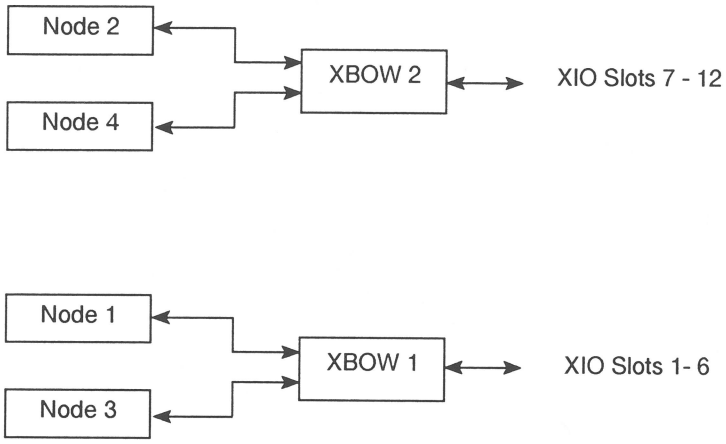


The midplane has active components mounted on it that distribute I/O data from the nodes to the XIO boards. The midplane also connects nodes to routers.

Student Notes

3-19 FOIL: Midplane (continued)

There are 12 XIO slots and 2 XBOW ASICS mounted on the midplane. Each XBOW ASIC connects to half of the XIO slots. Two node boards connect to one or the other XBOW ASIC.

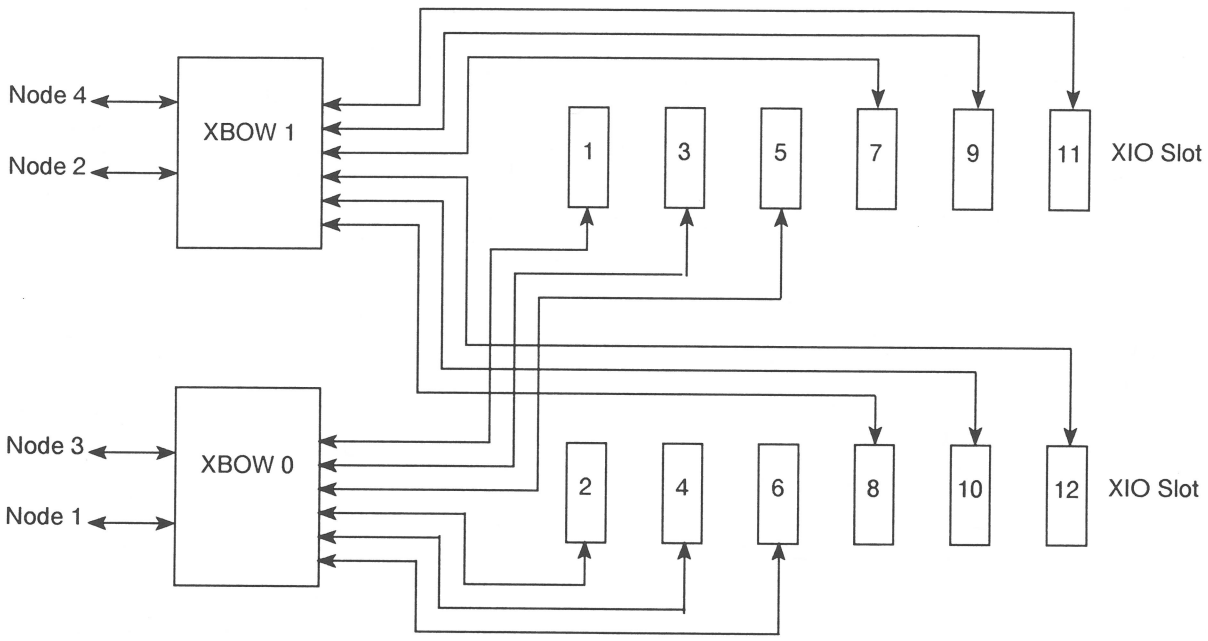


Student Notes

3-20 FOIL: XIO Slots

The physical locations of the XIO slots and the XBOW ASICs are as follows:

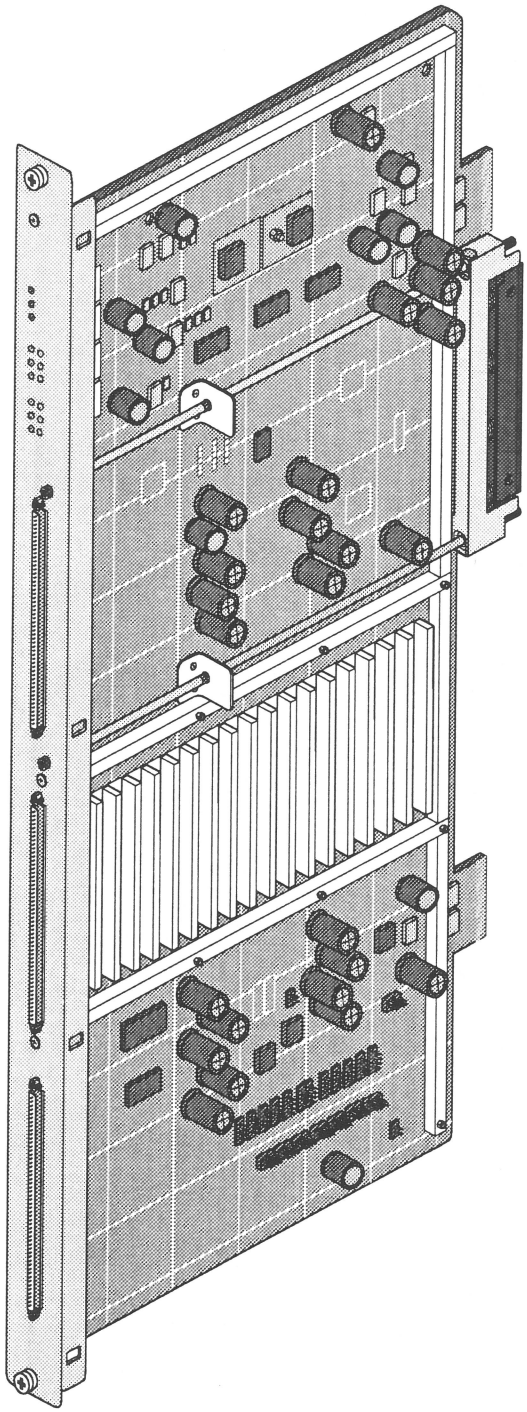
Slots are numbered from top to bottom, left to right, as viewed from the front of the midplane.



Student Notes

3-21 FOIL: Router

A module can contain a maximum of two routers. The router has a **crossbar** switch as its main component, which switches a node and connects to another node or to another router.



Student Notes

3-22 FOIL: Router (continued)

The **crossbar** switch allows all six of the standard router's ports to operate simultaneously at **full-duplex**.

There are **four** types of router boards.

- **Null Router**
  - The Null Router connects two node boards in the same deskside enclosure (up to four R10000 processors).
  - The Null Router configuration **cannot be expanded**.
- **Standard Router**

  The Standard Router can cable between 2 and 32 node boards together (from 2 to 64 CPUs) that are physically located in 1 to 8 enclosures.
- **Star Router**
  - The Star Router interconnects up to **four node boards**.
  - Three- and 4-node configurations are created using a **standard** Router with a **Star** Router.
- **CRAY Router**

  The CRAY router is used in conjunction with standard routers to expand the system from 33 to 64 node boards (65 to 128 CPUs) in a hierarchical hypercube.

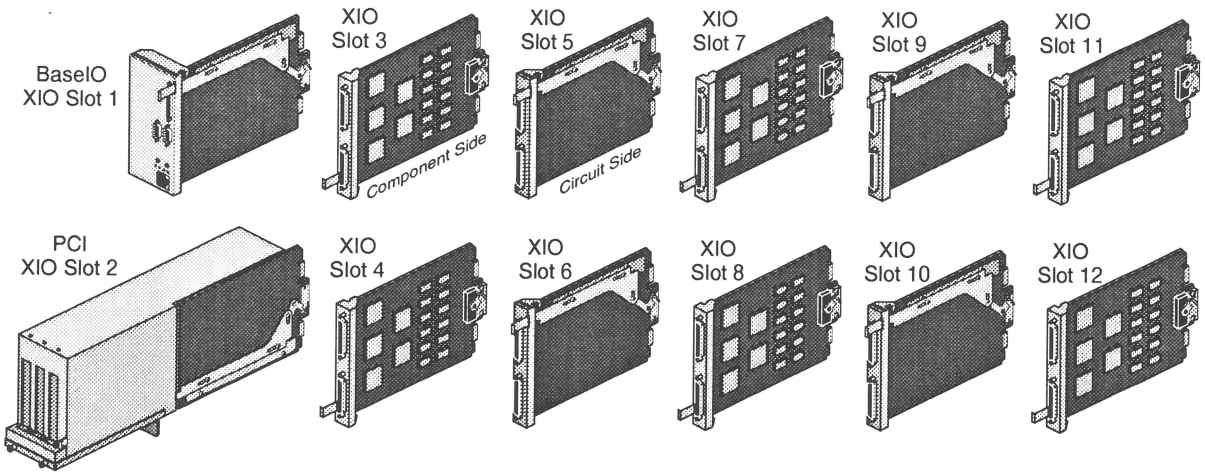
Student Notes

3-23 FOIL: XIO Boards

Origin2000 systems use an advanced input/output subsystem, which consists of a number of high-speed XIO links. XIO supports a wide range of SGI and third-party I/O devices.

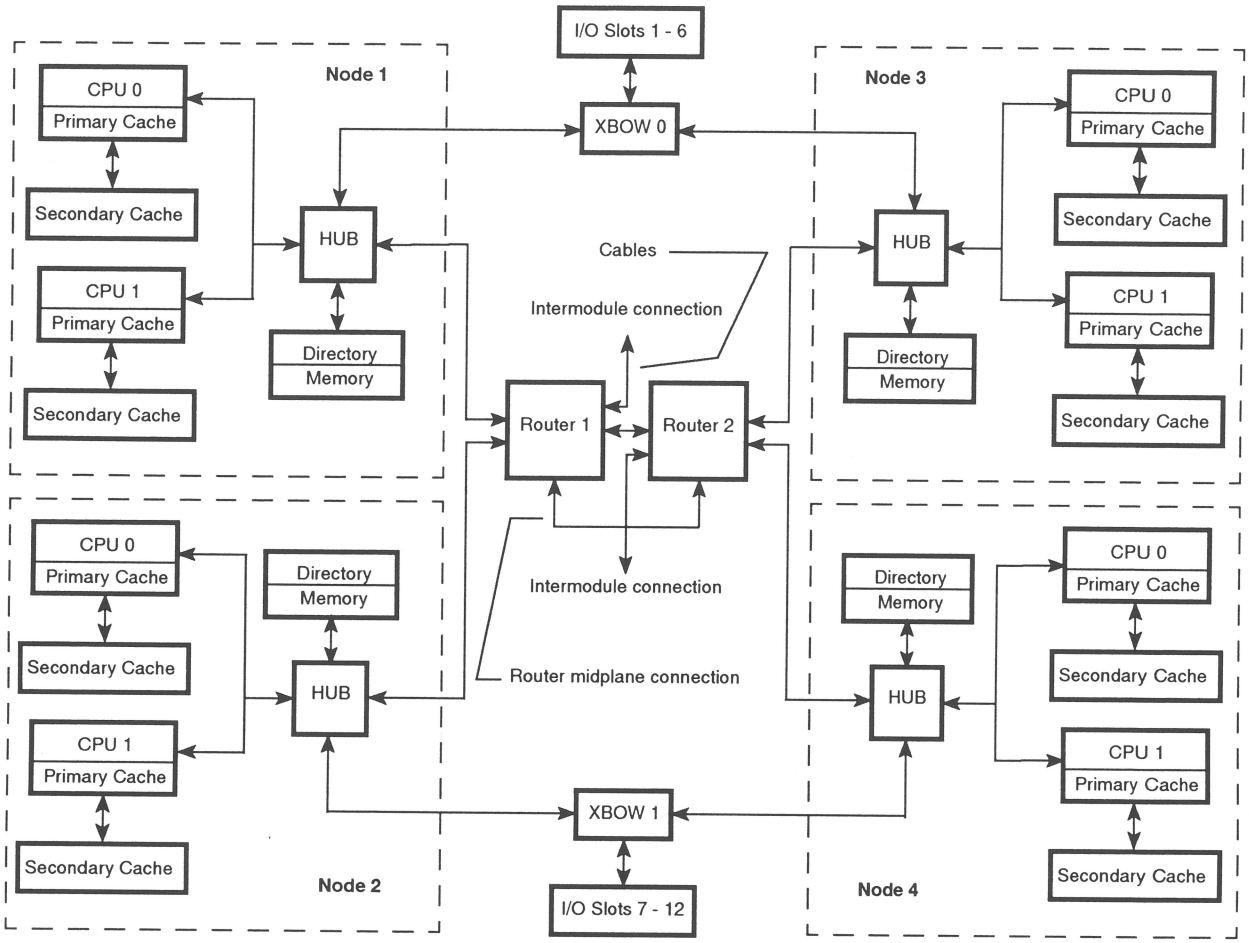
The internal I/O interfaces originate in each HUB and terminate on the targeted I/O controller. XIO uses the same physical link technology as CrayLink Interconnect, but it uses a protocol that has been optimized for I/O traffic. The XBOW ASIC is a crossbar routing chip that is responsible for connecting a maximum of two nodes to six I/O slots.

XIO boards convert the XBOW's XIO protocol to PCI protocol.



Student Notes

3-24 FOIL: Block Diagram



Module Block Diagram

The preceding diagram represents one module that is populated with four nodes.

- Any node can communicate with any other node in the module through the internal routers.
- Any node can communicate with other nodes in another module through an internal router that is cabled to an external router.

Student Notes

3-25 FOIL: Connections

Node/Router connections:

- Nodes 1 and 2 connect to router 1 through midplane connections.
- Nodes 3 and 4 connect to router 2 through midplane connections.

Node/XBOW connections:

- Nodes 1 and 3 connect to XBOW 1.
- Nodes 2 and 4 connect to XBOW 2.

XBOW connections:

- XBOW 1 connects to XIO slots 1 through 6.
- XBOW 2 connects to XIO slots 7 through 12.

Router/Router connections:

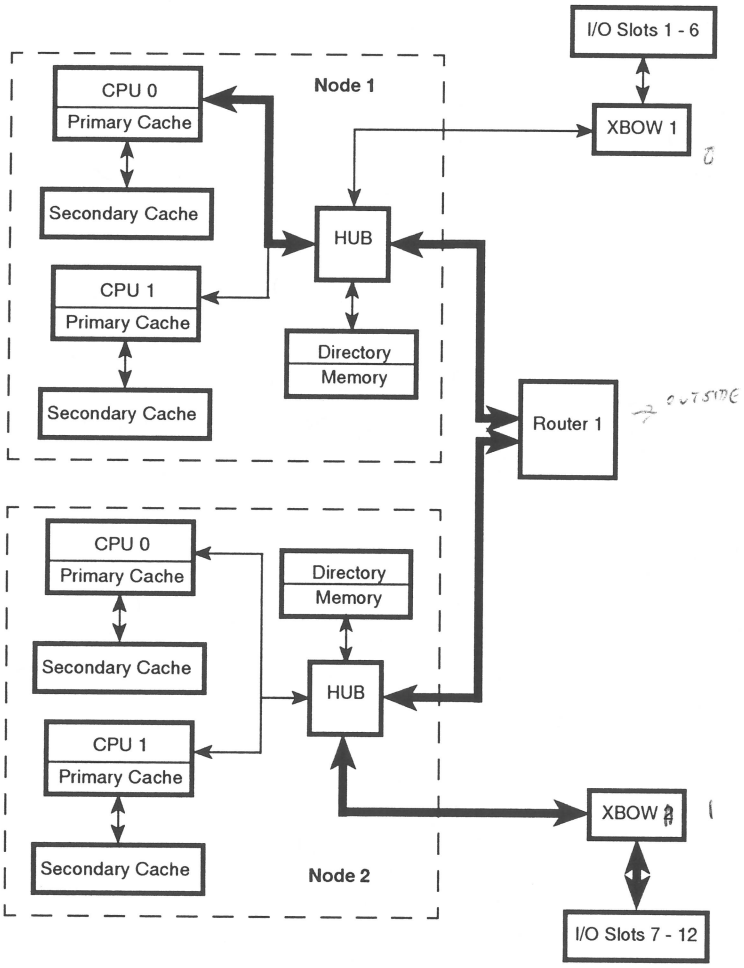
- Router 1 connects to router 2 through a midplane connection and possibly through a router-to-router cable.
- If the internal router-to-router cable is not used, there are three cables for connections to external routers; otherwise, there are two cable connections available.

Student Notes

3-26 FOIL: Indirect Connections

In some cases there may not be a direct connection. For example, node 1 connects to XBOW 1, which connects to XIO slots 1 through 6. What if node 1 needs a connection to XIO slot 7?

- Because nodes 2 and 4 connect to XBOW 2 (which connects to XIO slots 7 through 12), the node 1 connection to the XIO slot must go through one of these nodes.
- The operating system sets up router tables at boot-up time, so the node-to-node-to-XBOW route is predetermined.
- Therefore, the following path may be possible:
  - node 1 HUB to router 1
  - router 1 to node 2 HUB
  - node 2 HUB to XBOW 2
  - XBOW 2 to XIO slot 7

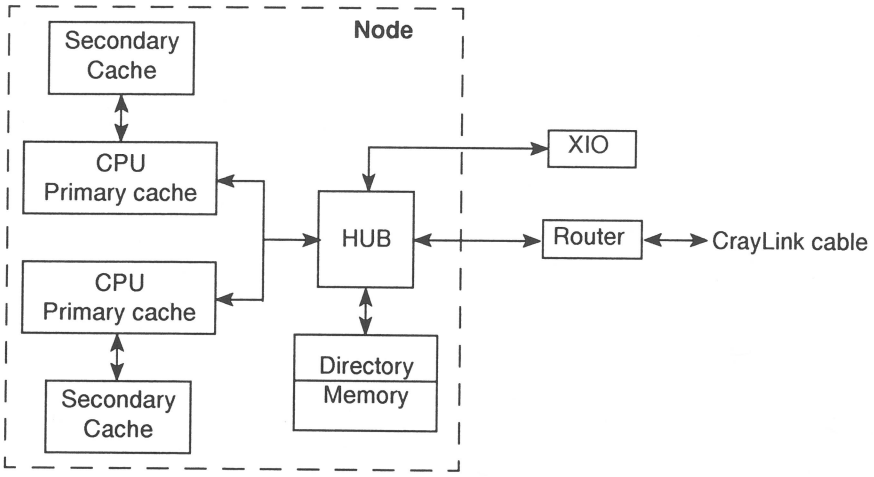


Student Notes

3-27 FOIL: Data Integrity

Data integrity must be maintained between the various components that compose the system.

For example, data that travels between the CPUs on the node and the local memory is checked for accuracy and is, in some cases, correctable. Data between the node and the XBOW ASIC, between the node and the router, and between routers is checked for accuracy and is retransmitted if an error is detected.



CPU to Primary Cache

Data that is sent to and from the **CPU** and **primary cache** is checked by **parity.** There is no error correction.

CPU to Secondary Cache

Data that is sent to and from the **CPU** and **secondary cache** is protected by **ECC** (SECDED). Single-bit errors are correctable and multiple- bit errors are detectable.

HUB to Memory

Data that is sent to and from **memory** is protected by **ECC** (SECDED). Single- bit errors are correctable and multiple-bit errors are detectable.

Student Notes

3-28 FOIL: Data Integrity (continued)

HUB to CPU

HUB data and address that is sent to and from the CPU is protected by **ECC** (SECDED). Single-bit errors are correctable and multiple-bit errors are detectable. System commands from the HUB are checked by parity.

HUB to Router

HUB to router data is protected by the **link-level protocol** that all ASIC-to-ASIC data paths use.

HUB to XBOW

HUB-to-XBOW data is protected by the **link-level protocol** that all ASIC-to-ASIC data paths use.

Router to Router

Router-to-router data is protected by the **link-level protocol** that all ASIC-to-ASIC data paths use.

Link-Level Protocol (LLP)

All ASIC-to-ASIC data transfers are done through source synchronous drivers/receivers (SSD/SSR). These SSD/SSRs create and interpret the high- speed, source synchronous signals that enable communication between ASICs within the module enclosure. They convert 64 bits of data transmitted at the core frequency of 100 MHz to 16 bits of data transmitted at 400 MHz.

The LLP interfaces with the SSD/SSR and provides error-free transmission of data between chips. It contains a synchronizer to interface directly with the core of the ASIC.

Error detection is made by using a **cyclic redundancy code** (**CRC**), and correction is made by retransmission. Both 8- and 16-bit links are supported.

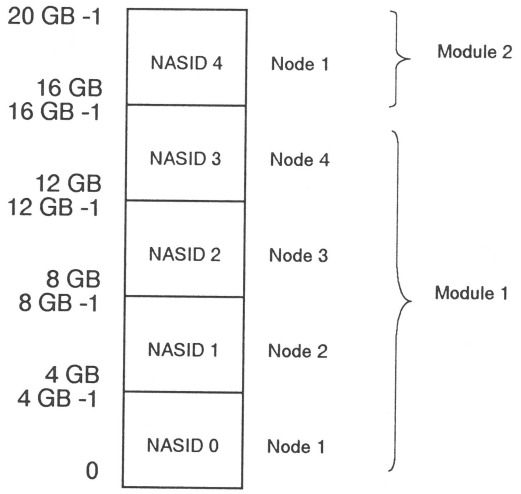
Student Notes

3-29 FOIL: Memory Organization

Distributed Shared Address Space

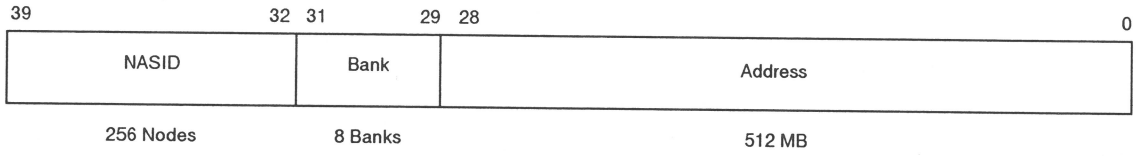
Origin2000 memory is located in a single shared address space of 4 GB per node, whether the node is fully populated or not. The more nodes in the system, the larger the memory space.

By distributing the Origin2000 memory among processors, memory latency is reduced: accessing memory close to a processor takes less time than accessing remote memory. Although physically distributed, main memory is available to all nodes.



Each node slot in the system is assigned a **NUMA** Address Space Identifier (**NASID**) whether the slot is populated or not, starting with NASID 0 for node slot 1.

The memory address includes the node number, the bank number, and the memory address within the bank.



Student Notes

3-30 FOIL: Cache Coherence

While data exists in either local or remote memory, copies may exist in various processor caches. Keeping these copies consistent is the responsibility of the logic of the various HUB ASICs. This logic is collectively referred to as **cache-coherence protocol**.

Memory Blocks, Lines, and Pages.

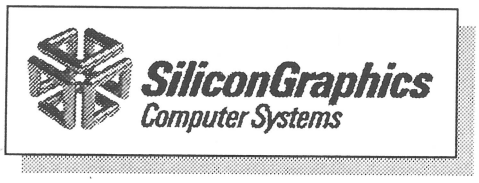
- The smallest unit of data that can be present in memory is broadly referred to as a **block**; a block is the **smallest unit of data** over which **coherence is maintained**.
- Block of data in a **cache**.
  - Called a line or **cache line**.
  - In an Origin2000 system, secondary cache lines are fixed in size at 32 words, or **128 bytes**.
- Block of data in **main memory**.
  - Called a **page**.
  - The main memory page sizes are **multiples of 4 KB**, usually **16 KB**.

The Origin2000 system maintains coherence via a directory-based coherence protocol. In order to implement the directory in a cost-effective manner for small systems and still retain the ability to scale to large systems, two different directory widths are supported (limited and full).

The limited directory supports up to 32 processors using a 16-bit wide directory bus. The full directory supports up to 1,024 processors using a 48-bit directory bus. Each word in the directory entry is protected by a SECDED ECC code.

The memory/directory (MD) interface of the HUB ASIC handles all the cache-coherence operations for its processors' memory references.

Please refer to *Origin and Onyx2 Theory of Operations Manual (007-3439-001)* for further information.



Module 4: System Components

Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

4-2 FOIL: Module Objectives

- Name and briefly describe the function of each of the eight major components that make up the Origin2000 module.
- Name and briefly describe the function of each of the three major components on the node board.
- List the four ports on the HUB ASIC and what they are connected to.
- Describe the memory banking used on the node board
- Name the component that maintains cache coherence in the Origin2000 memory subsystem.
- Determine which memory DIMMs and which directory DIMMs may be used in an expanded directory.
- Name the four router board types and describe their differences.
- List the four functions of the midplane.

Student Notes

4-3 FOIL: Module Topics

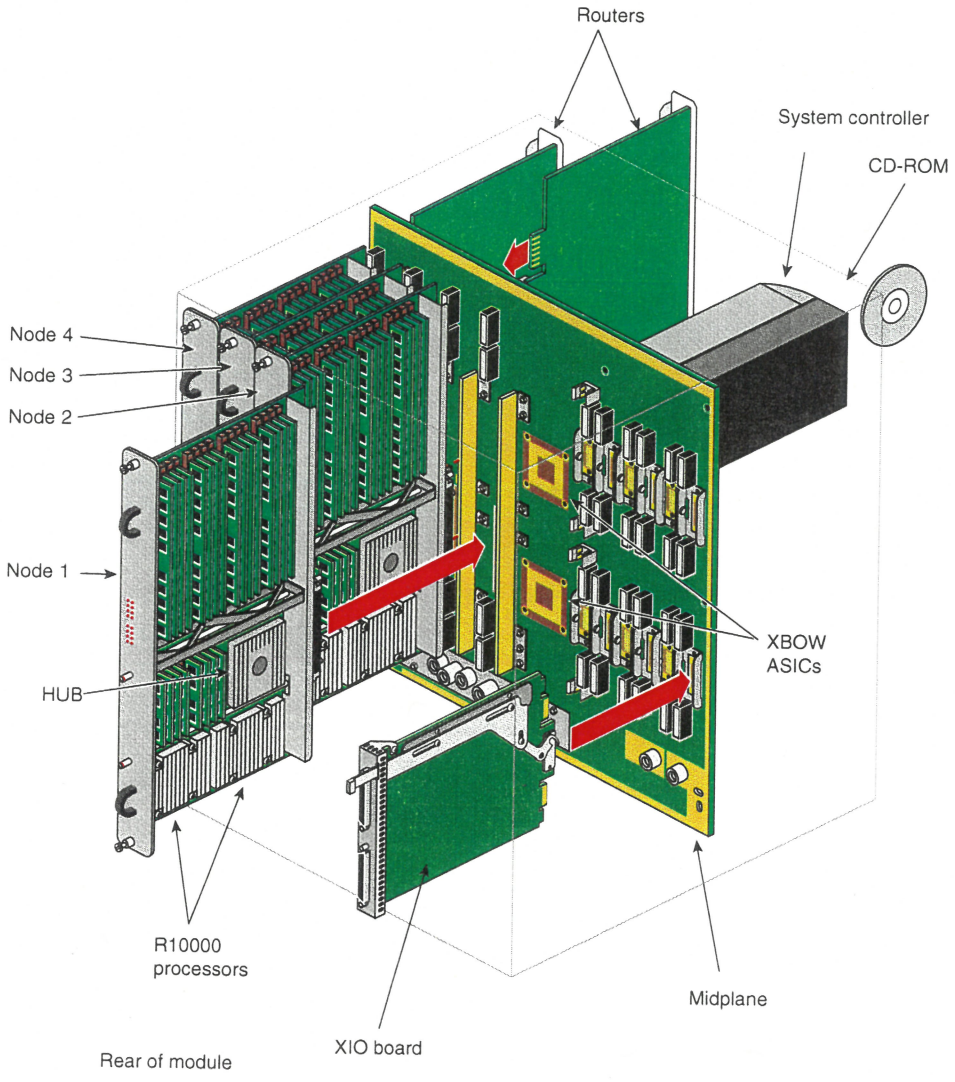
- Node board components.
  - Processors and cache.
  - Specifications.
  - Features.
- HUB ASIC.
  - Four ports.
  - Block transfer engine.
  - Error detection and correction.
- Memory.
  - Configurations.
  - Directory memory.
  - Error detection and correction.
- Router board.
  - Router types.
  - ASIC.
  - 6 ports.
- Midplane.
  - XBOW ASICs.
  - XIO slots.
  - Clock.
  - Power.
- Power and cooling.
  - Fan tray.
  - Power supply.

Student Notes

4-4 FOIL: Origin2000 Major Components

The Origin2000 module consists of eight major components.

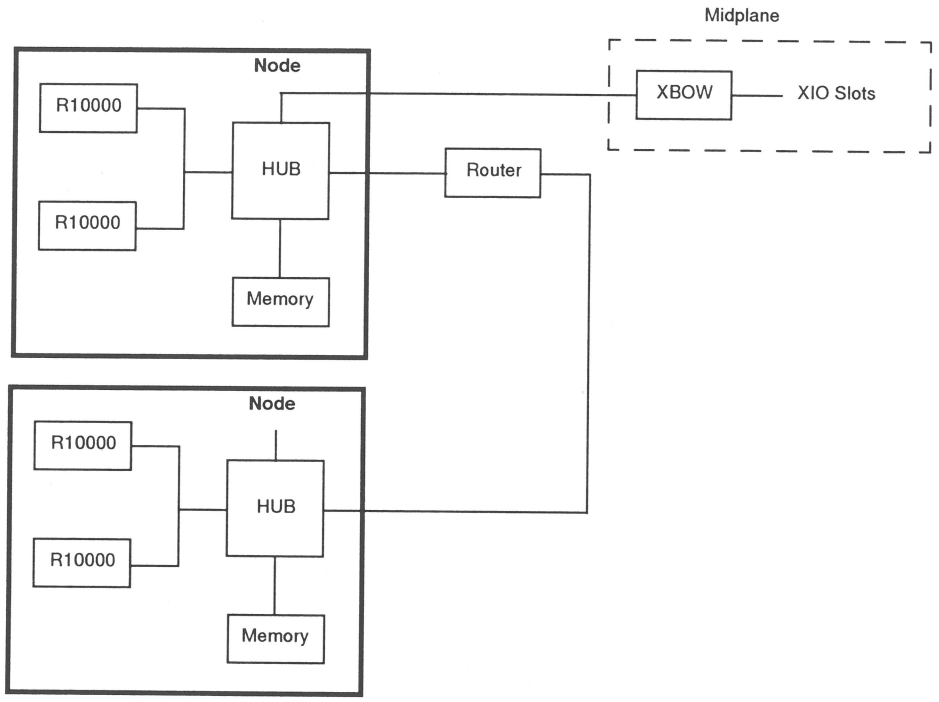
- Node board.
- Router board.
- Midplane.
- Module system controller.
- System disk.
- Power supply.
- Fan tray.
- Various XIO boards.



Student Notes

4-5 FOIL: SN0 Component Introduction

This module describes each of the main components, both electronic and those for power and cooling, that compose the Origin2000 server computer system.



Basically, **nodes** contain the processors and their associated caches, distributed memory, and the HUB ASIC, which controls all node communication.

Nodes communicate with other nodes through **routers**.

Nodes access I/O devices through **XBOW** ASICs that are mounted on the midplane.

The module is cooled by a fan tray that contains nine speed-controlled fans.

The module has its own power supply, which is mounted in the bottom of the module, that supplies power through the midplane.

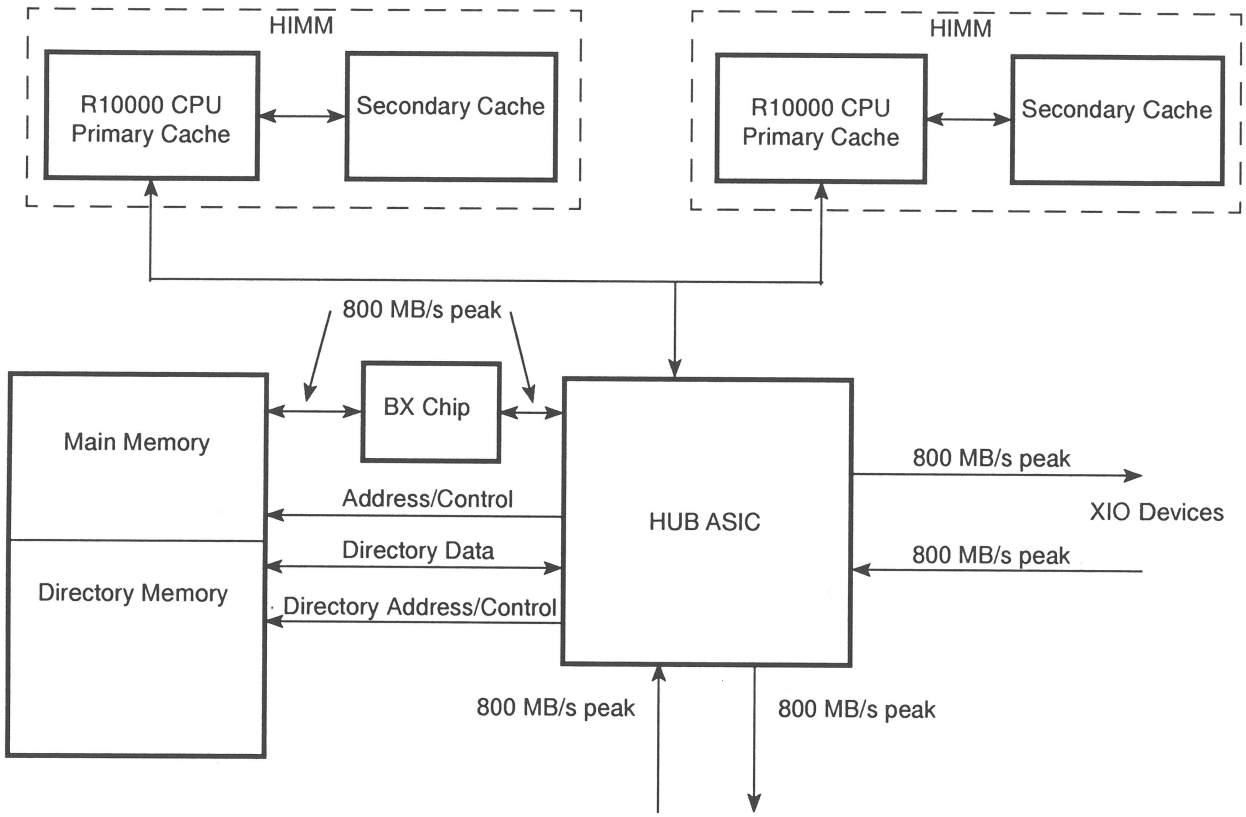
Student Notes

4-6 FOIL: Node Board

The basic building block of an Origin system is the Node board, which plugs into the rear of the module.

The Node board contains :

- Up to two HIMMs, each of which contains.
  - One R10000 processor with 32 KB each of both instruction and data caches in the chip.
  - One or four MB of secondary cache.



Node Block Diagram

- The HUB ASIC with ports to the processors, memory, I/O, and the CrayLink interconnect.
- Main and directory memory.
- BX chips.

Student Notes

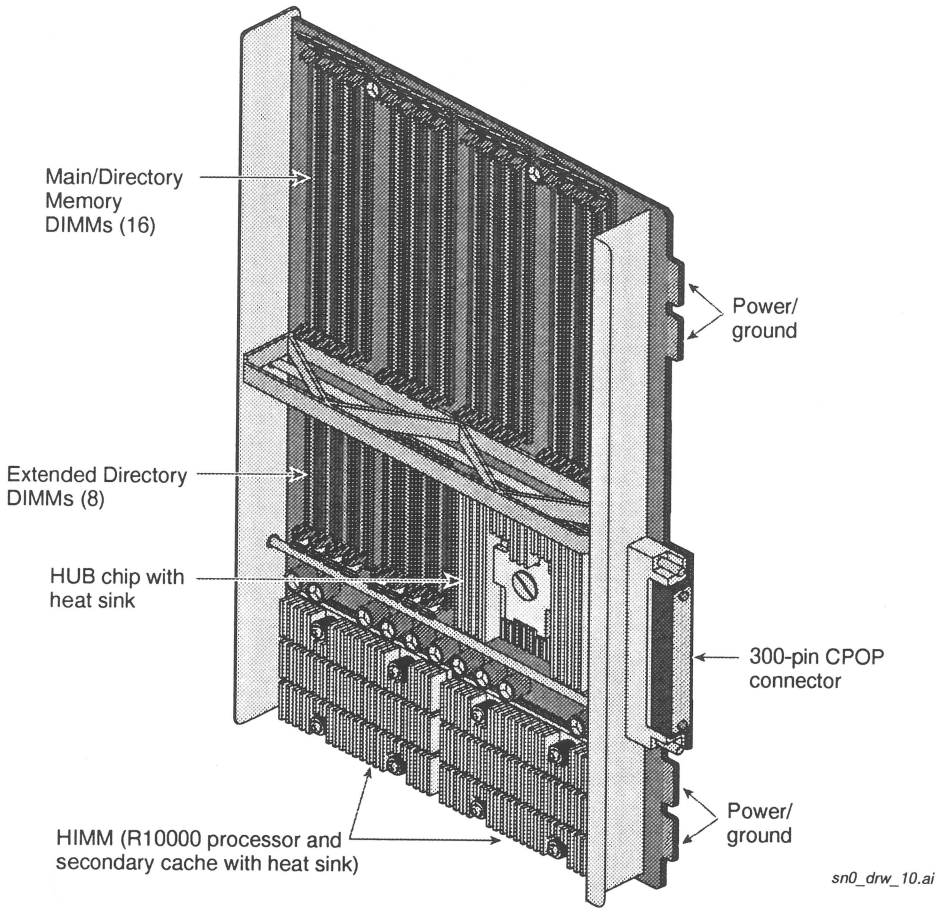
4-7 FOIL: Node Board Components

The **node board** is referred to as the **IP27** board. The node board components are

- One or two **R10000** (T5) processors.
  - Integral instruction and data caches of 32 KB each.
  - A speed of **195 MHz** or **180 MHz.**
  - Secondary cache of 1 or 4 MB per processor.
  - Processor and its associated secondary cache are mounted on a Horizontal Inline Memory Module (HIMM).
- One **HUB** ASIC.
  - Responsible for interconnecting all the parts of the node.
  - Has **4 ports**.
    - **Processor interface.**
    - **Memory interface.**
    - **Network interface.**
    - **I/O interface.**
- Main and Directory Memory.
  - Uses SDRAM parts.
  - Mounted on dual inline memory modules (DIMMs).
  - Running at 50 MHz supplying 800 Mbytes/s of raw data bandwidth.
  - Each node can support up to 4 Gbytes of main memory.
  - Directory memory is used for cache coherence.

Student Notes

4-8 FOIL: Node Board Components (continued)



- **R10000™** microprocessor.
  - 4-way super-scalar architecture that fetches and decodes four instructions per cycle
  - Out-of-order instruction execution
  - 5 separate execution units
  - 128-bit dedicated secondary cache data bus
  - 3.2 Gbytes/second peak data transfer rate to secondary cache
  - Connection to the HUB's Processor Interface (PI)

Student Notes

4-9 FOIL: HUB ASIC

Has four ports

- **Processor Interface (PI)**

  This port interfaces with the two R10000 processors.

- **Network Interface (NI)**

  This port interfaces with the CrayLink network that comprises the node-to-node interconnection fabric.

- **Memory/Directory Interface (MD)**

  This port interfaces with the memory directory, which maintains cache coherency, and with local main memory.

- **XIO Interface (II)**

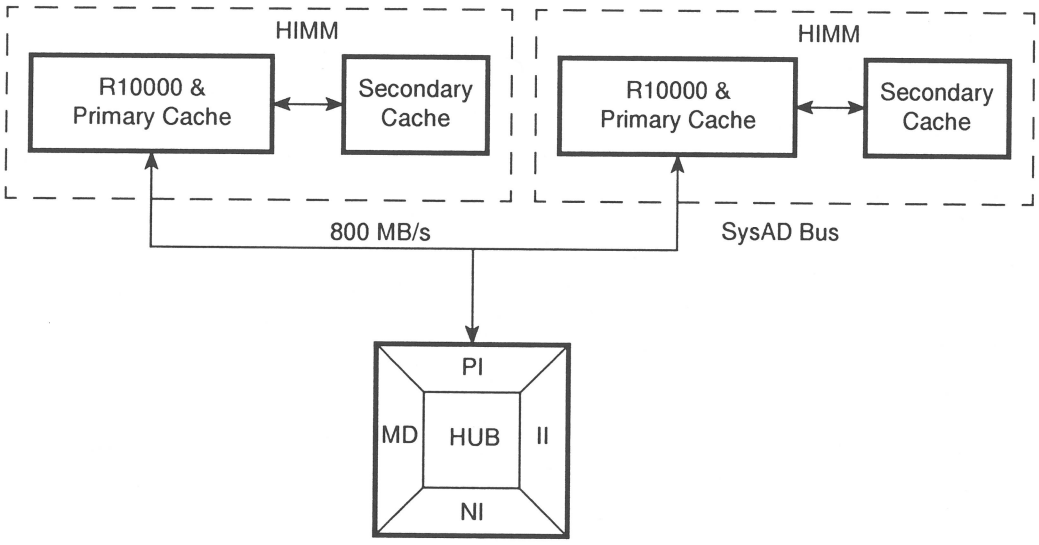
  This port interfaces with one of the two XBOW ASICS that are mounted on the midplane. This connection allows the node to communicate with the XIO devices that are installed in the module.

Student Notes

4-10 FOIL: Processor Interface (PI)

- The **PI** contains logic that interfaces directly with the processors and performs the following:
  - Arbitration.
  - Error handling and flow control of the system address/data command buses (Sys AD).
- The **PI** is designed to support up to two pricessors in the dedicated node.

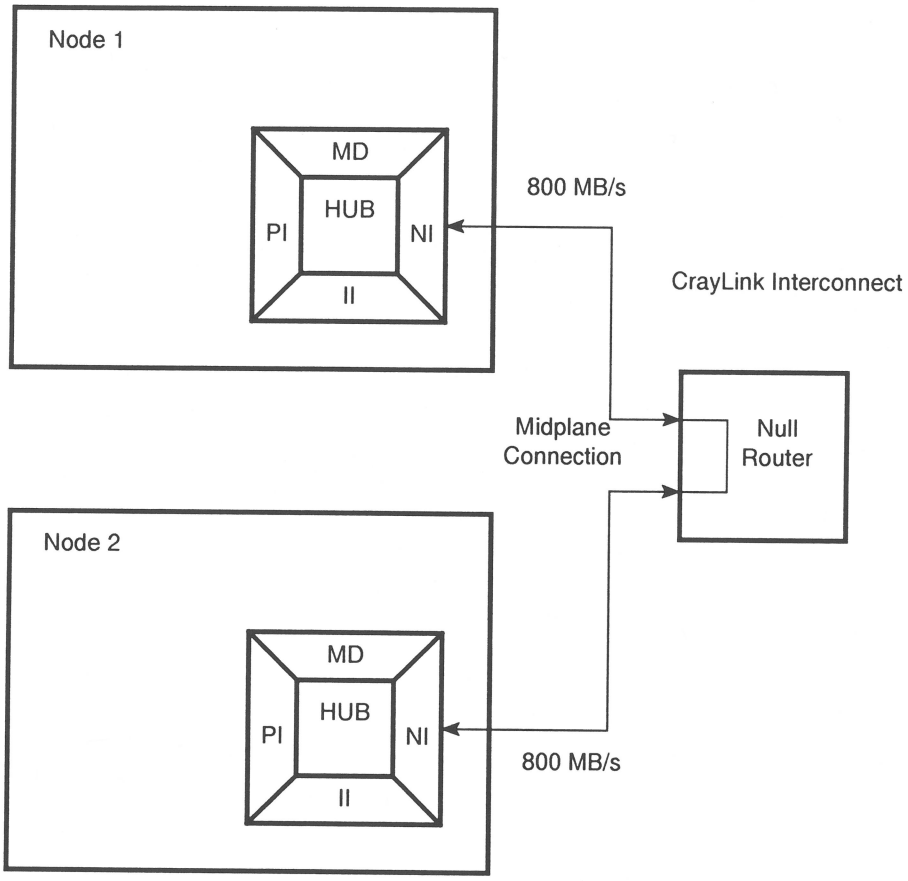
  This means that the processors send commands to and receive commands from the external agent (PI) only.



Student Notes

4-11 FOIL: Network Interface (NI)

- The network interface of the HUB chip connects the node to other nodes, whether through routers, or through a Null Router, shown below.
- Each network connection consists of a **pair** of **16-bit unidirectional links** that each provide **800 Mbytes/s** of raw bandwidth.



2 Node (4 CPU) Configuration

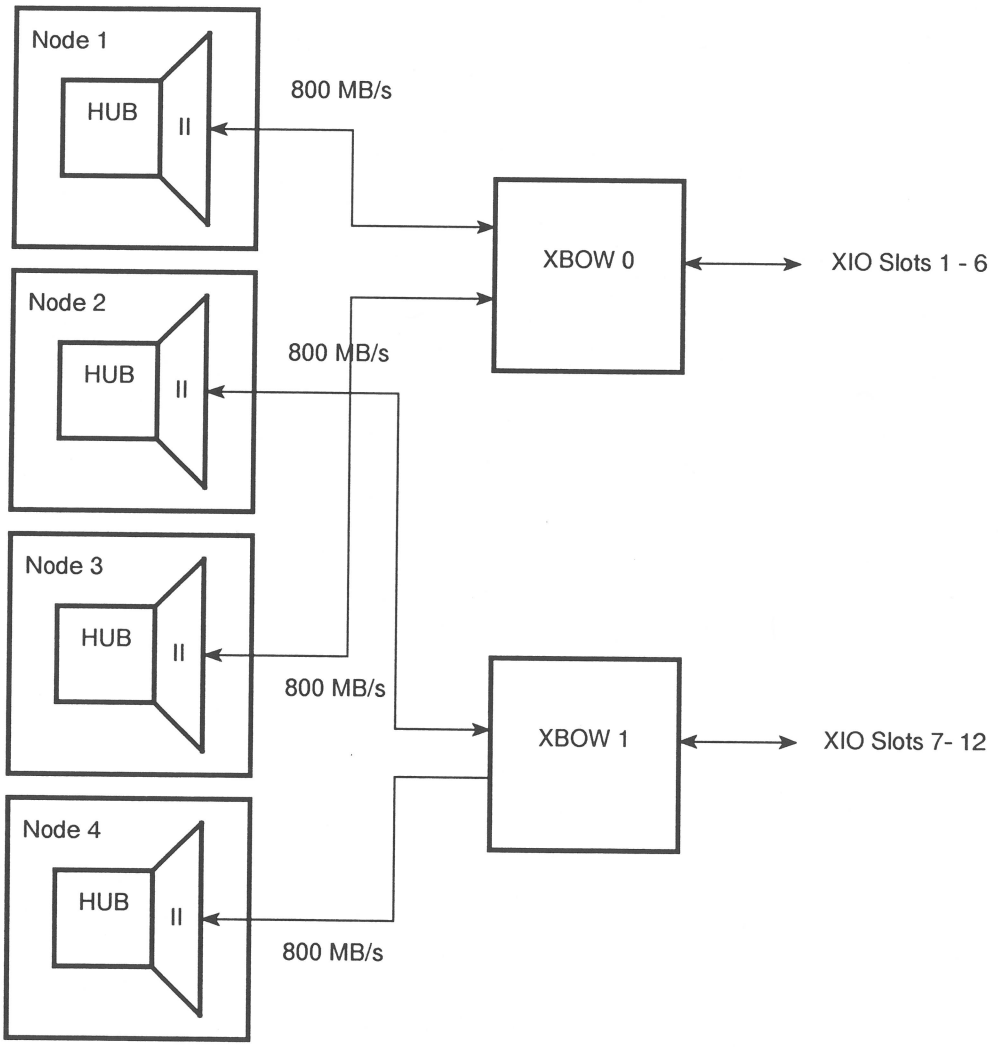
- The links use a low-swing CMOS signalling technology (STL) that SGI developed to allow direct very high-speed ASIC to ASIC intra-cabinet links.
- The network links can be buffered with differential ECL buffers to drive signals on 3-meter cables between Origin cabinets.

Student Notes

4-12 FOIL: XIO Interface (II)

- Each Node has an attachment port to the I/O subsystem.
- This port uses a **protocol** known as **Crosstalk** (XTALK) and attaches to an intelligent crossbar switch known as the XBOW.
- There are two **XBOWs** mounted on the midplane of the Origin module.
- Devices that connect to a XBOW are known as **Widgets** or **XIO** devices.

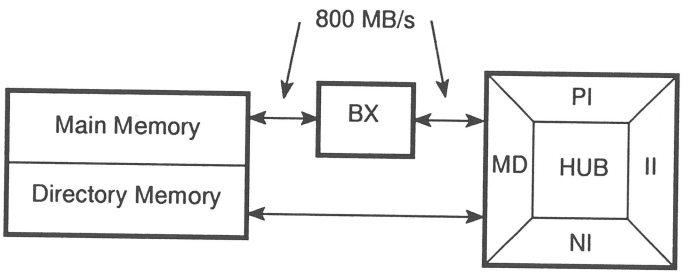
  Examples of XTALK devices are Nodes, graphics subsystems, high- speed network controllers, video options, disk controllers, and Base I/O boards.
- The II port has an **800-Mbyte/s** input and output raw peak bandwidth.
- The XBOW port connects to **six I/O slots on the midplane**.



Student Notes

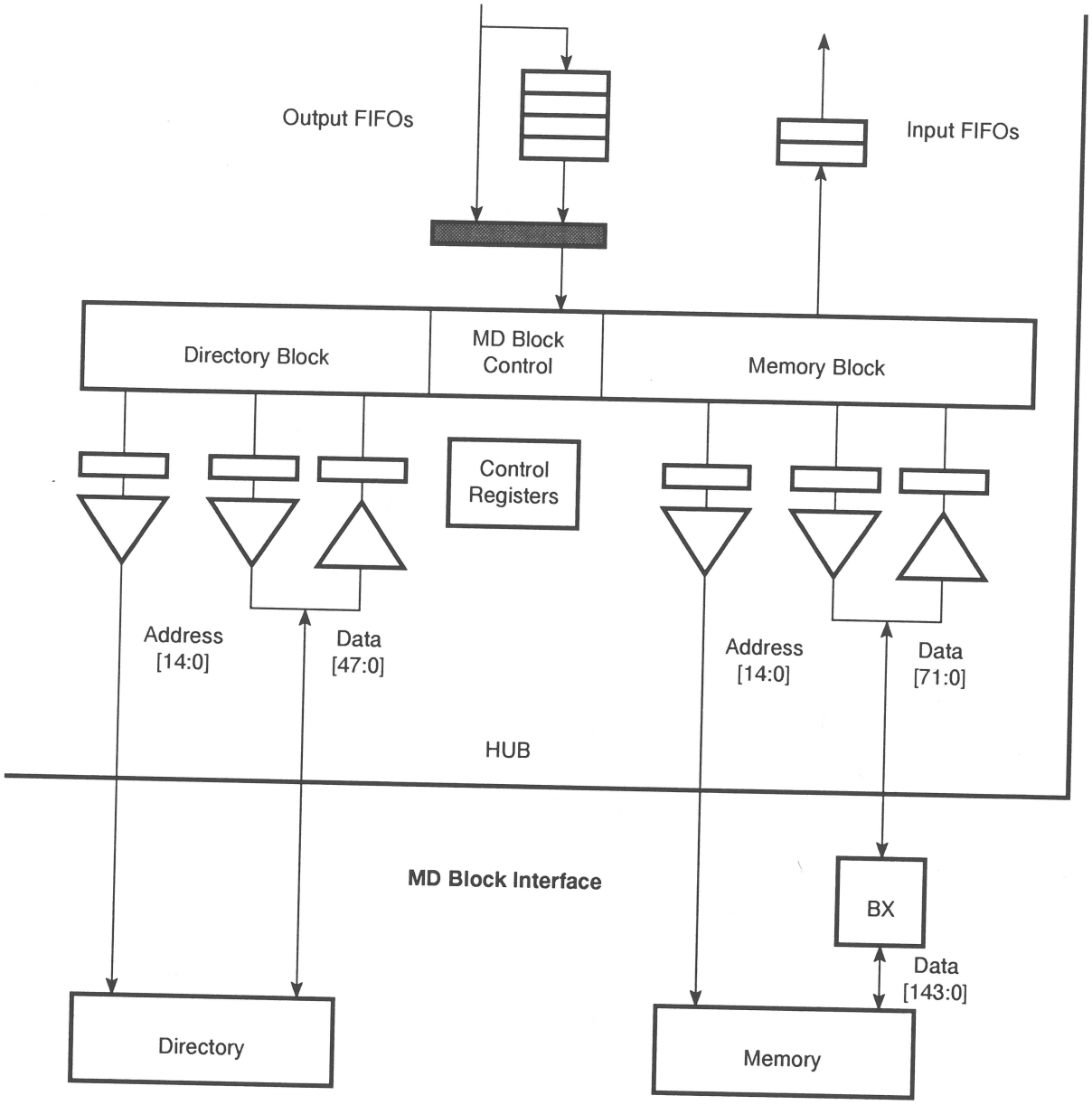
4-13 FOIL: Memory/Directory Interface (MD)

- The MD block of the HUB chip controls the interface to the **memory** and **directory DIMMs**.
- Raw bandwidth to memory is **800 Mbytes/s.**
- The HUB interfaces to **main memory** through a bus exchanger chip (**BX**).
- The HUB interfaces directly to **directory memory.**
- The HUB is also responsible for **ECC generation and checking** for both the directory and memory.
- The **HUB maintains coherence** by using the directory memory.
- Block transfers:
  - The block transfer engine (BTE) is a hardware copy device that is located in each HUB chip in the Origin series computer systems.
  - The kernel can do block transfers for page migration and replication.
  - The BTE is designed to transfer one or more cache lines of data from a double-aligned source address to a cache-aligned destination address after being given the source and destination address, and length.
- The HUB has a small cache to enhance the performance of the fetch-and-op synchronization primitives.
- There are four incoming and two outgoing FIFOs.



Student Notes

4-14 FOIL: Memory Block Diagram



Student Notes

4-15 FOIL: Memory

- Main memory is implemented using SDRAM parts that are mounted on **dual inline memory modules** (DIMMs) running at **50 MHz** and supplying **800 Mbytes/s** of raw data **bandwidth**.
- Main and directory memory data integrity are protected by a SECDEC **ECC**.
- Main memory is **cacheable** by all processors in the system.
- Memory capacities:
  - With 16-Mbit technology, each node can support up to 1 Gbyte of memory in the node in increments of 64 or 128 Mbytes.
  - With 64-Mbit technology, each node can support up to 4 Gbytes of memory in the node in increments of 512 Mbytes.
- There are **eight DIMM-pair slots per node**, which can be populated with a pair of DIMMs of any of three possible individual densities.
- Directory memory resides in the same DIMMs along with main memory for systems that include 16 nodes or fewer.
  - Systems with **more than 16 nodes require extended directory DIMMs**.
  - The extended directory DIMMs reside in eight additional DIMM slots.

Student Notes

4-16 FOIL: Memory Layout

- There are 16 slots for 8 DIMM bank pairs of main and directory memory and 8 additional slots for extended directory memory, if required.
  - The main memory DIMM slots may be partially populated.
  - **Bank 0 must be populated.** NOT ANYMORE
  - As few as one DIMM bank pair may be installed in the node.
  - **Each DIMM in the pair must be of the same type.**
  - DIMM bank pairs should be populated from the lowest to the highest slots. **Holes are allowed.**
  - Different DIMM bank pairs may be of different DIMM types.
  - Each DIMM bank pair requires 2 slots: a lower and an upper slot.

    For example: DIMM bank pair 0's slots are located at MMXL0 and MMYH0.
  - If there are more than 16 nodes, the corresponding extended directory slot must be populated with the same capacity DIMM, in this case, slot DIRX0.

The memory DIMMs contain both main memory and directory memory.

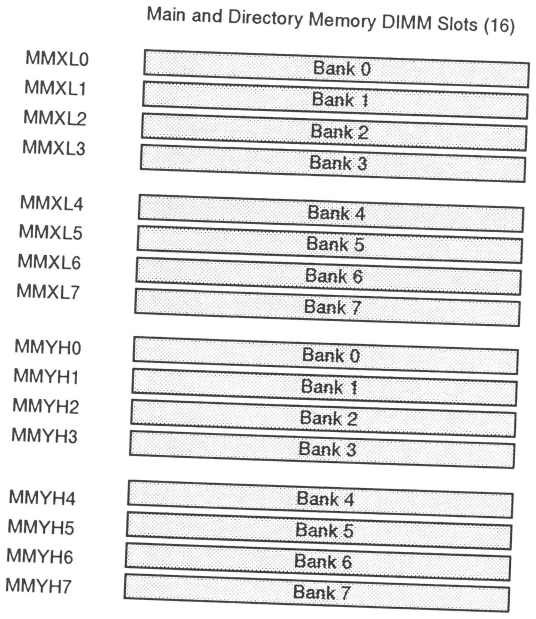
- The DIMM pair implements 144 bits of memory and 16 bits of directory.
  - 144 bits (128 bits of data and 16 bits of ECC)
  - 16 bits of directory (11 bits of data and 5 bits of ECC) for the limited directory

The extended directory DIMMs contain only directory entries. These DIMMs are an extension of the 16 directory bits that are contained on the memory DIMMs.

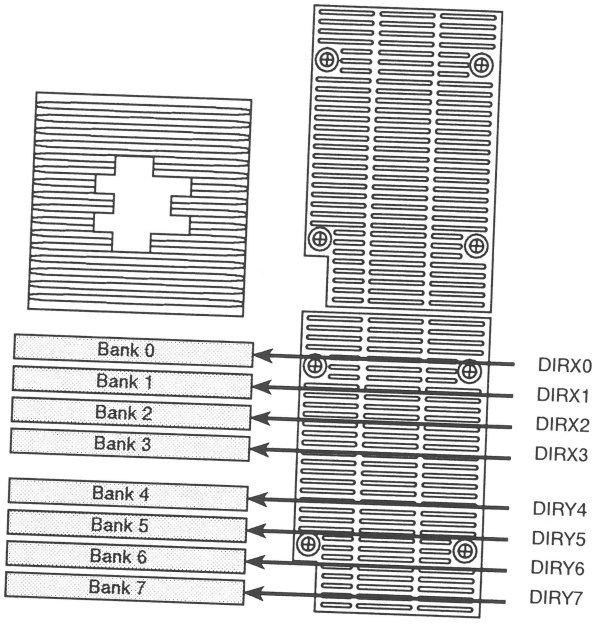
- The combination of the 16 directory bits on the memory DIMMs and the 32 bits on the extended directory DIMMs provide 48 bits (41 bits of data and 7 bits of ECC) for the full directory entry.
- Each DIMM pair contains 2 physical banks of SDRAM parts, with each DIMM physical bank containing 2 logical banks.

Student Notes

4-17 FOIL: Memory Layout (continued)



Node Board Bank Layout



Extended Directory DIMM Slots (8)

Student Notes

4-18 FOIL: Memory DIMMs

Main memory DIMMs

DIMM Type	Chip Depth	Chip Width	Number of Chips/ DIMM	Single DIMM Capacity	Total DIMM Capacity	SDRAM Technology
A Yellow	2 Mbit 1 Mbit	8 16	18 1	32 Mbytes	64 Mbytes to 512 Mbytes	16 Mbits
B Green	4 Mbit 2 Mbit	4 8	36 2	64 Mbytes	128 Mbytes to 1 Gbytes	16 Mbits
C Red	16 Mbit 8 Mbit	4 8	36 2	256 Mbytes	512 Mbytes to 4 Gbytes	64 Mbits

Extended Directory DIMMs

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

Nodes populated with **A or B type DIMMs** require that extended directory DIMMs be populated with **D type DIMMs.**

Nodes populated with **C type DIMMs** require that extended directory DIMMs be populated with **E type DIMMs.**

Main Memory and Extended Directory DIMM Configurations

Main Memory Pair Type	Extended Directory Type
A, A Yellow	D White
B, B Green	D White
C, C Red	E Blue

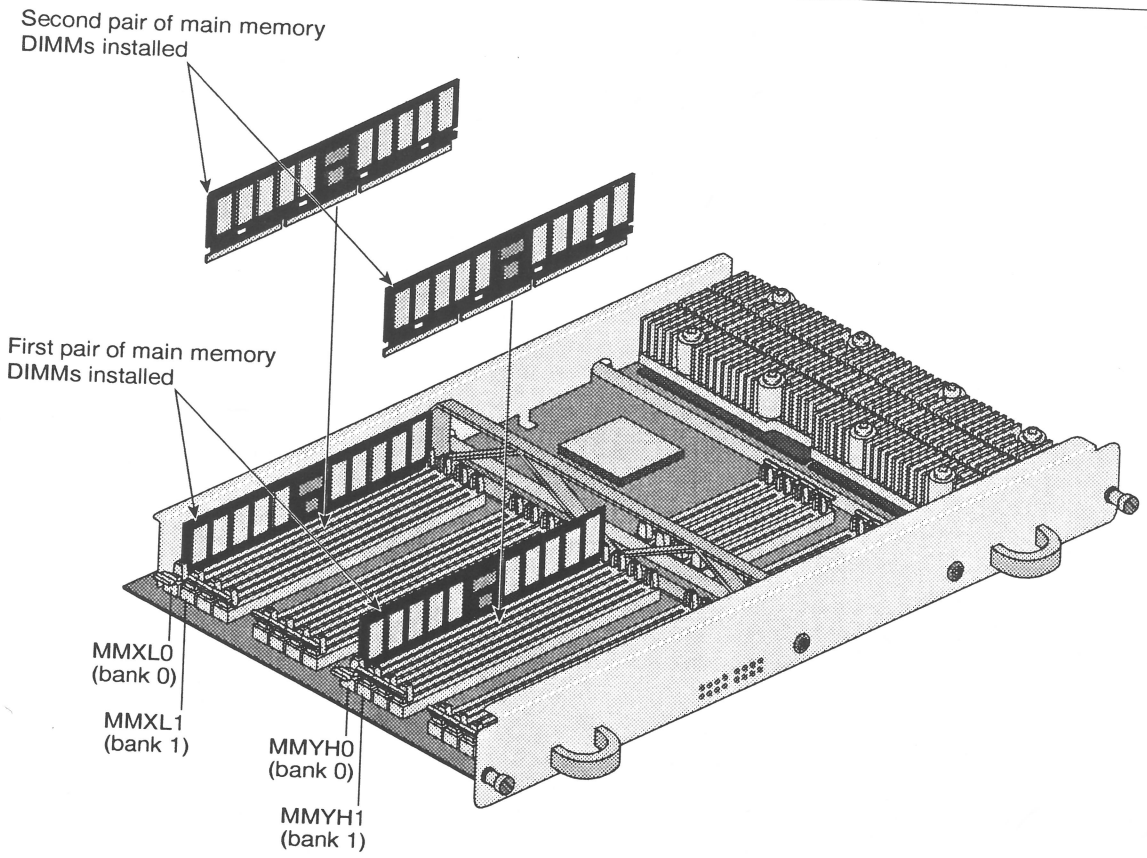
Student Notes

4-19 FOIL: DIMM Location Chart

DIMM Triplet	DIMM Locations	Bank Number
1st	MMXL0, MMYH0, DIRX0	Bank 0
2nd	MMXL1, MMYH1, DIRX1	Bank 1
3rd	MMXL2, MMYH2, DIRX2	Bank 2
4th	MMXL3, MMYH3, DIRX3	Bank 3
5th	MMXL4, MMYH4, DIRY4	Bank 4
6th	MMXL5, MMYH5, DIRY5	Bank 5
7th	MMXL6, MMYH6, DIRY6	Bank 6
8th	MMXL7, MMYH7, DIRY7	Bank 7

Student Notes

4-20 FOIL: DIMM Combination Examples



The following list includes examples of legal and illegal **main memory** configurations:

- **Example 1**

  MMXL0 Type A

  MMYH0 Type A

  MMXL1 Type C

  MMYH1 Type C

  MMXL3 Type B

  MMYH3 Type B

  All other bank slots are empty. The node count is less than or equal to 16 nodes.

  The above example is **LEGAL** because each of the slot pairs have the same type of DIMM. The hole between banks 1 and 3 is legal.

Student Notes

4-21 FOIL: DIMM Combination Examples (continued)

- **Example 2**

  MMXL0 Type C

  MMYH0 Type C

  All other bank slots are empty. The node count is less than or equal to 16 nodes.

  The above example is **legal** because the lowest bank is populated first.

- **Example 3**

  MMXL0 Type A

  MMYH0 Type B

  MMXL1 Type A

  MMYH1 Type B

  All other bank slots are empty. The node count is less than or equal to 16 nodes.

  The above example is **not legal** because the bank pair DIMMs are not of the same type.

- **Example 4**

  MMXL4 Type C

  MMYH4 Type C LEGAL

  All other bank slots are empty. The node count is less than or equal to 16 nodes.

  The above configuration is **not legal** because you must populate bank 0 first.

  NOT LEGAL

Student Notes

4-22 FOIL: DIMM Combination Examples (continued)

The following examples show legal and illegal **main and extended directory memory DIMM** configurations:

- **Example 5**

  MMXL0 Type A
  MMYH0 Type A

  MMXL1 Type C
  MMXH1 Type C

  MMXL2 Type B
  MMYH2 Type B

  DIRX0 Type D
  DIRX1 Type E
  DIRX2 Type D

  All other bank slots are empty. The node count is greater than 16 nodes.

  The above configuration is **legal** because the bank pairs are of the same type, and the corresponding extended directory slots are of the same type.

- **Example 6**

  MMXL0 Type A

  MMYH0 Type A

  DIRX0 Type E

  All other bank slots are empty. The node count is greater than 16 nodes.

  The above configuration is **not legal** because the main memory DIMM types do not match the extended directory DIMM type.

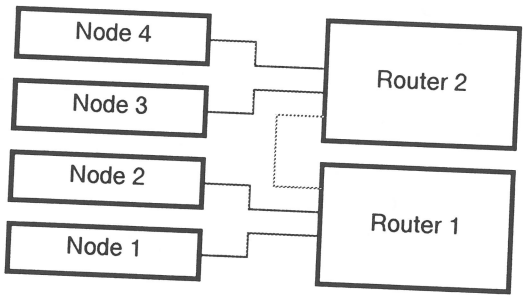
Student Notes

4-23 FOIL: Router Board Types

Four types of routers are used in various Origin2000 configurations.

- Null Router
- Standard Router
- Star Router
- Cray Router

In all cases except the Cray Router, node board 1 and node board 2 connect to router 1 through the midplane, and node board 3 and node board 4 connect to router 2. In addition, router board 1 connects to router board 2 through the midplane, except when using the Null Router, in which case there is no router-to-router connection.

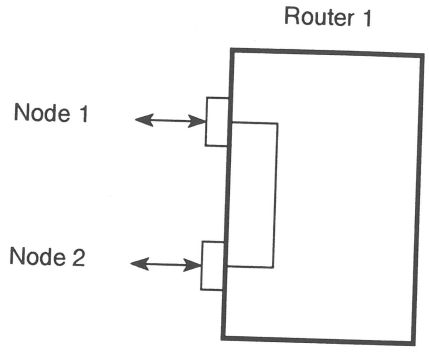


Midplane Node-to-router and Router-to-router Connections

Student Notes

4-24 FOIL: Null Router

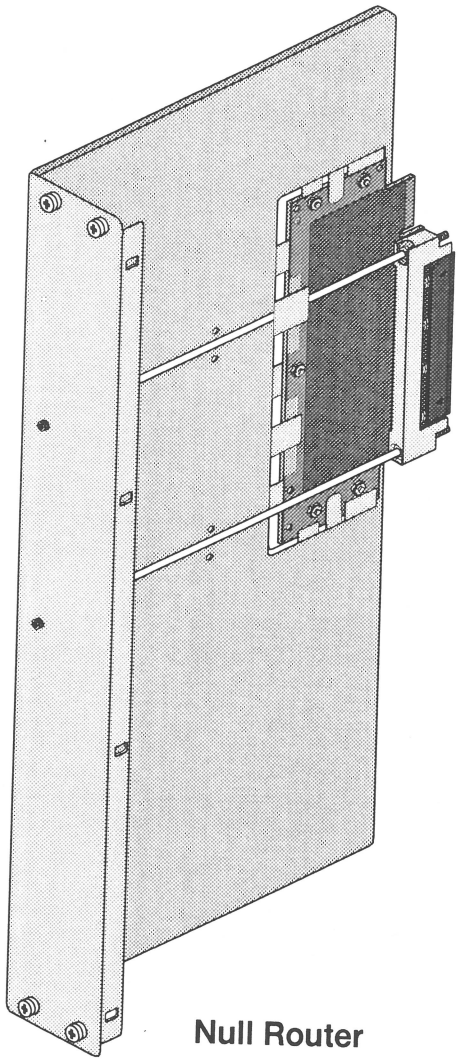
The Null Router is not really a router; rather it is a set of wires mounted on a board that jumpers one node board to the adjacent node board. The Null Router occupies the Router 1 slot, connecting Node 1 to Node 2.



Null Router
In a 2-node Configuration

Student Notes

4-25 FOIL: Null Router (continued)



Null Router

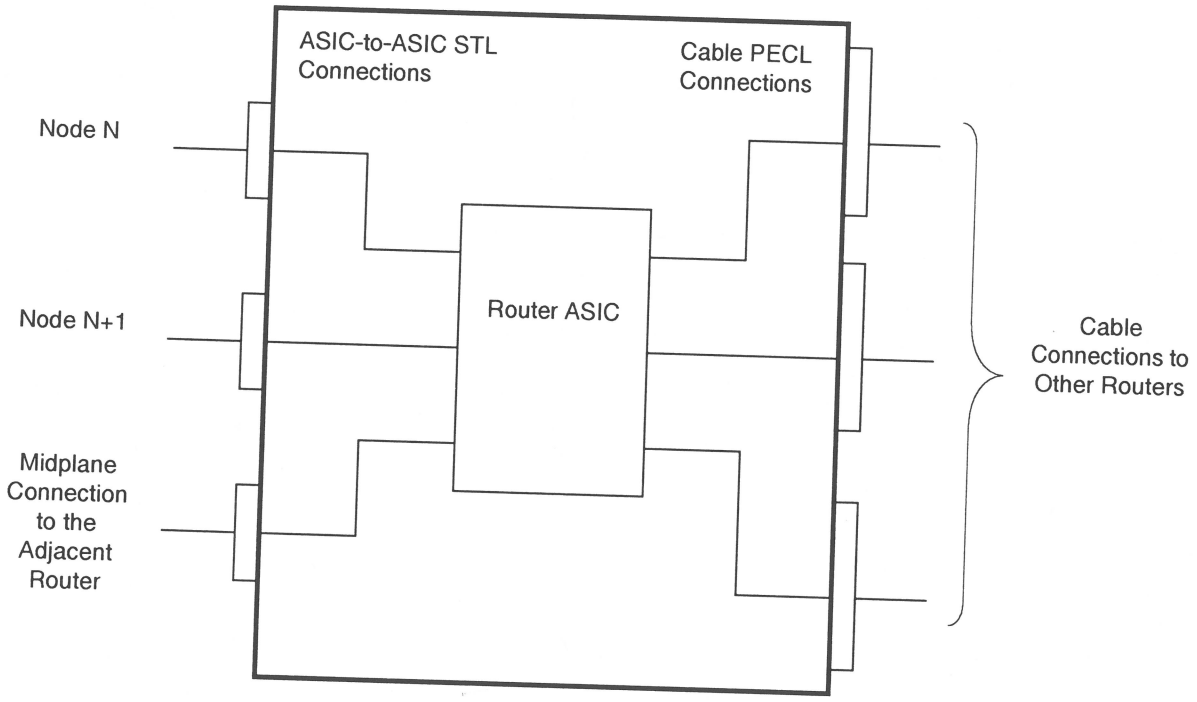
Student Notes

4-26 FOIL: Standard Router

As many as two standard Routers mount in the front of the deskside module.

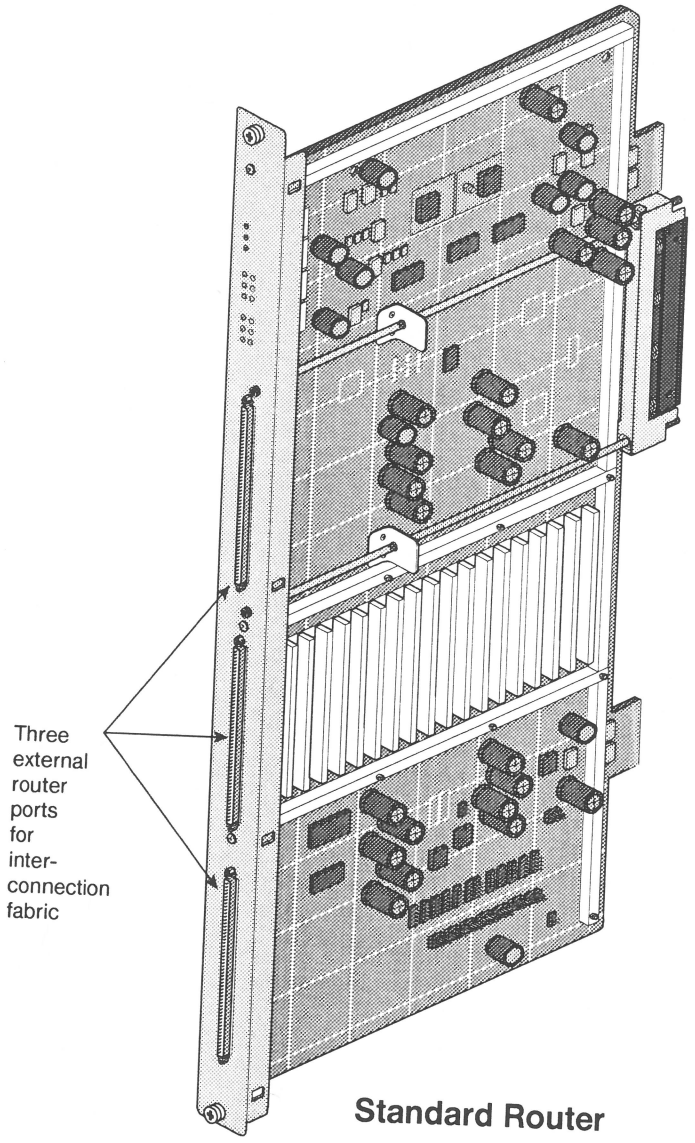
The Standard Router's main component is the router ASIC, which is an intelligent crossbar switch with 6 ports.

- Three ports are Silicon Graphics Transistor Logic (STL) that are used for ASIC-to-ASIC (through the midplane) connections.
- The other three ports are Positive Emitter Coupled Logic (PECL) that are used on the three cable ports on the outside edge of the board. These connections connect to other routers.
- The router ASIC also contains routing tables that are filled at boot-up.



Student Notes

4-27 FOIL: Standard Router (continued)



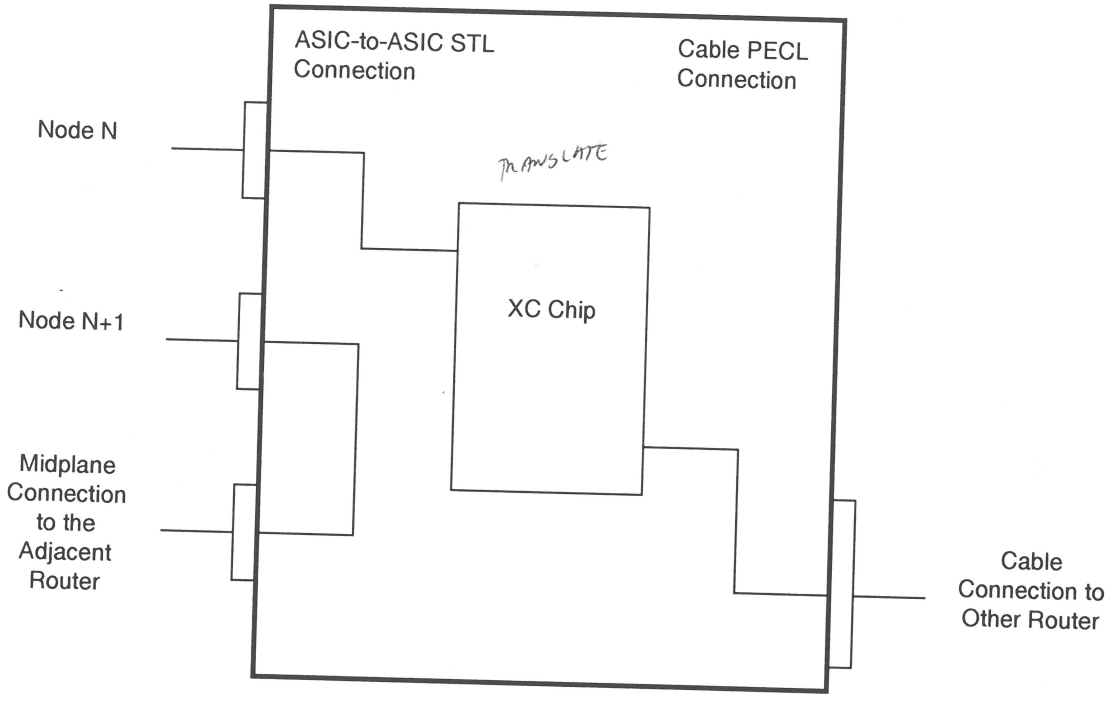
Standard Router

Student Notes

4-28 FOIL: Star Router

The Star Router is used only in conjunction with a Standard Router in a 3- or 4-node module.

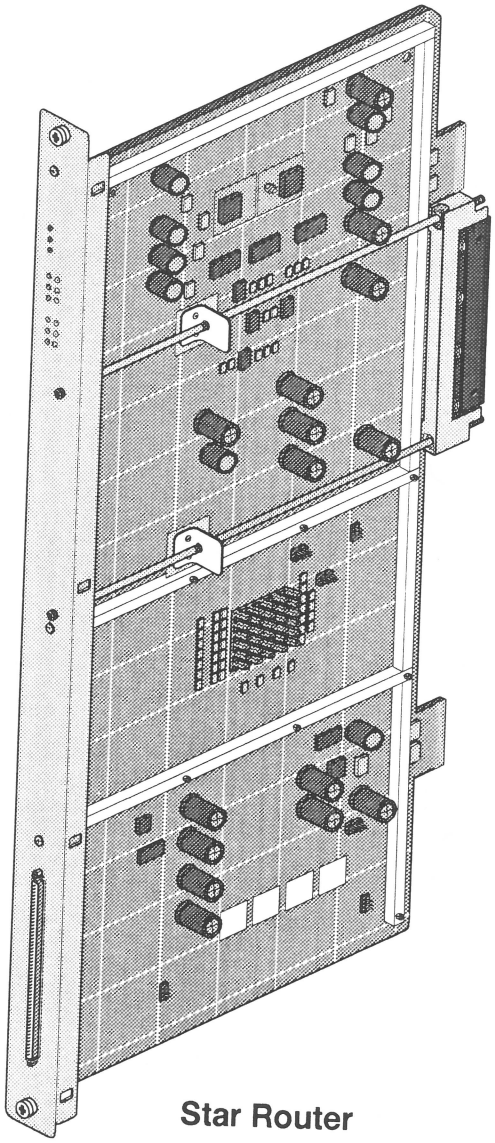
- The Star Router has no router ASIC, but it does include an XC chip that converts PECL to STL.
- There is no crossbar switch in the Star Router.



- ONLY USED IN DESKSIDE

Student Notes

4-29 FOIL: Star Router (continued)



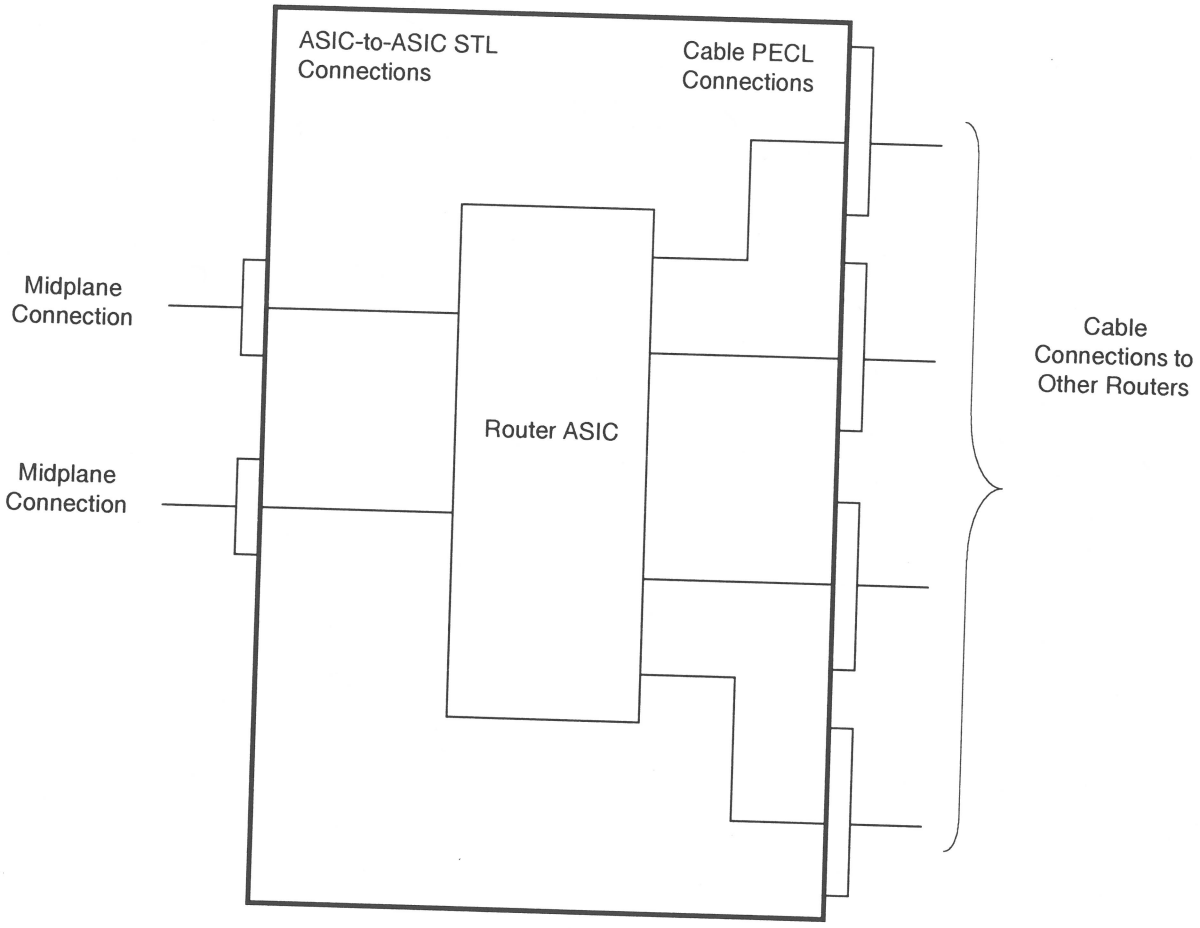
Star Router

Student Notes

4-30 FOIL: Cray Router

The Cray Router is used only in the second-level routing that is required in systems that have more than 64 processors.

The Cray Router has a router ASIC with 4 PECL connections and 2 STL connections.

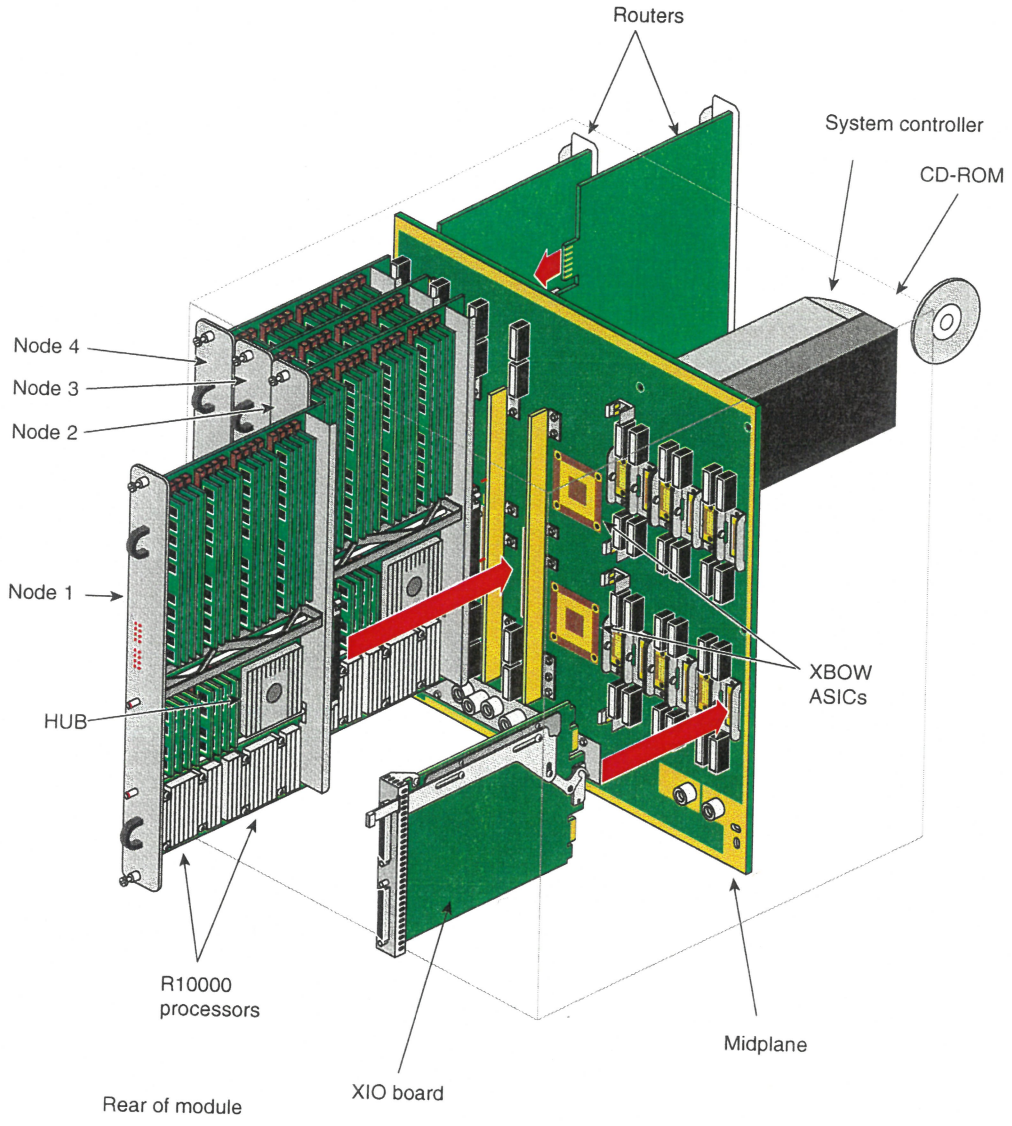


Student Notes

4-31 FOIL: Midplane

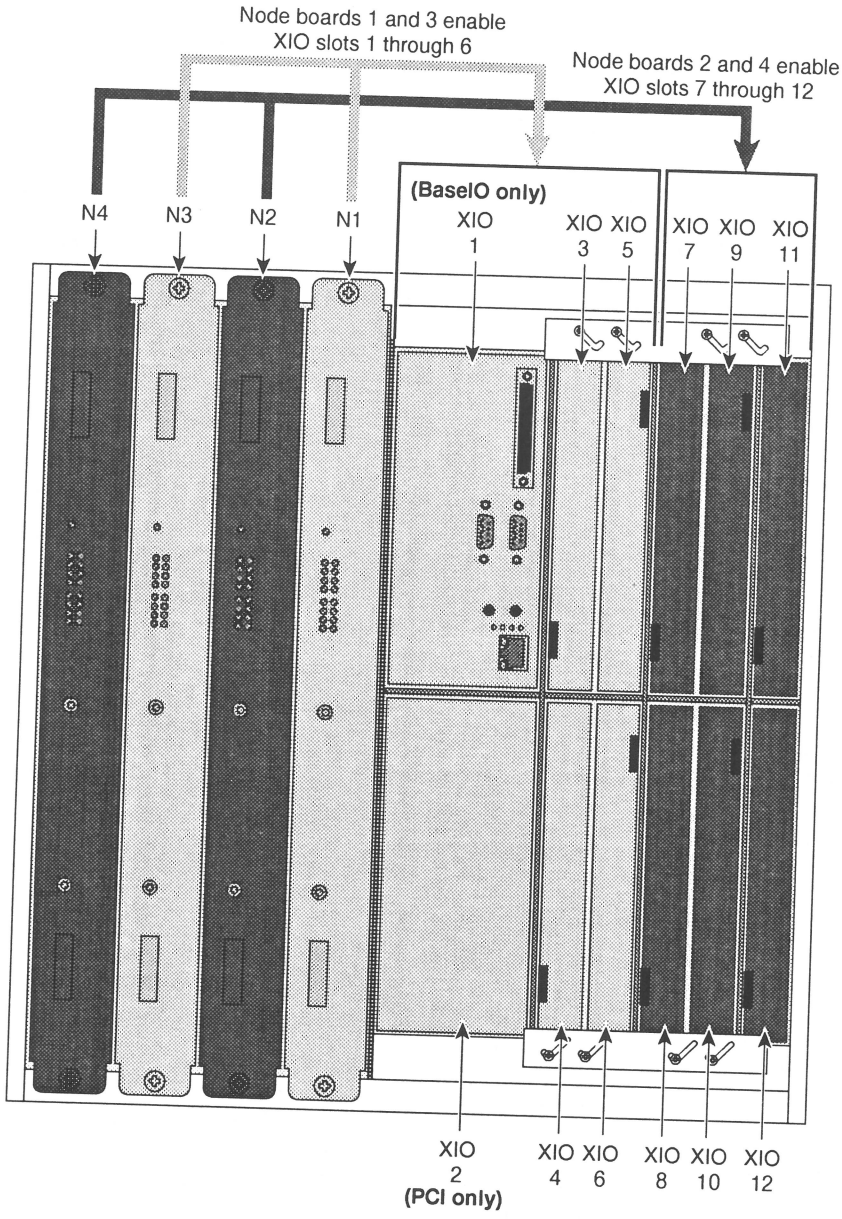
The midplane is mounted in the center of the module.

- Boards connect to each side of the midplane.
- The midplane distributes power and clock to the boards in the module.
- The midplane has active components on it that.



Student Notes

4-32 FOIL: Board Numbering



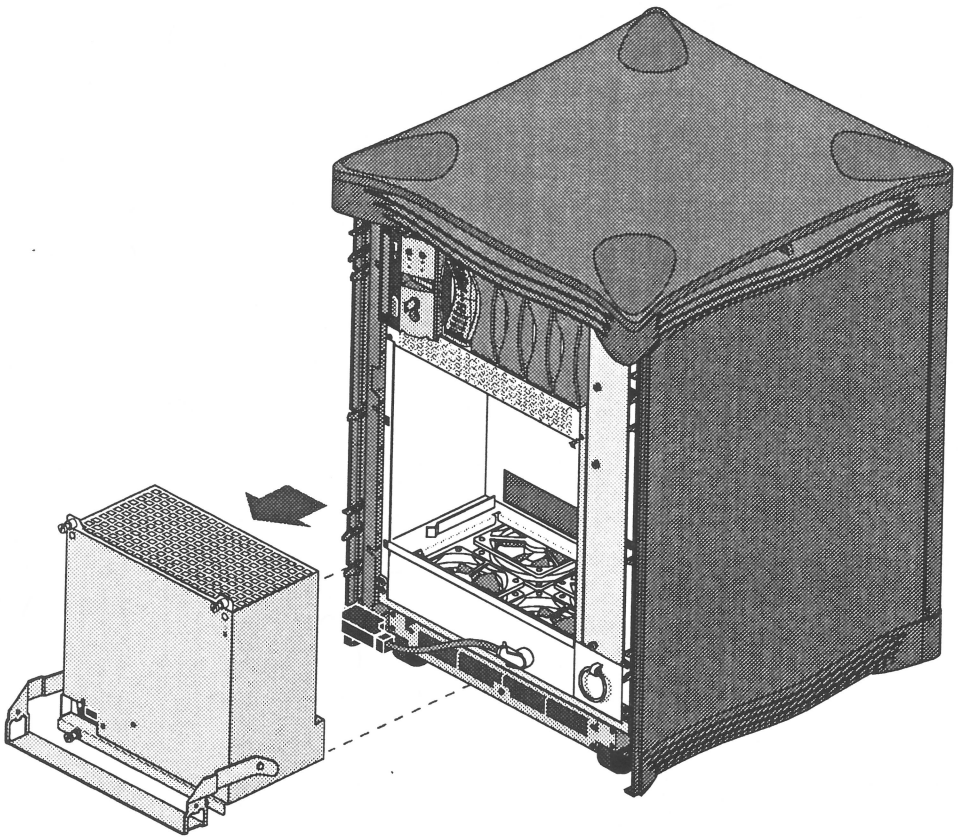
Student Notes

4-33 FOIL: Power and Cooling

A50/60 Hz power supply that plugs into the midplane provides power to the boards. Input power is 110 Vac or 220 Vac at 20 amps.
< IF 2 NODES MUST USE 220V

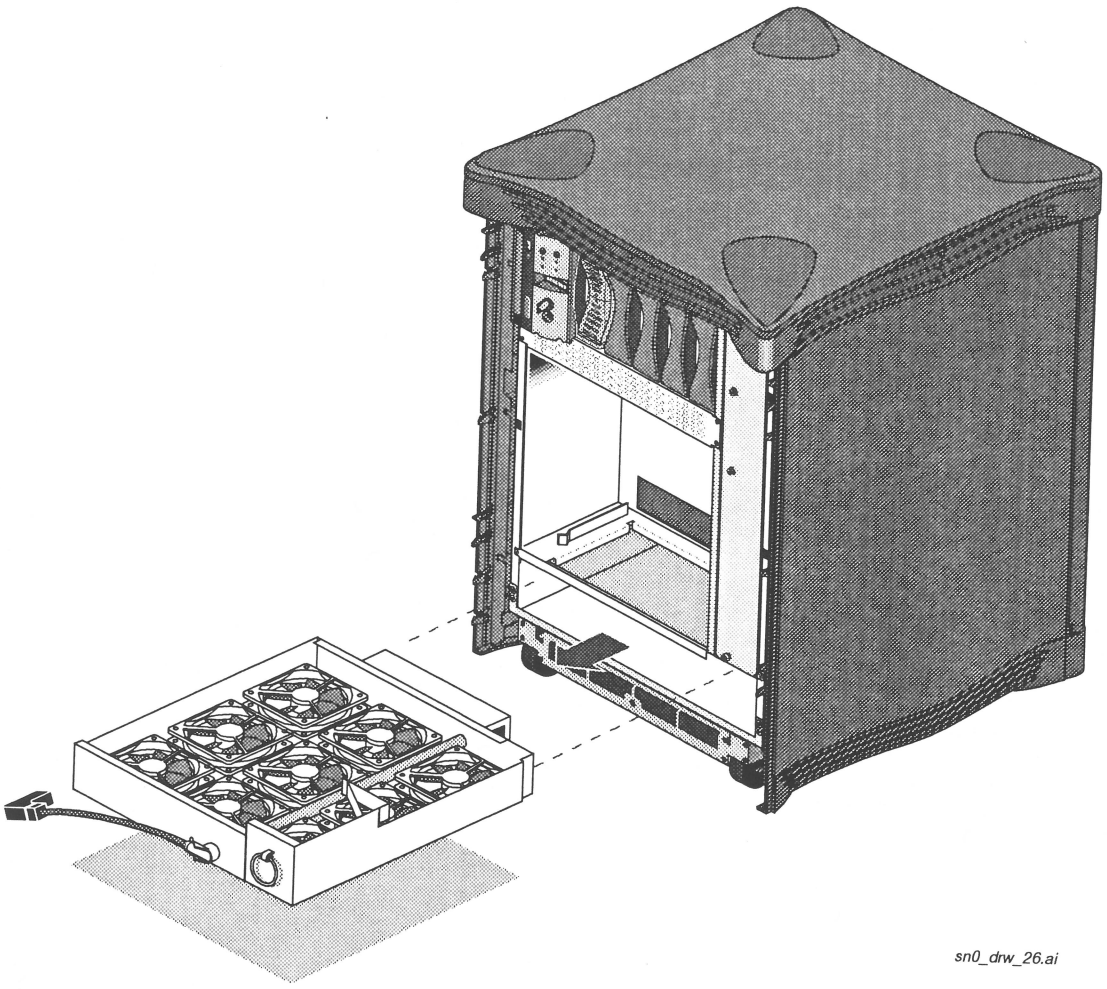
A fan tray mounted at the bottom of the module provides component cooling.

- Air is pulled by the fan tray through the top of the module, down over the boards, and exits through the bottom of the module.
- There are nine fans in the module.
- Speed control of the fans is regulated by the system controller.
- If one of the six rear fans stops running, the system controller increases the speed of the fans to compensate for the decreased airflow.
- If one of the three front fans stops running, the module powers down.

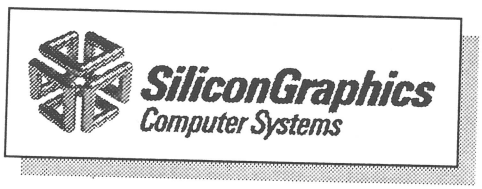


Student Notes

4-34 FOIL: Power and Cooling (continued)



sn0_drw_26.ai



Module 5: Hardware Configurations

Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

5-2 FOIL: Objectives

- Name the three ASICs that are used in the system interconnect fabric used in the Origin2000™ system and their functions.
- List the node board-to-router board connections.
- Name which type of router is used in the following configurations:
  - 2-node
  - 4-node
  - 8-node
  - 16-node
  - 64-node

Student Notes

5-3 FOIL: Module Topics

- CrayLink™ Interconnect Connections.
- Configurations.
  - 2 nodes.
  - 4 nodes.
  - 8 nodes without XpressLink™ cables.
  - 8 nodes with Xpress Link cables.
  - 16 nodes.
- 2- through 64-node cabling diagrams.

Student Notes

5-4 FOIL: CrayLink Interconnect

The CrayLink interconnect is a collection of very high speed links and routers that is responsible for interconnecting hubs that compose the system.

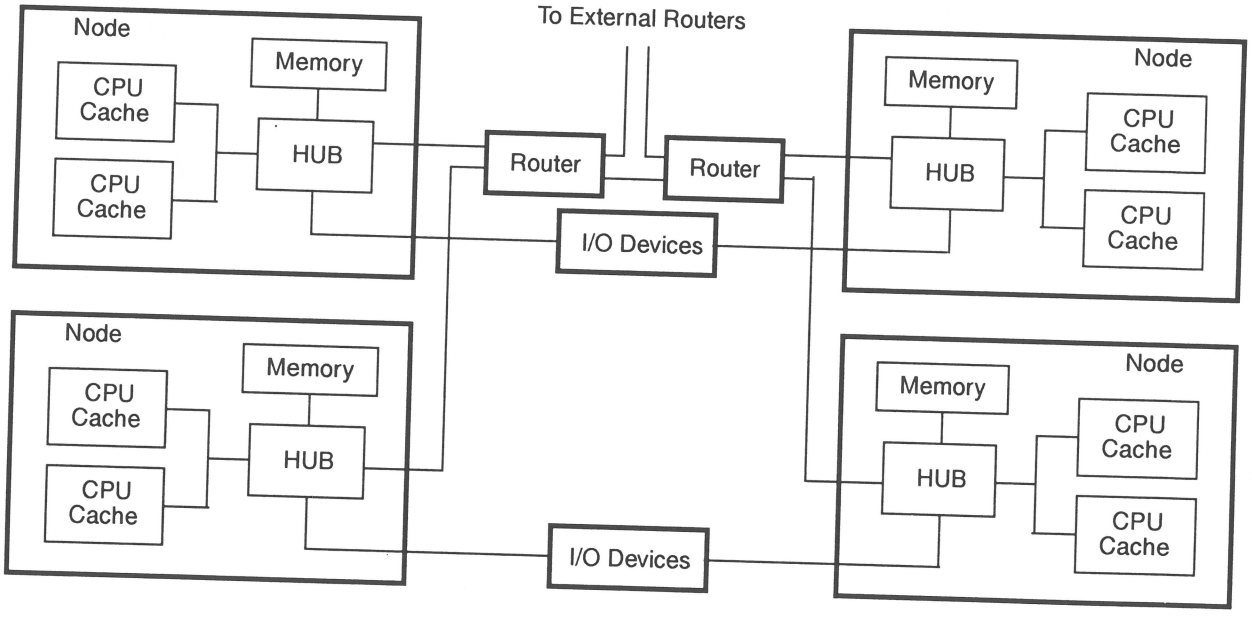
The important attributes of CrayLink Interconnect are:

- low latency.
- scalable bandwidth.
- modularity.
- fault tolerance.

The Origin2000 system uses an interconnection fabric and crossbar switches. The interconnection fabric is a web of dynamically allocated switch-connected links that attach nodes to one another. Crossbars are part of the interconnection fabric and are located inside several of the ASICs. There are three crossbar ASICs:

- XBOW for I/O.
- Router for node-to-node communication.
- HUB ASIC for communication between node components; and for communication to I/O and routers.

Crossbars dynamically link ASIC input ports with their output ports.



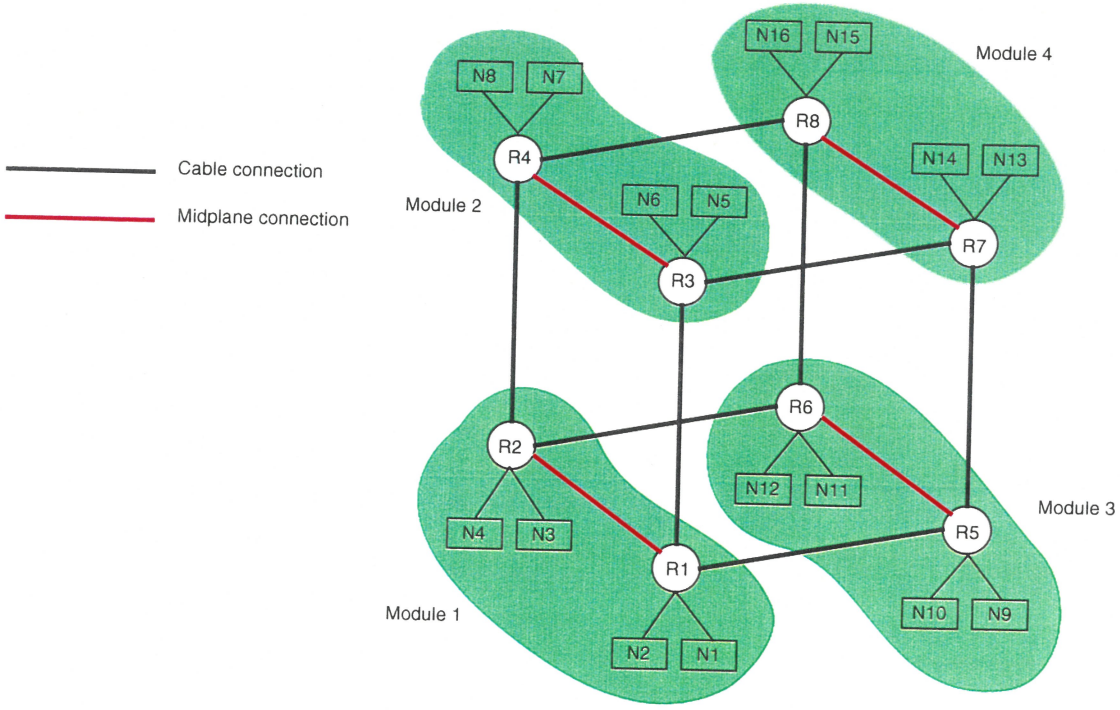
Origin2000 Module Architecture

Student Notes

5-5 FOIL: Interconnect Fabric

The interconnection fabric is a set of switches, called **routers**, that are linked by **cables** in various configurations, or topologies. The interconnection fabric differs from a standard bus in the following ways:

- The interconnection fabric is a mesh of multiple point-to-point links that are connected by the routers. These links and switches allow **multiple transactions to occur simultaneously.**
- The links permit **extremely fast switching**. Each **bidirectional link sustains as much bandwidth as the entire Everest bus.**
- The interconnection fabric **does not require arbitration** nor is it limited by contention, but a bus must be accessed through arbitration.
- More routers and links are added as nodes are added, which increases the interconnection's bandwidth. A shared bus has a fixed bandwidth.
- The topology of the CrayLink Interconnect is such that the **bandwidth grows linearly** with the number of nodes in the system.



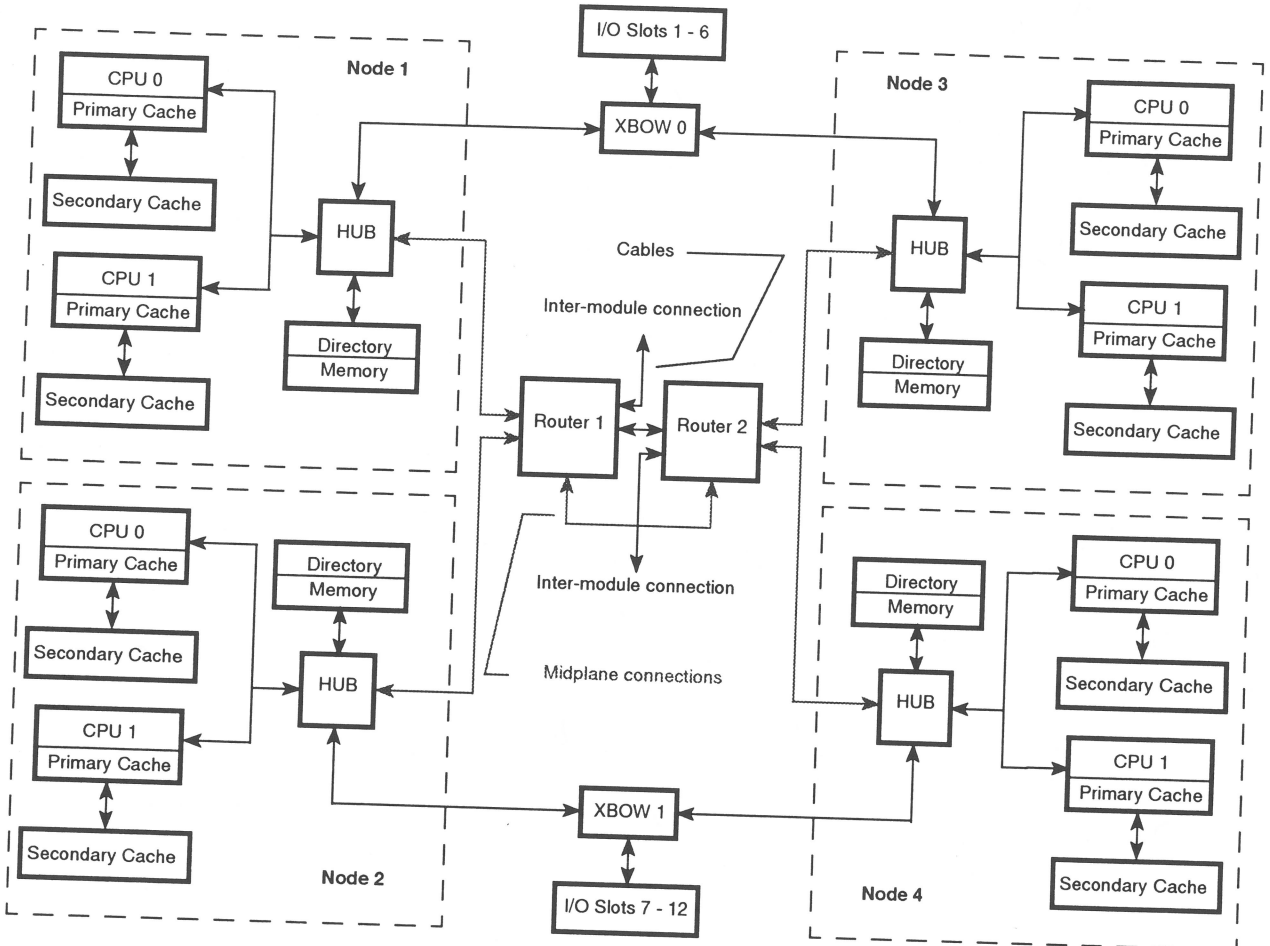
The interconnection fabric provides a **minimum of two separate paths to every pair of Origin2000 nodes**. This redundancy allows the system to bypass failing routers or broken interconnection fabric links. Each link is additionally protected by a CRC code and a link-level protocol, which retry any corrupted transmissions and provide fault tolerance for transient errors.

Student Notes

5-6 FOIL: Midplane-to-Router Connections

As shown in Module 4 (System Components), each router has three midplane connections:

- Two node-to-router connections.
- One router-to-router connection.



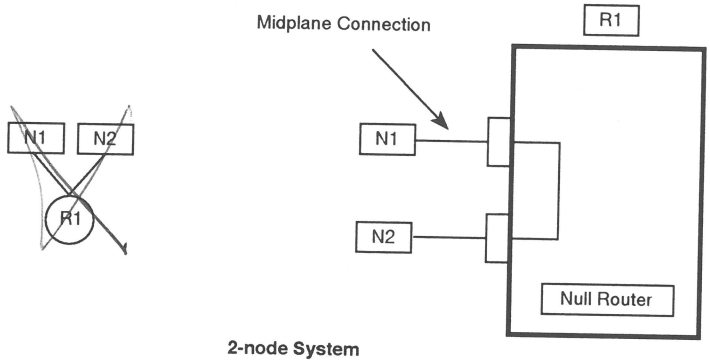
Module Block Diagram

Student Notes

5-7 FOIL: Configurations

Two-node System

The two nodes are connected by a Null Router through the midplane.



2-node System



Student Notes

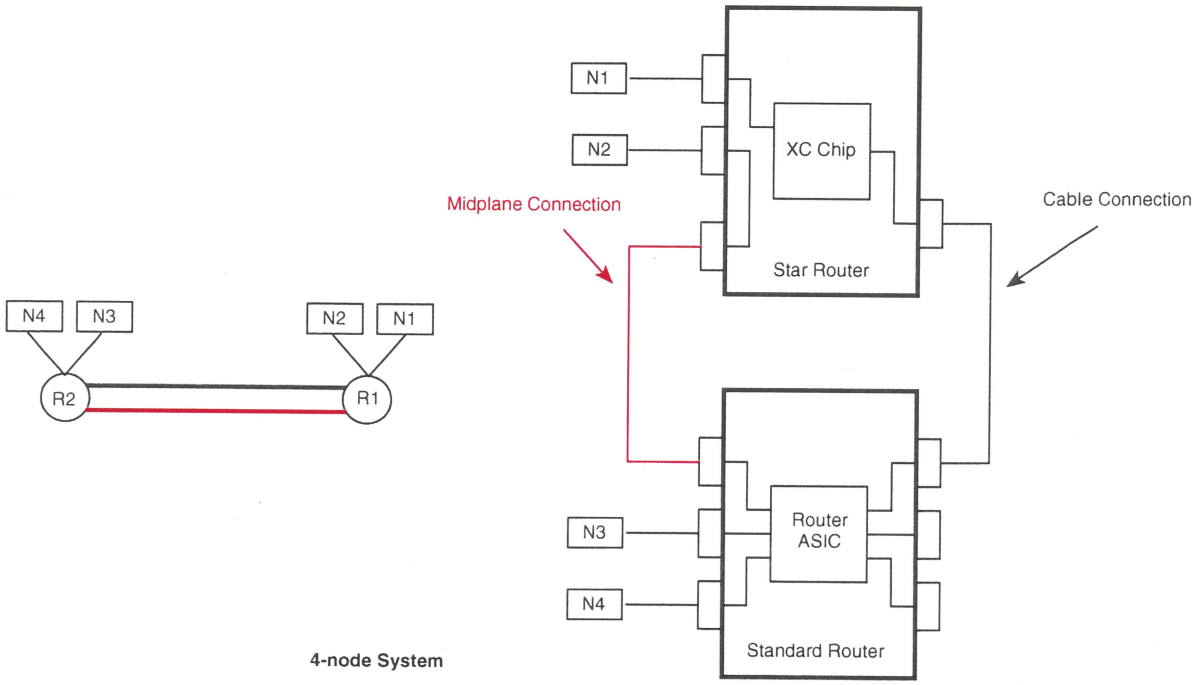
5-8 FOIL: Configurations (continued)

Four-node System

Two of the nodes are connected to a Star Router via a midplane connection and the other two nodes are connected to a standard router via the midplane connection. The two routers are connected by a cable (actually a jumper).

The XC chip in the Star Router converts STL to PECL and vice versa. The XC chip does not have a crossbar switch.

This system is contained in one module.



4-node System

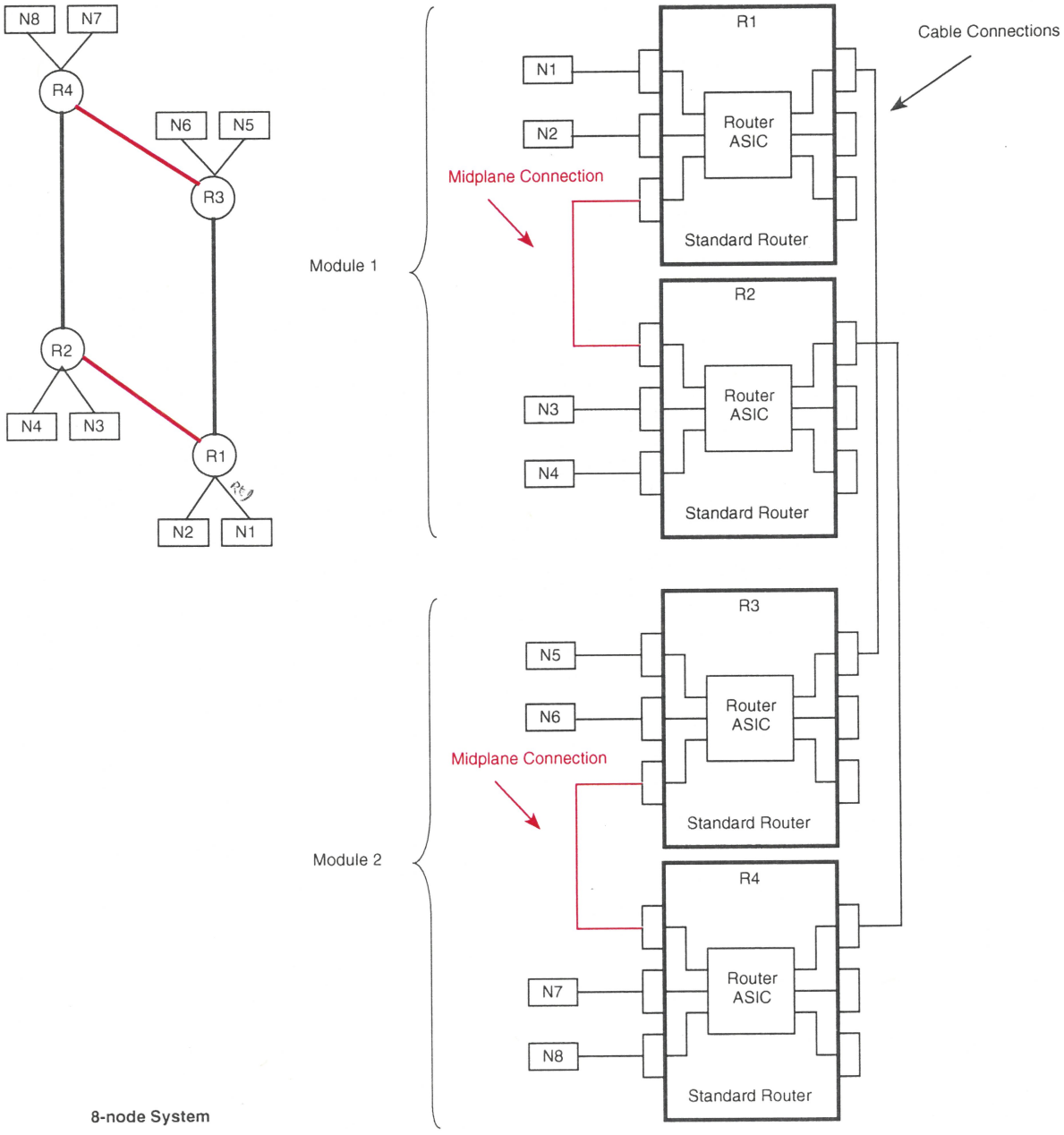
Student Notes

5-9 FOIL: Configurations (continued)

Eight-node System

Eight nodes are connected to four standard routers. This system is normally mounted in two modules, which are mounted in a rack.

The two nodes in each of the two modules are connected by the midplane connection. The two modules are connected to each other by cables.



8-node System

Student Notes

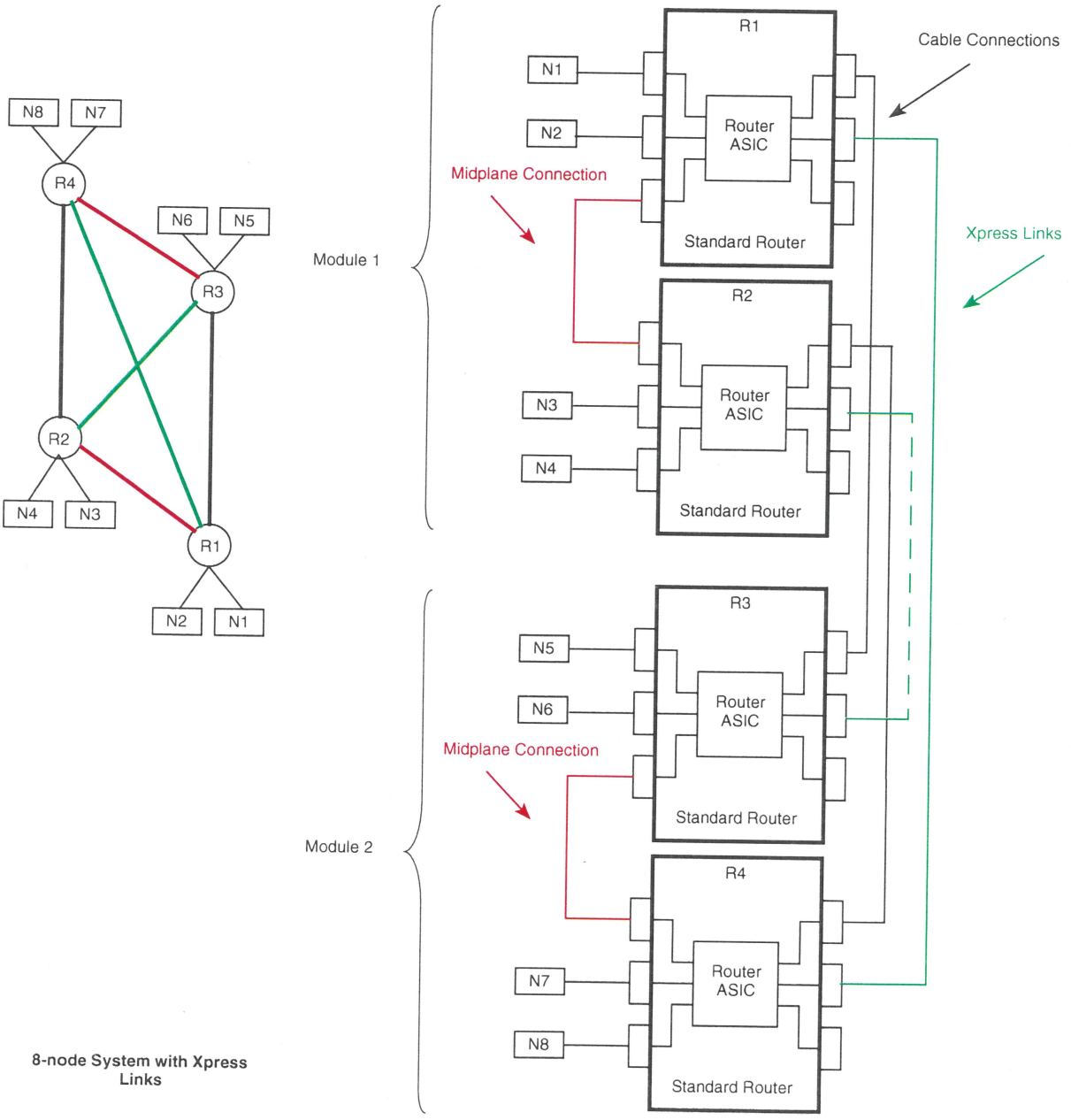
5-10 FOIL: Configurations (continued)

Eight-node System with XpressLinks

Eight nodes are connected to four standard routers.

The two nodes in each of the two modules are connected by the midplane connection. The two modules are connected to each other by cables.

XpressLinks are shown on the diagram using green lines. The use of XpressLinks eliminates one hop.



8-node System with Xpress Links

Student Notes

5-11 FOIL: Configurations (continued)

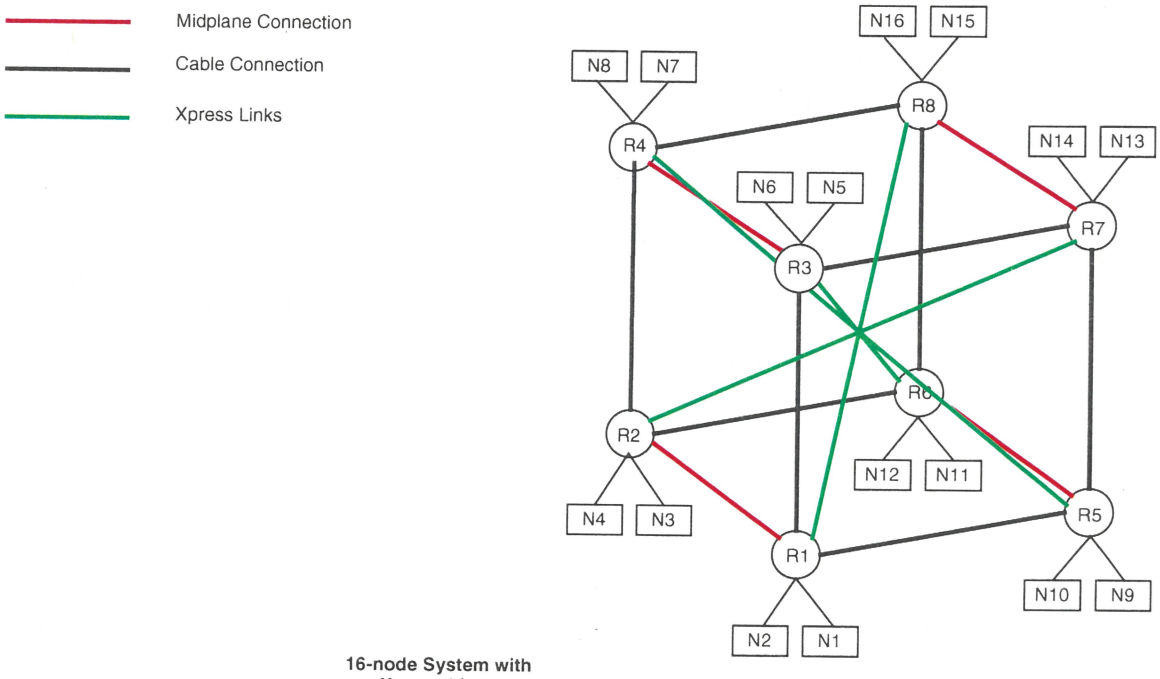
Sixteen-node System with XpressLinks

A sixteen-node system is normally mounted in two rackmount chassis.

This configuration differs from the eight-node configuration in that there are no XpressLinks between the modules in the rack, but the XpressLinks are inter-rack.

There are only six ports on a router. To reduce latency between racks, XpressLinks are installed between:

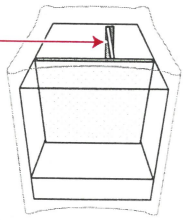
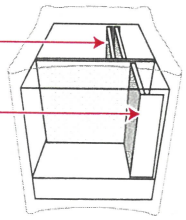
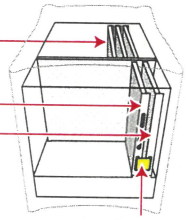
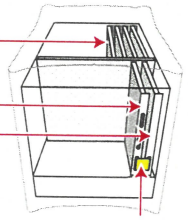
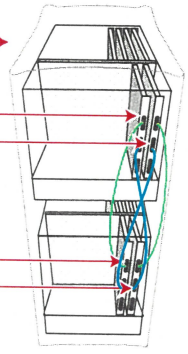
- Router 1 in rack 1 to router 8 in rack 2.
- Router 2 in rack 1 to router 7 in rack 2.
- Router 3 in rack 1 to router 6 in rack 2.
- Router 4 in rack 2 to router 5 in rack 2.



16-node System with Xpress Links

Student Notes

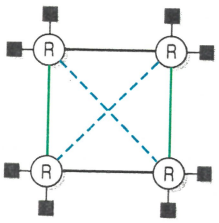
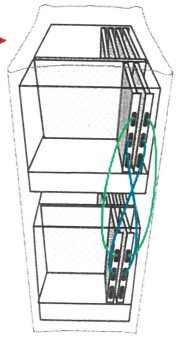
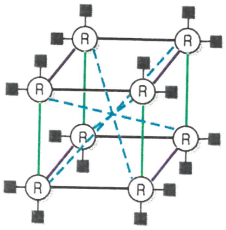
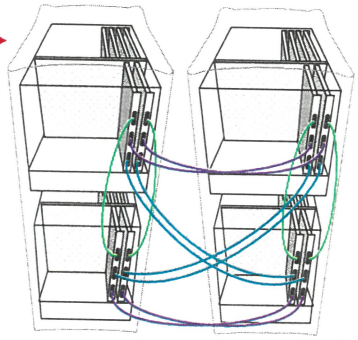
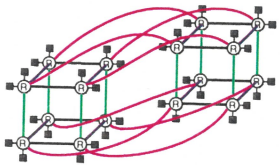
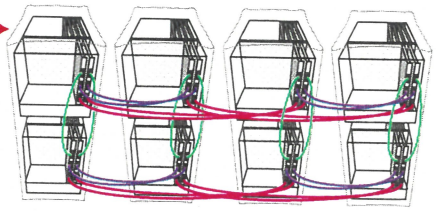
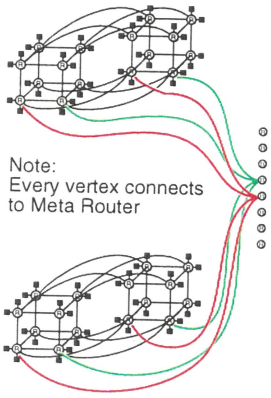
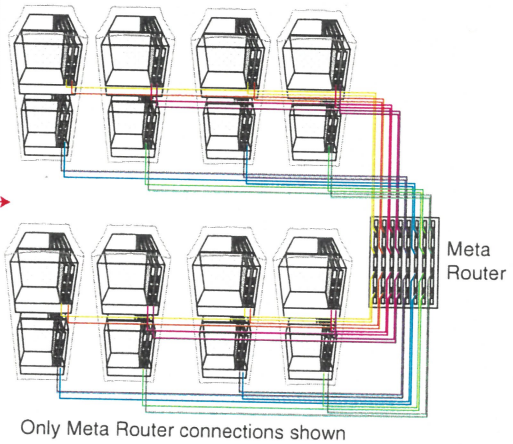
5-12 FOIL: Configuration Table

Schematic	Number of Node Boards	Router Board Types	Configuration
R10K Processor R10K Processor N	1 Node Board (up to 2 processors)	None	
N N NR	2 Node Boards (up to 4 processors)	Null Router Board	
N N N R SR IR1 Jumper	3 Node Boards (up to 6 processors)	Router Board Star Router Board	 IR1 Jumper
N N N R SR N IR1 Jumper	4 Node Boards (up to 8 processors)	Router Board Star Router Board	 IR1 Jumper
N N N R R N N R R N N N With Express Links	8 Node Boards Rackmount System (up to 16 processors)	Router Board Router Board Router Board Router Board	

N = Node Board
NR = Null Router Board
SR = Star Router Board
R = Router Board

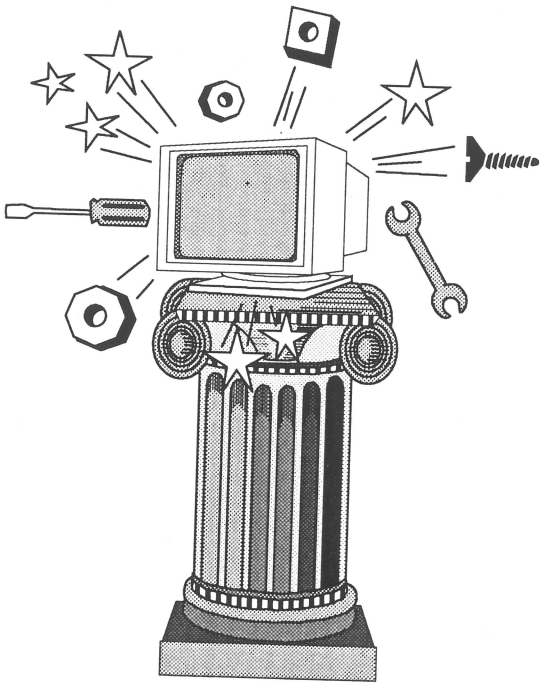
Student Notes

5-13 FOIL: Configuration Table (continued)

Schematic	Number of Node Boards	Configuration
 With Express Links	8 Node Boards (up to 16 processors) Rackmount System	
 With Express Links	16 Node Boards (up to 32 processors) Multi Rackmount System	
	32 Node Boards (up to 64 processors) Multi Rackmount System	
 Note: Every vertex connects to Meta Router	64 Node Boards (up to 128 processors) Multi Rackmount System with a Meta Router	 Meta Router Only Meta Router connections shown

SiliconGraphics
Computer Systems

Module 6: ESD: Board Handling & Removal



Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

6-2 FOIL: Objectives

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

- Take the proper precautions to guard against ESD damage to equipment.
- Prevent damage to the exterior skins of the equipment.
- Handle the system components without damaging them.

Student Notes

6-3 FOIL: Module Topics

- ESD precautions.
- Module skin removal.
- Board removal.

Student Notes

6-4 FOIL: ESD

Electrostatic discharge (ESD) can be a very serious problem. ESD can damage sensitive electronic components. This damage can cost us money by causing:

- Excessive down time.
- Intermittent problems.
- DOA failures.

It is important that SSEs do not cause ESD failures.

ESD Prevention

You can prevent ESD by following several steps:

- Touch a table before you touch a component; this equalizes the difference in potential.
- Use ESD mats.
- Use wrist straps, heel straps, and smocks.

We must also educate our customers about the potential damage caused by ESD. By teaching ESD precautions to the customers at the site we may be able to avoid some incidents.

Student Notes

6-5 FOIL: General Parts Removal

Observe the following precautions when you remove or replace parts on the system:

- Ensure that power has been removed before you install or remove parts.
- Use proper ESD precautions.
- Do not remove parts from the ESD bag until you are ready to use them.
- Have the proper tools at hand.
- Do not exceed the 1.25-inch bend radius of cables when you handle them.
- Do not touch connectors.
- Place skins that you remove in an area where they will not be scratched or damaged.

Student Notes

6-6 FOIL: Lab Project 1: ESD: Board Handling & Removal

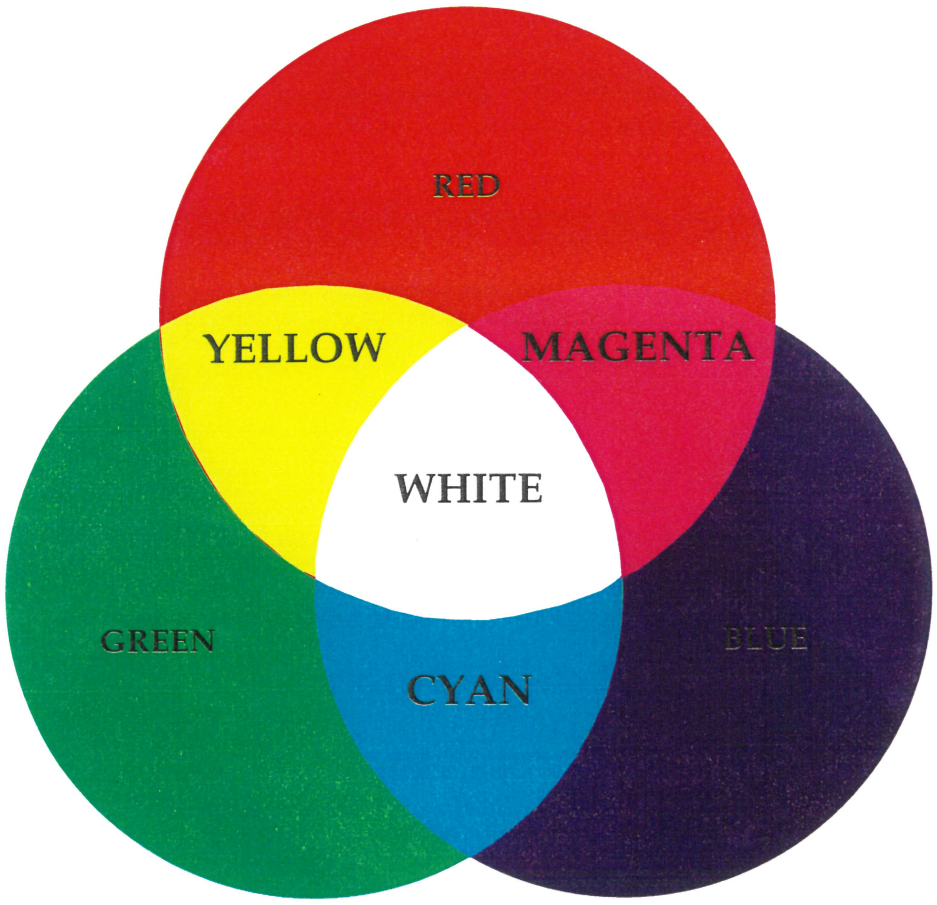
- Student teams will perform this lab project at a classroom rack system.
- Time allotted to complete project: **90 minutes**.
- After completion of the lab project, the student should be able to:
  - Take the proper precautions to guard against damaging equipment due to ESD.
  - Prevent damage to the exterior skins of the equipment.
  - Handle the system components without damaging them.
  - Remove and install the following FRUs:
    - Exterior skins.
    - Cables.
    - Cable management system.
    - IO6 board.
    - Node board.
    - Router board.

12-8 FOIL: Secondary Colors

- Secondary Colors: **Cyan**, **Yellow**, and **Magenta**.

  Mixing any two primary colors (of equal intensity) creates a secondary color.

  - **Cyan** = Blue & Green
  - **Yellow** = Green & Red
  - **Magenta** = Red & Blue



Student Notes

12-9 FOIL: Colors Exercise

This is an instructor demonstration.

- Login to **gtutor**.
- Execute **pixels**.
- Execute **colors**.
- Execute **showmap**.
  - Start **cedit** (color editor).
  - Start **interpolate**.
- **Edit** an **existing color** in the colormap.
- **Create** a **new color** in the color map.

Student Notes

12-10 FOIL: Screen Coordinates

- Monitor resolution varies with the model.
- Actual screen coordinates for the **standard** monitor are:

  X = rows are **1,920** pixels wide (horizontal).

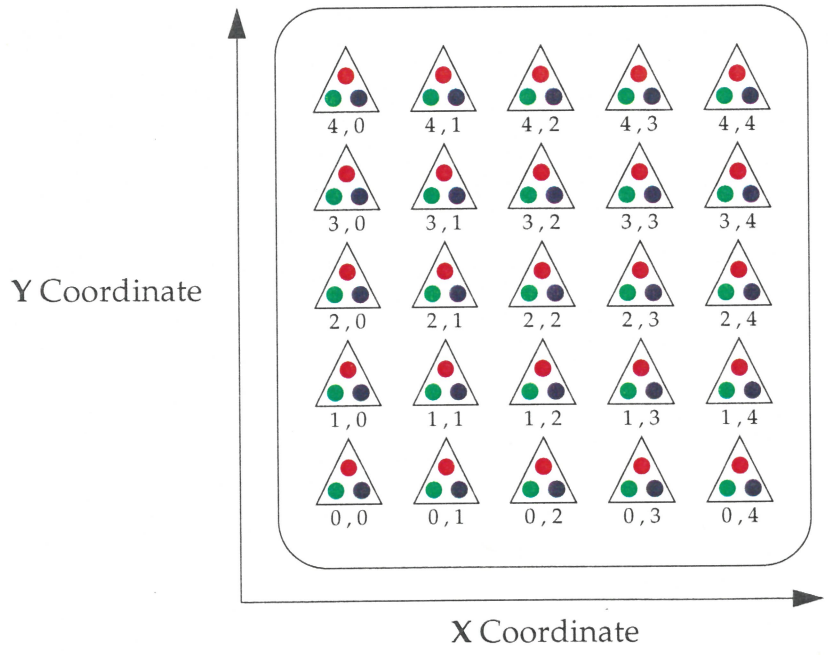
  Y = spans are **1,200** pixels high (vertical).

- Each pixel is supported by hundreds of bit planes, the number depending on the number of RM boards and the resolution that the programmer selects.

  Note: Lines that are drawn on a diagonal will be jagged (**aliased**).

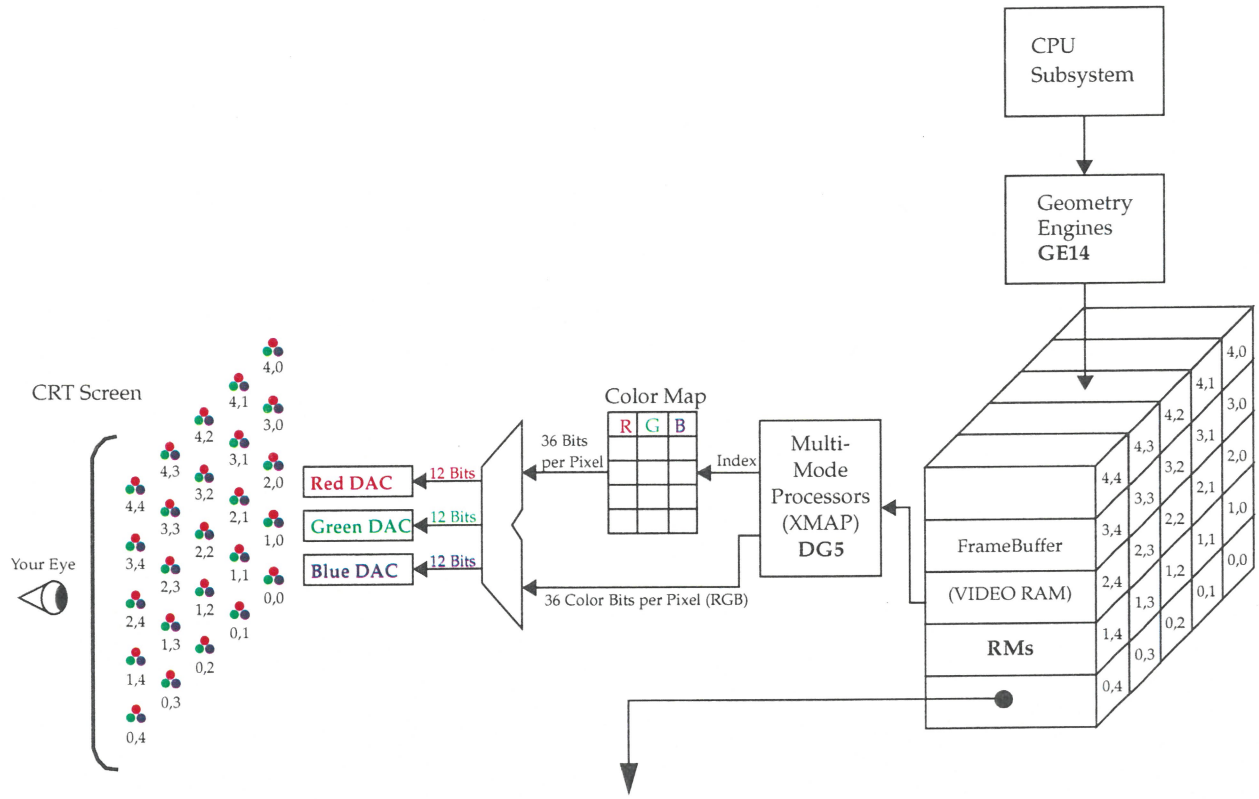
  For example: Starting point **0, 0**

  Ending point **3, 4**



Student Notes

12-11 FOIL: Pixel vs. FrameBuffer



Buffer Name	Front Image Buffer				Rear Image Buffer				Z-Buffer	Accumulation Buffer	Multi-Sample (Anti-aliasing)	Stencil	Overlay / Underlay	Window ID
	R	G	B	A	R	G	B	A						
Number of Bits	48 Bits				48 Bits				Varies	Varies	Varies	Varies	Varies	Varies

- The depth of the individual pixel memory depends on the following:

  Number or **RM boards**, **resolution** and **sweep rate** of monitor, and graphics functions that the programmer is using. **For example:** stereo vision, single-buffer, etc.

- **Example**: If the specified resolution is **1280 x 1024**, then the depth equals:

  1 RM7 = 512 bits of depth.

  2 RM7 = 1,024 bits of depth.

  4 RM7 = 2,048 bits of depth.

Student Notes

12-12 FOIL: Image Buffers

Depending on the color mode that the programmer is uses, the **Image Buffers** hold the following:

- **RGB Mode**:
  - Image buffer holds **RGBA** data for each screen pixel.
  - 48 Bits: 12 Red, 12 Green, 12 Blue, and 12 Alpha.
- **Color Map Mode**:
  - Image Buffer holds an address (index) in the Color Map.
    - For **DG5-8**, the color map is a **32K x 36** bit memory.
    - For **DG5-2**, the color map is a **8K x 36** bit memory.
  - The addressed Color Map location holds 36 bits of **RGB**.
- **XMAP** (Multi-Graphics Processor) outputs the **RGB** data at a rate synchronized to the Graphics Monitor: 60 HRz N.I., 30 HRz N.I., 30 HRz I., 25 HRz (PAL).
- **Static** images use **Single Buffer** mode.
  - Pipe loads the back buffer.
  - XMAP unloads the buffer.
- **Dynamic** images use **Double Buffer** mode to smooth out motion.
  - Pipe loads the back buffer.
  - XMAP unloads the back buffer while pipe fills the front buffer.
  - XMAP unloads the front buffer while pipe fills the back buffer.

Student Notes

12-13 FOIL: Motion Exercise

This is an instructor demonstration.

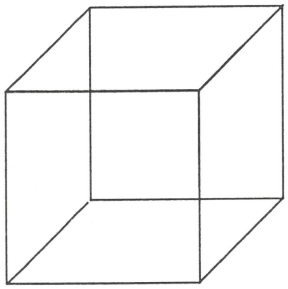
- Execute **single.buffer**.
- Execute **double.buffer**.
- Execute **snowman**.

  This scene is drawn is **single-buffer** mode, and as the cursor is moved out of the window and then back in, the scene blinks.

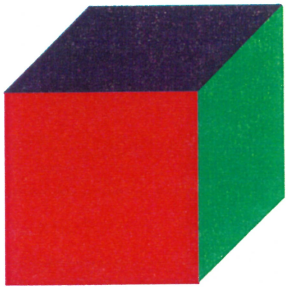
Student Notes

12-14 FOIL: 2D Drawings

- Points, lines, and polygons are drawn on a **2-dimensional** plane.
  - **X** = Horizontal position in the plane.
  - **Y** = Vertical position in the plane.
- Images drawn are flat, therefore they cannot be **rotated**, **moved**, or **scaled**.
- **3-dimensional** objects can be simulated by connecting several flat polygons together or by using the wireframe approach.



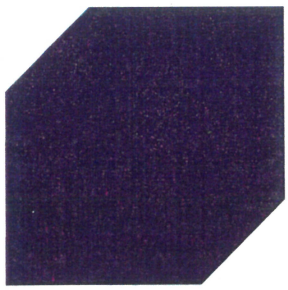
Wireframe



3 Flat Polygons

Note: Because each polygon is a different color, the object appears to be 3-dimensional.

If all the polygons were the same color, the object would appear as a six-sided geometric figure (hexagon).



Student Notes

12-15 FOIL: 2D Exercise

This is an instructor demonstration.

- Execute **pseudo.cubes**.
- Execute **hexagon**.

Student Notes

12-16 FOIL: 3-Dimensional Space

- The programmer defines a **3-dimensional** viewing volume called **World Space**.
- 3D objects are **projected** (drawn) into the viewing volume.
- Three types of viewing volumes are available:
  - **Orthographic** world space.
  - **Perspective** world space.
  - **Window** world space.

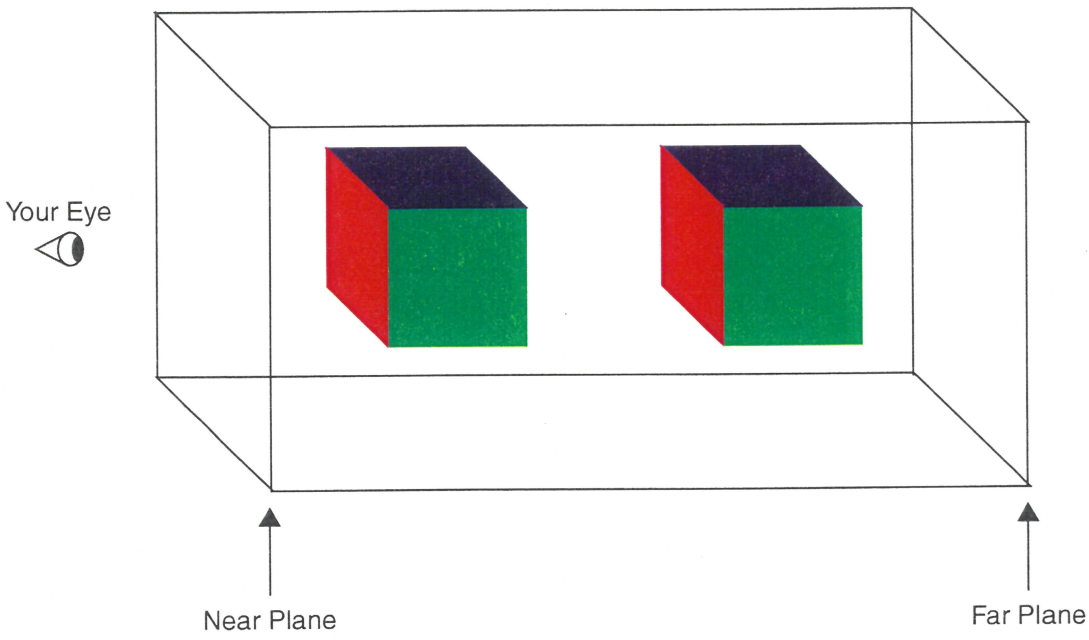
    This is basically the same as perspective, but it allows the programmer to perform special effects.
- Three coordinates define the volume dimensions:
  - **X** = Horizontal
  - **Y** = Vertical
  - **Z** = Depth
- The viewing volume (world space) is defined by six clipping planes: **Near, far, left, right, top**, and **bottom**.
- The **Geometry Engine** calculates how 3D objects are projected into the **3-dimensional** volume, but displayed on a **2-dimensional** screen.

Student Notes

12-17 FOIL: Orthographic Space

Orthographic projections cause objects (that are the same size) to look the same size no matter how **near** of **far** away they are to the viewer (your eye).

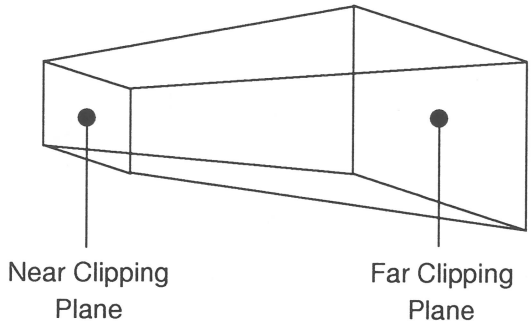
There is no true perspective of objects projected into this space.



Student Notes

12-18 FOIL: Perspective Space

- This space allows objects to be viewed as they would be seen relative to **size** and **distance** from the viewer.
- The perspective world space also has **six clipping planes**, but the viewing volume looks like a **truncated pyramid**.



Student Notes

12-19 FOIL: World Space Exercise

This is an instructor demonstration.

- Execute **ortho.space**.
- Execute **perspective.space**.

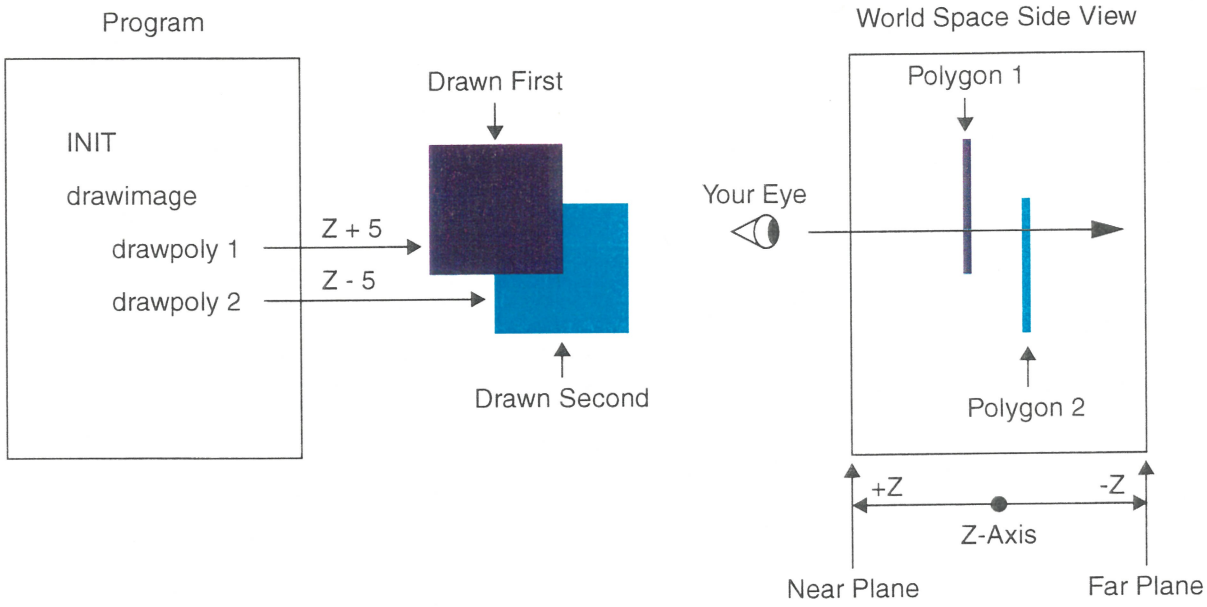
Student Notes

12-20 FOIL: Z Buffer

- These bit planes (of the **FrameBuffer**) are used for two functions:

  Hidden Surface Removal and **Depth Cuing**.

- **Hidden Surface Removal**:
  - Several objects can share the same **X** and **Y** coordinates.
  - There is only **one pixel** per **X/Y** coordinate.
  - **Z-value** specifies the **distance** from the eye.
  - The Geometry Engine loads the **RGBA** data of the **closest** object (for objects that share **X/Y** coordinates) into the **image buffer**.



- **Depth Cuing**:
  - Changing (interpolating) the **intensity** of a color along the **Z-axis**.
  - Geometry Engine performs the calculations.
  - Gives the effect of an object disappearing into the distance (**depth**).

Student Notes

12-21 FOIL: Z Buffer Exercise

This is an instructor demonstration.

- Execute **hidden.surfaces**.

  The **mouse** changes positions of the polygons relative to the **Z-value**.

- Execute **solid.line**.

  A **gray** line (**5** pixels wide) is drawn in 3D space **without Depth Cuing**.

- Execute **depth.cued.line**.

  The same line with **Depth Cuing** on.

Student Notes

12-22 FOIL: Lighting

- Realistic 3D modeling requires a **light source**.
  - **Several light** sources are available.
  - **Multiple** light colors can be specified.
- Four parameters must be specified by the programmer:
  - **Material Property**:

    How the object will **reflect** or **absorb** light.

    For example: Smooth, dull, plastic, gold, plaster, etc.
  - **Light Source**:

    This specifies the **intensity**, **color**, and **number** of lights.
  - **Lighting Model**:

    Specifies the **position** of the lights; any **ambient** light present when all the lights are off, and light **attenuation**.
  - **Orientation**:

    The **angle** of the model's surfaces to the lights.

Student Notes

12-23 FOIL: Lighting Exercise

This is an instructor demonstration.

- Execute **flat.sphere**.

  This is a magenta colored sphere without any light shown on it: it looks flat.

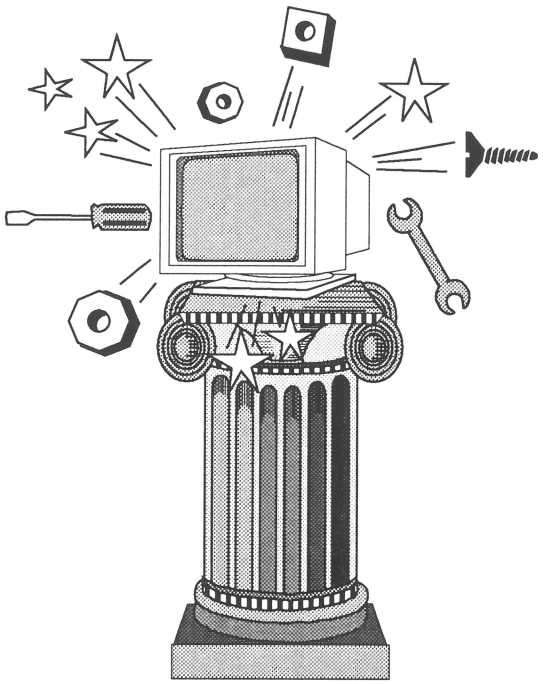
- Execute **lit.sphere**.

  This is the same sphere with a white light in geosynchronous orbit above the North Pole.

  Moving the mouse moves the sphere and the light together.

SiliconGraphics
Computer Systems

Module 13: The Onyx2™ System



Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

13-2 FOIL: Module Objectives

Upon completion of this module, using available documentation and the classroom Onyx2™ systems, you will be able to:

- Identify all the boards and cables required to install any of the various Onyx2 configurations.
- Match the functions performed to the individual Onyx2 boards.
- Correctly state all the special rules or warnings for installing the different configurations of the Onyx2 system.
- Correctly identify all the graphics configuration start-up processes that are called by the **init** process to initialize the graphics subsystem and state the order that the start-up processes are run.
- Manually start and stop the graphics system.
- Run the **irsaudit** diagnostic program.
- Configure the Onyx2 system using the **ircombine** utility.

Student Notes

13-3 FOIL: Onyx2 Products

- Offered in **deskside** or **rack** Systems.
  - **KEGO** was code name for the **deskside** system.
  - **KCAR** was code name for the graphics-insert module for the **rack** systems.
  - The **"K"** in the name implies **KONA** Graphics.

    KONA was the code name for the **InfiniteReality™** graphics system.
- Two versions exist for the deskside systems:
  - **Onyx2 Reality** system.

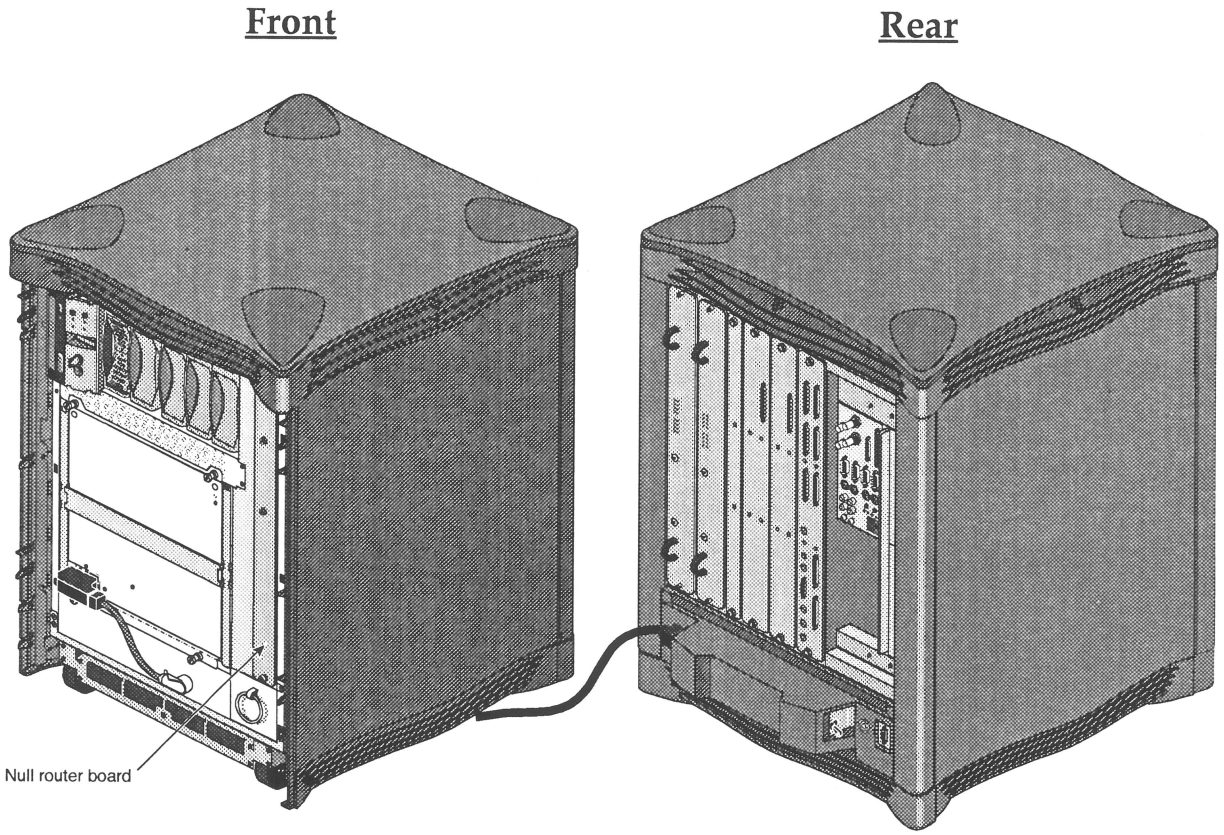
    InfiniteReality technology in a low cost package.
  - **Onyx2 InfiniteReality** system.

    Full performance Onyx2 system.
  - **Deskside** system offers only **one GFX pipe** (head).
- Several configurations are available for rack systems:
  - Onyx2 InfiniteReality.
  - **Two pipes** in each graphics-insert module (KCAR).
  - Multiple insert modules can be configured.
  - **Each pipe requires a separate node (IP27) board.**

Student Notes

13-4 FOIL: Onyx2 Reality

- **Deskside** chassis only.
- One or two node boards.
  - One to four CPUs.
  - 1-MB secondary caches.
- **GE14** board contains only **two Geometry Engine™** ASICs.
- Use the **RM8** raster manager board.
  - 16 or 64 MB of texture memory.
  - 40-MB frame buffer.
  - Processes 5-million meshed-triangles per second.



Student Notes

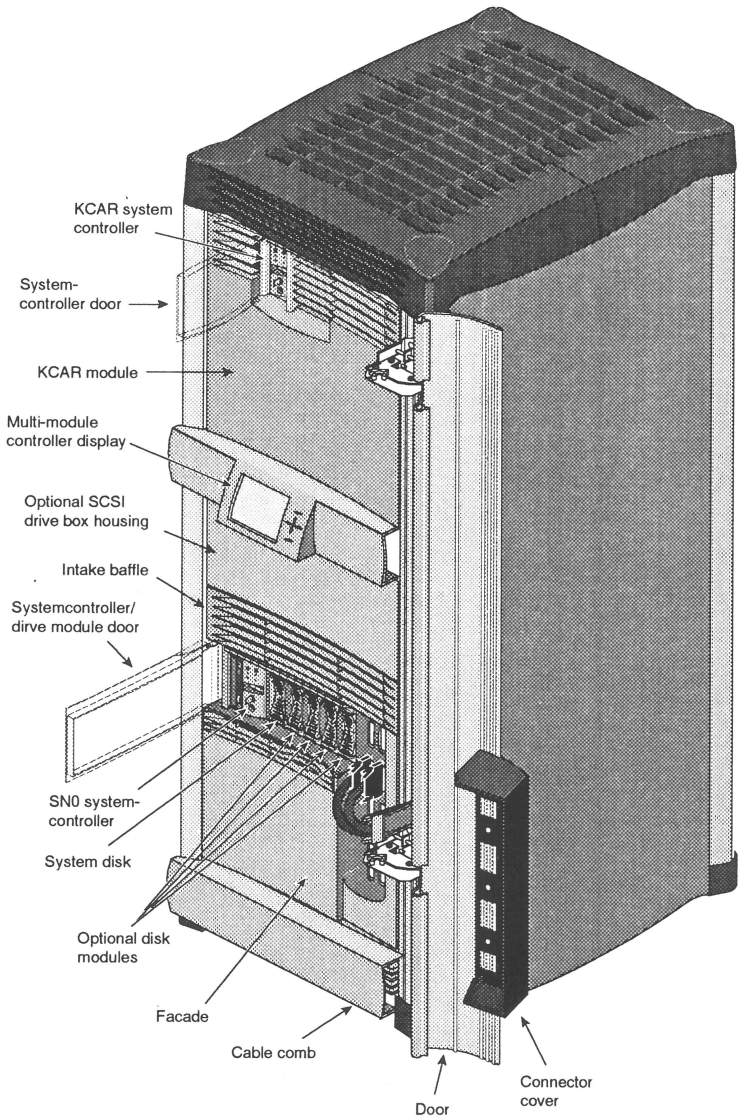
13-5 FOIL: Onyx2 InfiniteReality Deskside

- One or two node boards.
  - Two or four CPUs. 195 MHz
  - 4-MB secondary caches.
- **GE14** board contains **four Geometry Engine** ASICs.
- Uses the **RM7** Raster manager board.
  - 16 or 64 MB of texture memory.
  - 80-MB frame buffer.
  - Processes 11-million meshed-triangles per second.

Student Notes

13-6 FOIL: Onyx2 InfiniteReality Rack

- Uses the graphics-insert module (KCAR) with an Origin2000™ module.
- **2** to **24** CPUs with 4-MB secondary caches.
- **1** to **4** pipes (Onyx2 IR).
- Each GE14 has **four Geometry Engine** ASICs.
- Uses the **RM7** raster manager board.
  - 16 or 64 MB of texture memory.
  - 80-MB frame buffer.
  - Processes 11-million meshed-triangles per second.



Student Notes

13-7 FOIL: GFX Components

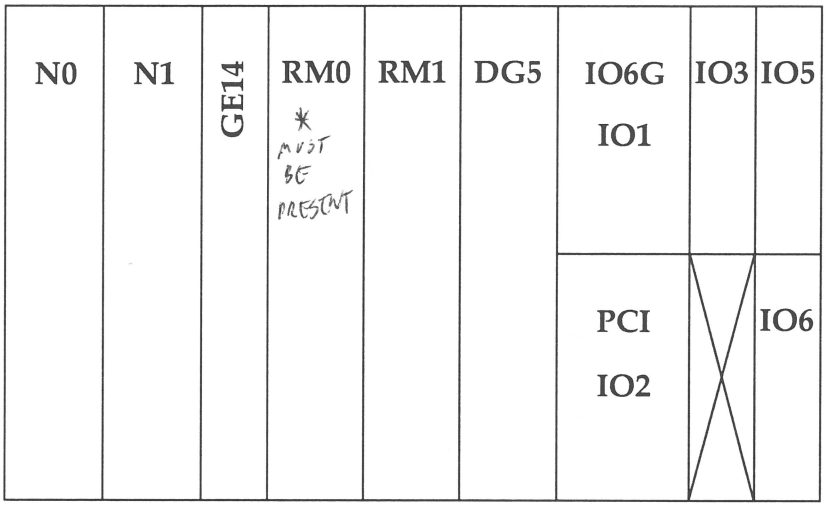
- **Graphic Boards**:
  - **RM8** (Reality deskside systems).

    FRU is a single board.
  - **RM7** (InfiniteReality - deskside or rack systems).

    FRU is a **pair** of boards connected by flex circuits.
    - **RM7**
    - **TM7**
  - **GE14** (2 or 4 Geometry Engine ASICs).
  - **DG5** (several versions).
- **Interface Boards**.
  - **IO6G** (used in place of the IO6).
  - **Xtown** (Cross town) board. X10 SLOT
  - **Ktown** (Kona town) board (used in KCAR). 2K CROSS-TOWN BRDS
- Graphics Monitor.
- Graphics keyboard and mouse.
- Cables:

Student Notes

13-8 FOIL: Deskside Slot Positions

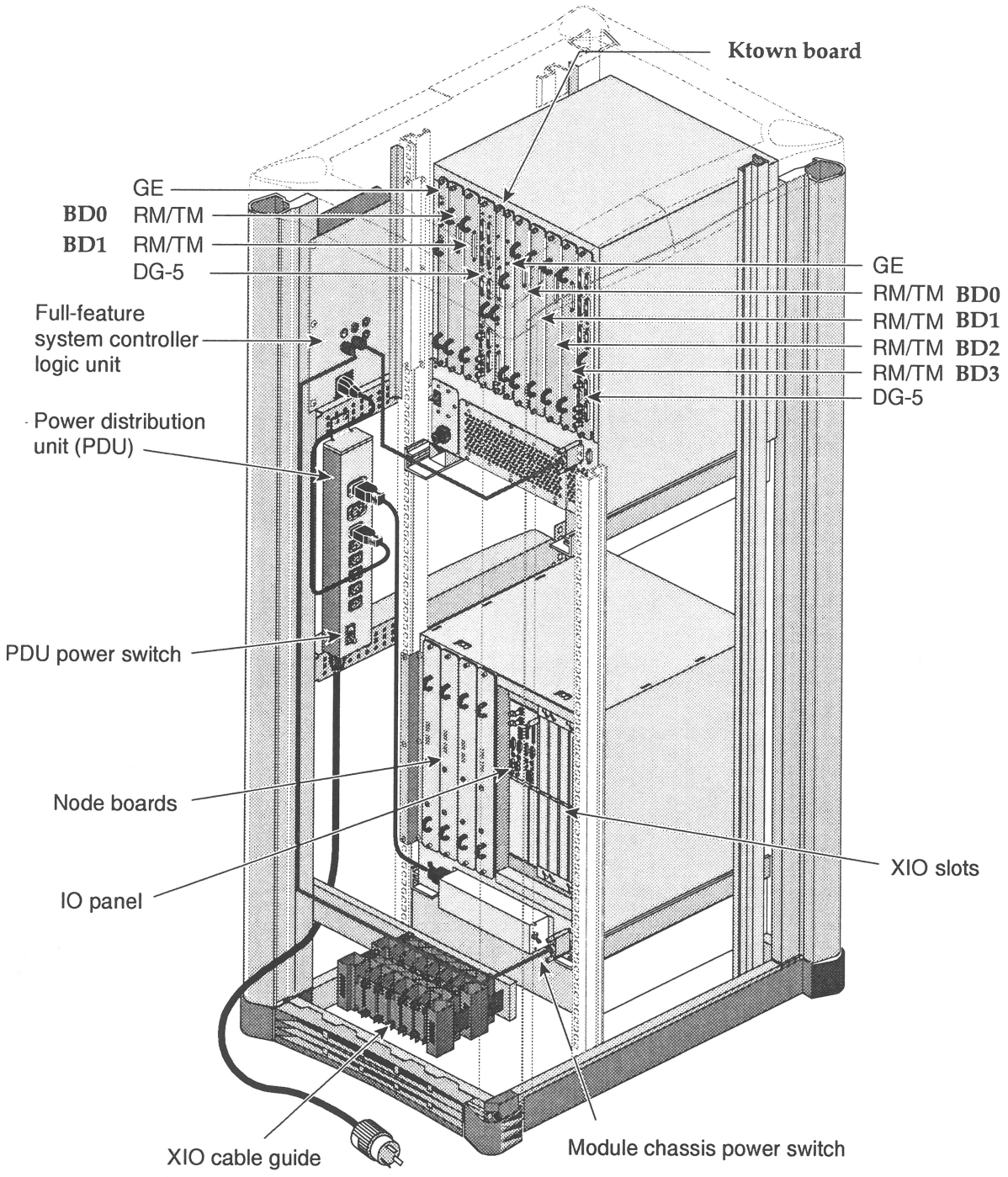


Rear View

- Slot IO3 can hold a full-size XIO card.
- Slot IO4 is unavailable because GE14 uses its port on the XBOW chip.

Student Notes

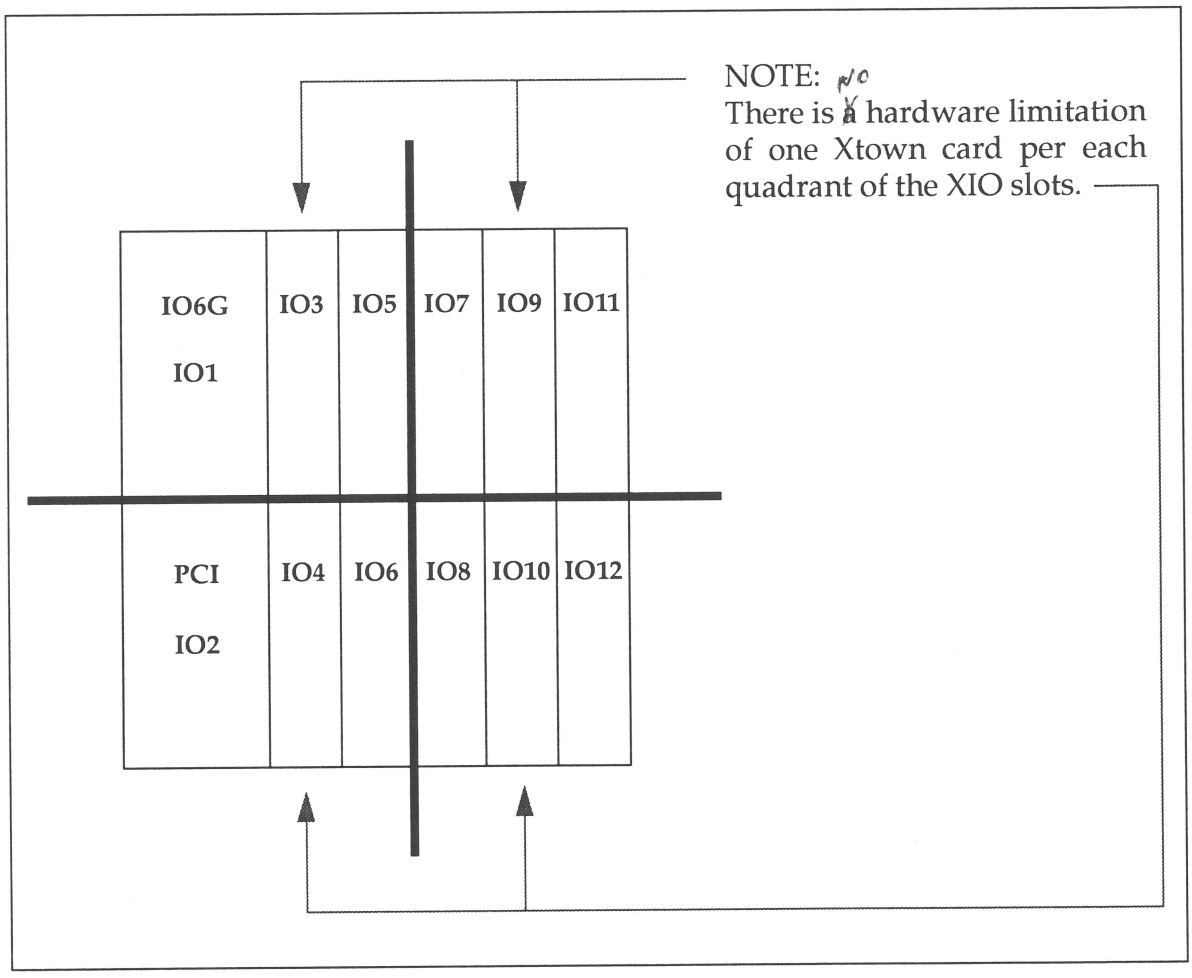
13-9 FOIL: Rack Slot Positions



Student Notes

13-10 FOIL: Crosstown Board (Xtown)

- Installs into an **XIO slot** (other than 1 or 2).
- Changes XBow 16-bit @ 400MHz into a slower, wider path for cabling.
- Uses the same technology used by the CrayLink connections.
  - The cable is ~~not~~ Red/Yellow color coded.
- Xtown looks **like a wire** to software.
  - No NIC.
  - No visible registers.
- **One Xtown per pipe** in rack configurations.

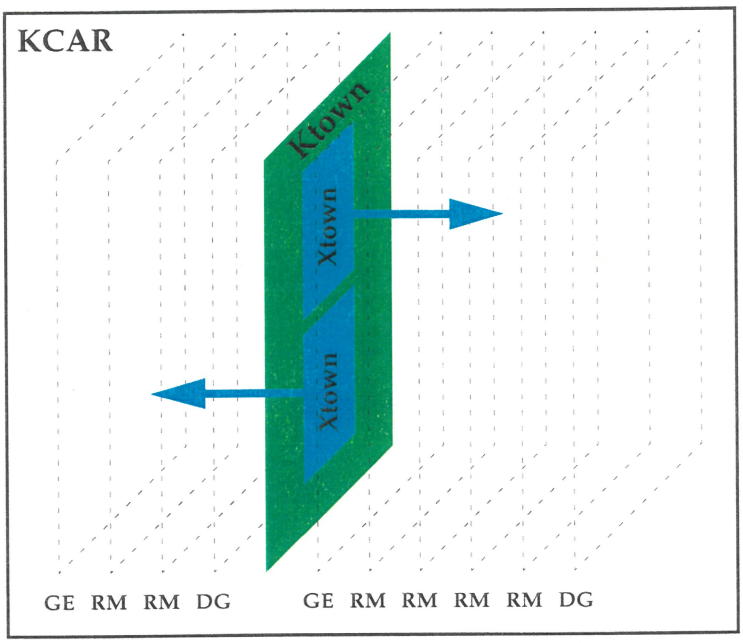


Student Notes

13-11 FOIL: KonaTown Board (Ktown)

- Functions as **two Xtowns** in the **KCAR** rack configuration.
- Upper link connection drives the **4-RM pipe** (to right).
- Lower link connection drives the **2-RM pipe** (to left).

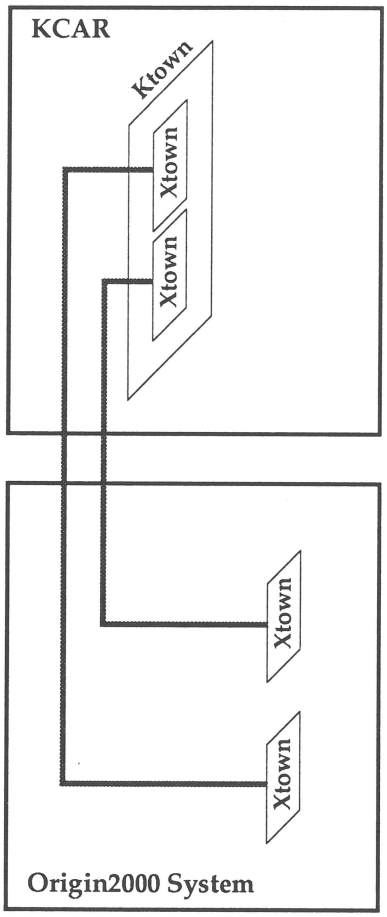
3-rm -false



Student Notes

13-12 FOIL: Single-rack Configuration

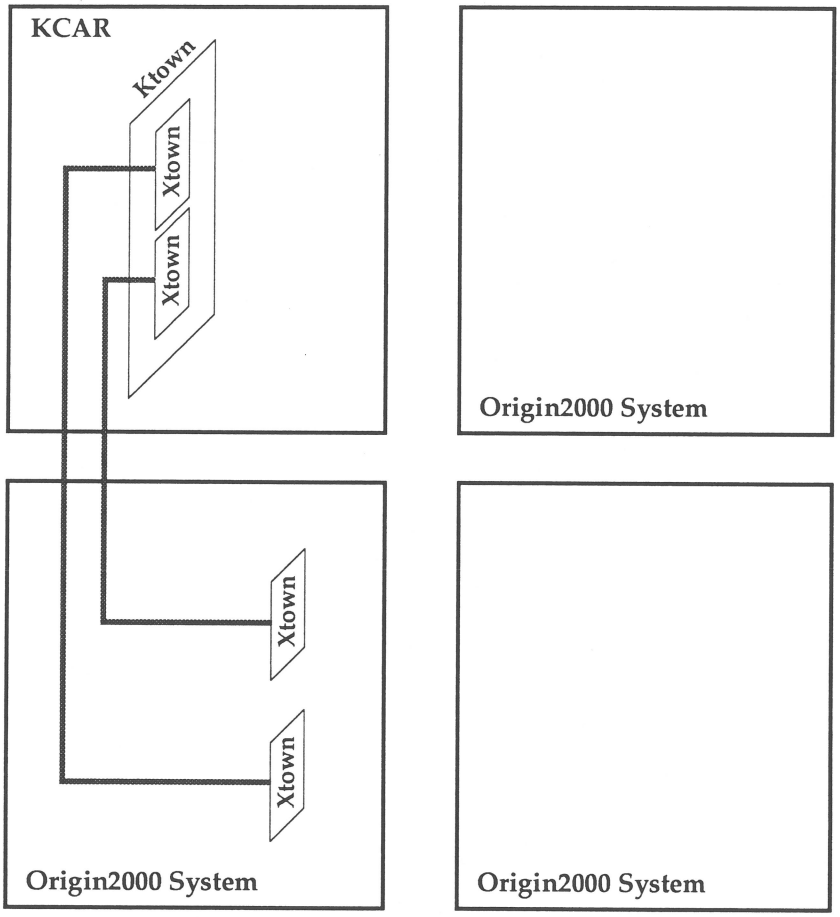
- 4, 6, or 8 CPUs (2, 3, or 4 nodes).
- Best to configure each pipe on a **separate** XBOW board.
- **Cannot** have more than one pipe per node.
- Probably best to use 4-RM side of KCAR first.



Student Notes

13-13 FOIL: Dual-rack Configuration - Maximum CPUs

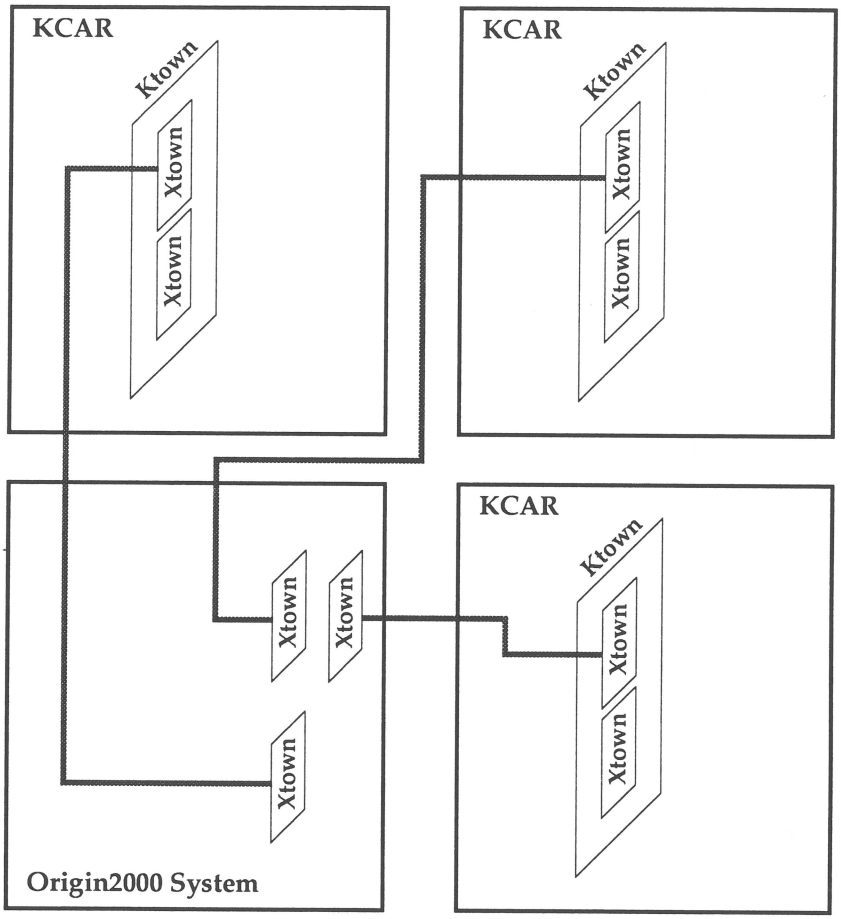
- 24 CPUs (12 nodes).
- 1 or 2 pipes.
  - One 2-RM pipe.
  - One 4-RM pipe.



Student Notes

13-14 FOIL: Dual-rack Configuration - Three 4-RM Pipes

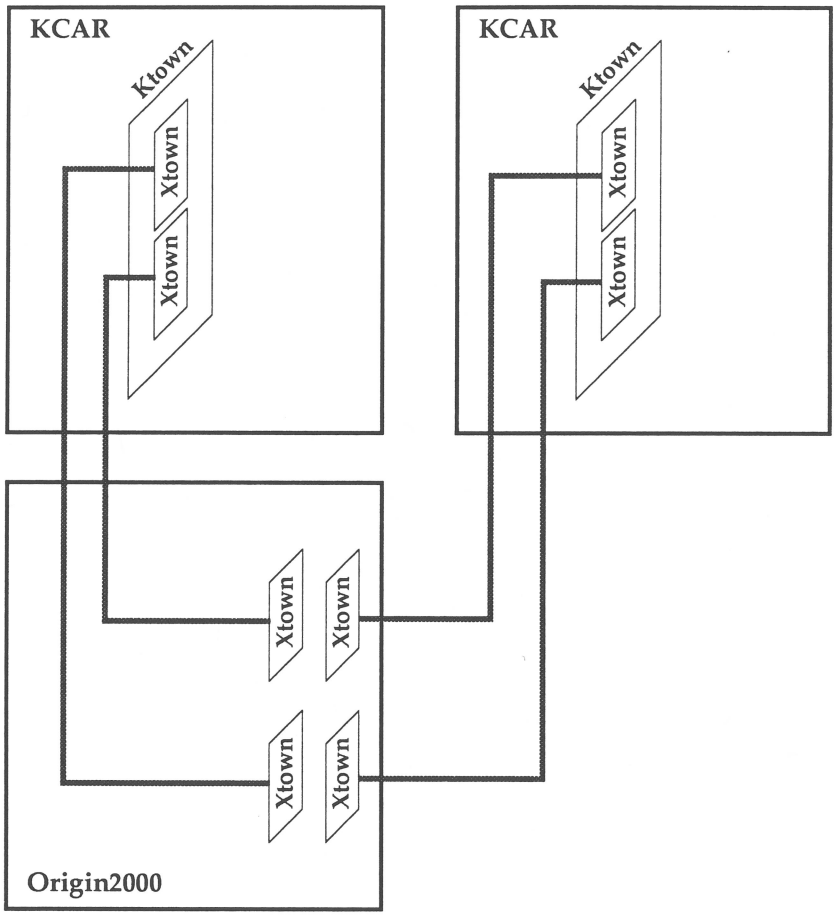
- 6 or 8 CPUs (3 or 4 nodes).
- One node per pipe.
- Three 4-RM pipes.



Student Notes

13-15 FOIL: Dual-rack Configuration - Maximum Pipes

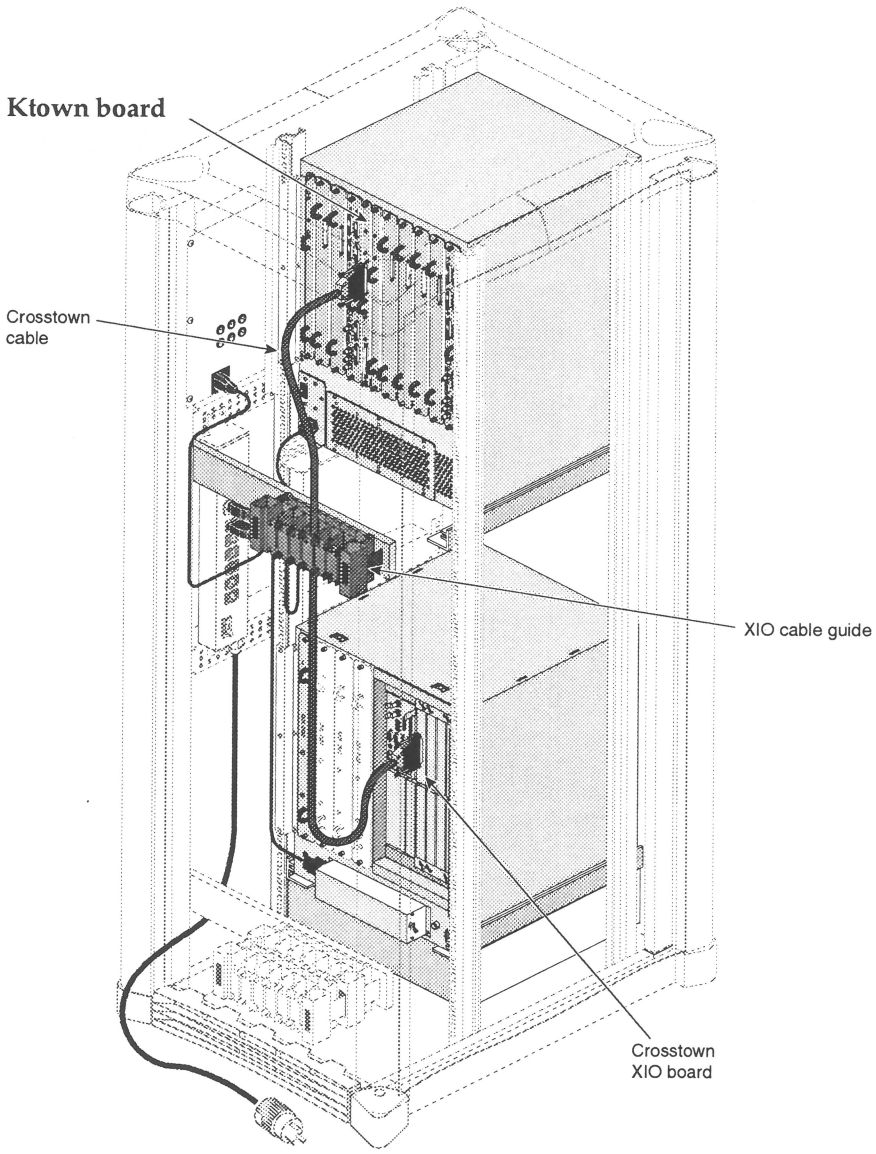
- 8 CPUs (4 nodes).
- One node per pipe prevents further expansion (will change in the future).
- Four pipes.
  - Two 2-RM pipes.
  - Two 4-RM pipes.



Student Notes

13-16 FOIL: Crosstown Cable

- Uses the same technology as a CrayLink cable.
  - Connects **Xtown** board to **Ktown** board.
  - Not used in deskside systems.
- The cable is ~~not~~ color coded. (YELLOW)



Student Notes

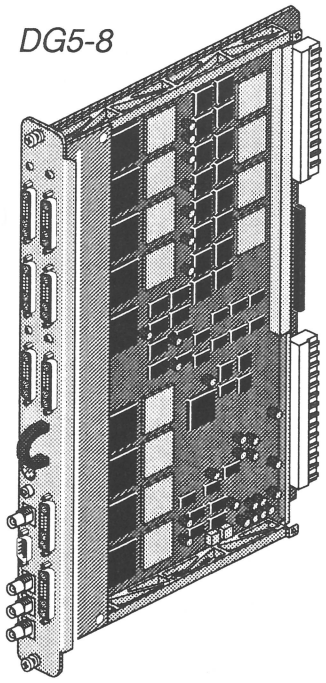
13-17 FOIL: Onyx2 Board Set

- **GE14** has two versions:
  - **2** Geometry Engines for Reality.
  - **4** Geometry Engines for InfiniteReality.
- **RM/TM** FRU pair has two versions:
  - **RM8** for Reality.
  - **RM7** for InfiniteReality.
- **DG5** has four versions:
  - **DG5-2** gives 2 output channels.
  - **DG5-8** gives 8 output channels.
  - **DG5-GV0** gives 3 output channels with **CCIR601 digital video**.
  - **DG5-DG0** (future release giving high-resolution video output).
- **Ktown** board, used in racks to connect the Origin™ chassis to the KCAR chassis.

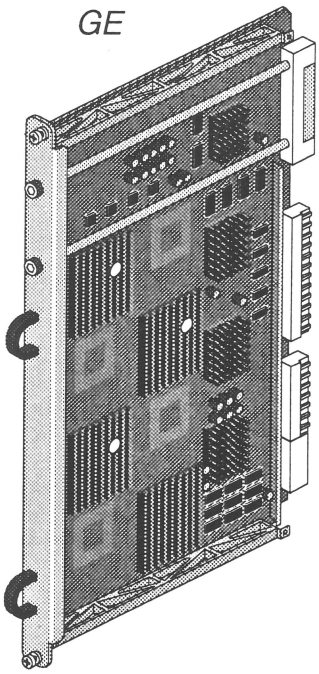
Student Notes

13-18 FOIL: Onyx2 Board Illustrations

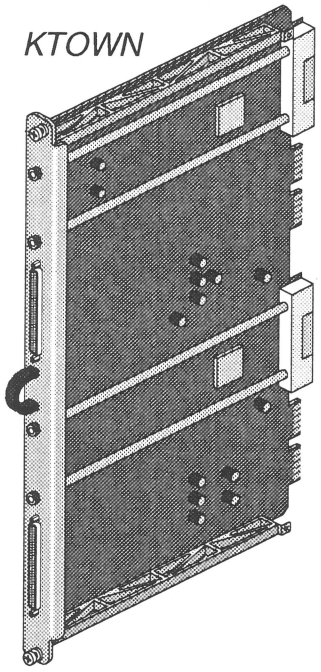
DG5-8



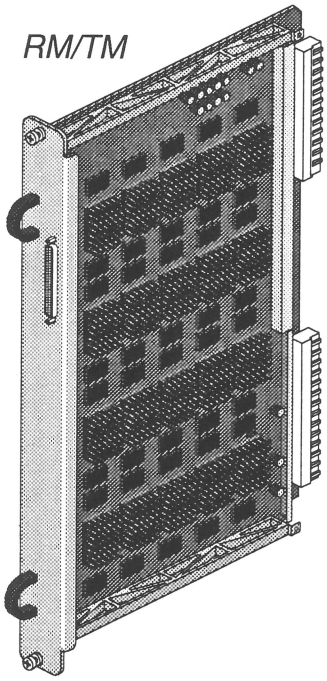
GE



KTOWN



RM/TM



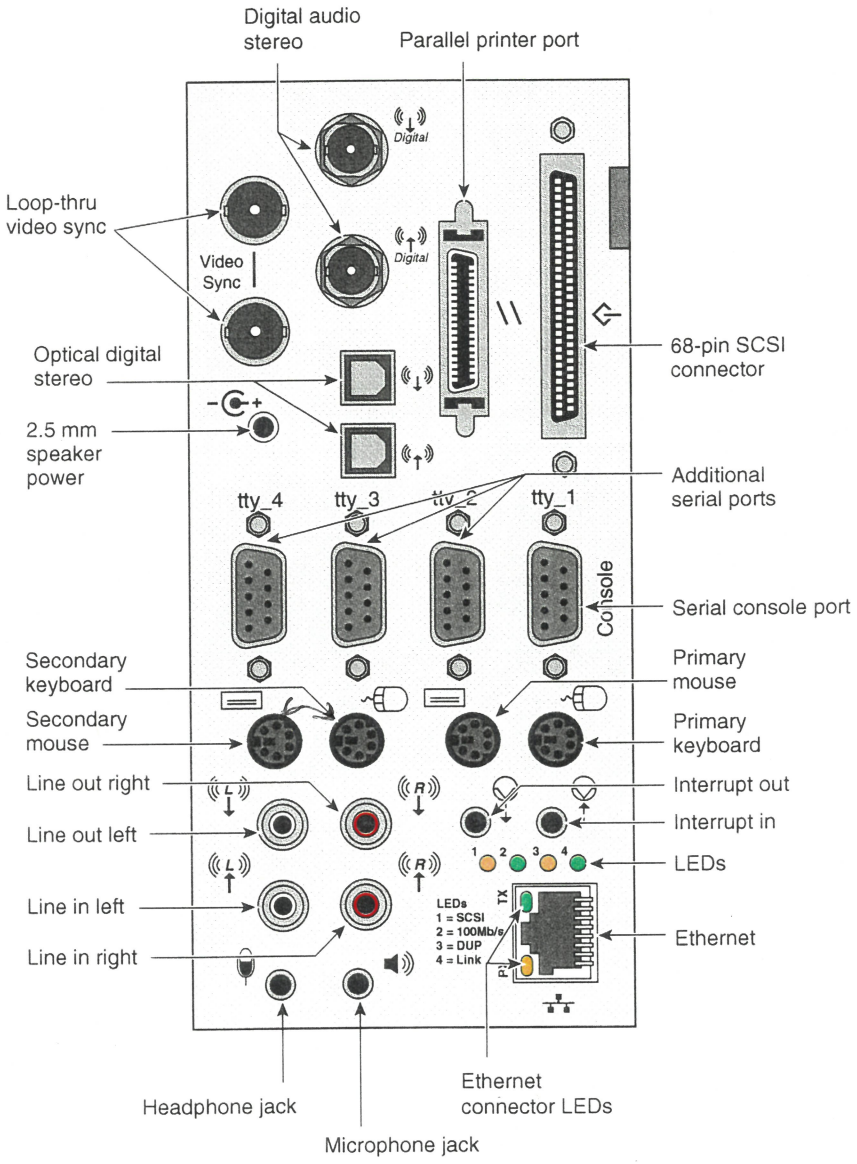
Student Notes

13-19 FOIL: IO6G

- Does everything the IO6 does.
- Two **Ultra SCSI single-ended** IO channels (busses).
  - Internal bus (**pci 0**).
  - External bus (**pci 1**).
- One **100 BaseT Ethernet™** port.
- **Four serial ports** (IO6 only has two).
- Offers customer **two-seat** capability.
  - Two keyboard connections.
  - Two mouse connections.
- **Stereo** line in and out.
- **Microphone** in.
- **Headphone** out.
- **Digital audio** (AES/EBU) in and out.
- **Optical digital audio** (ADAT).
- **Parallel printer port**.

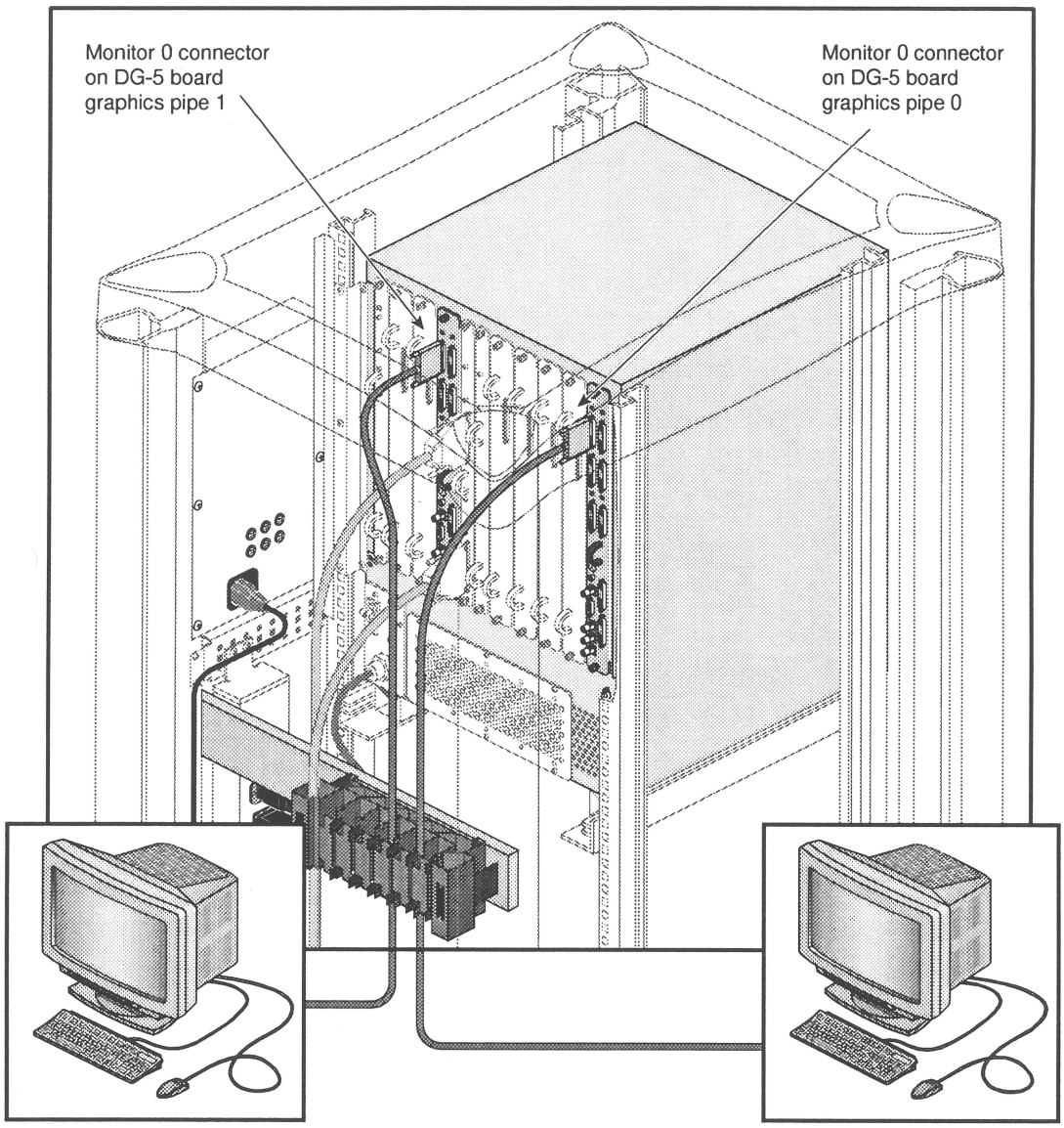
Student Notes

13-20 FOIL: IO6G Connections



Student Notes

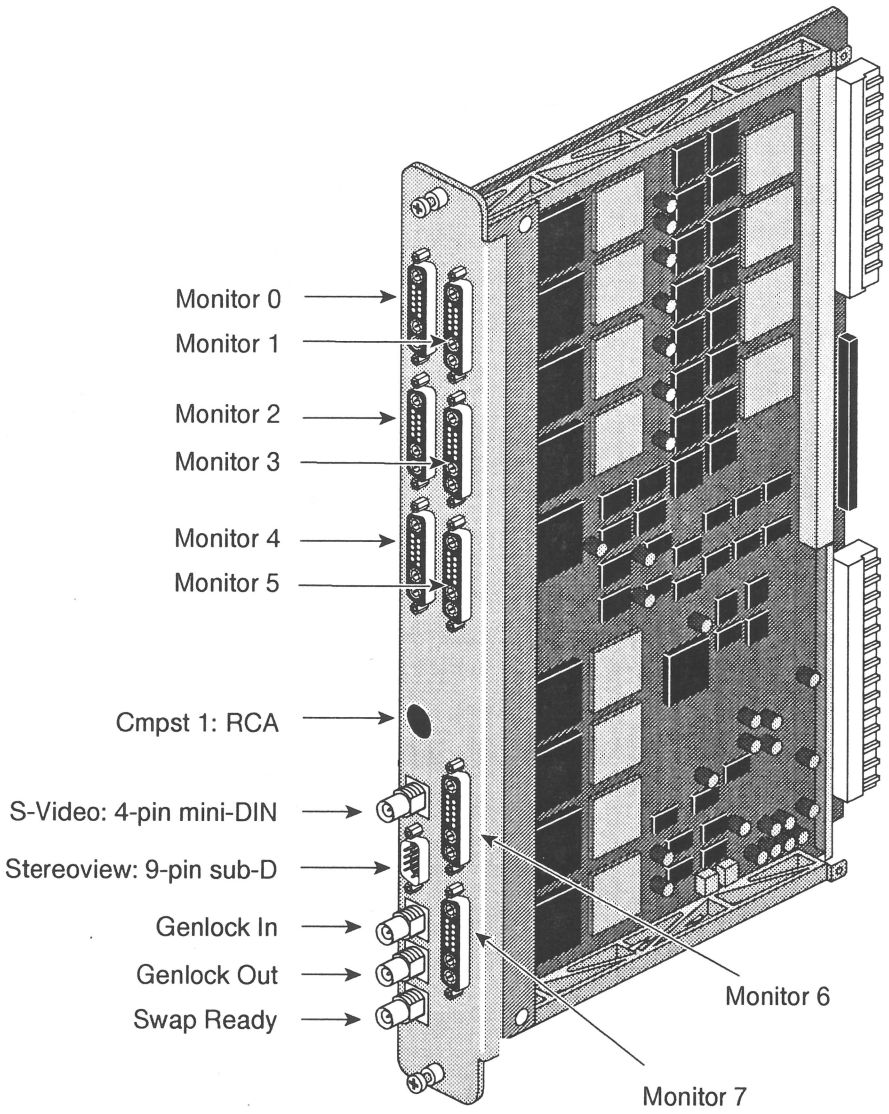
13-21 FOIL: GFX Monitor Connections



Student Notes

13-22 FOIL: DG5-8 Connections

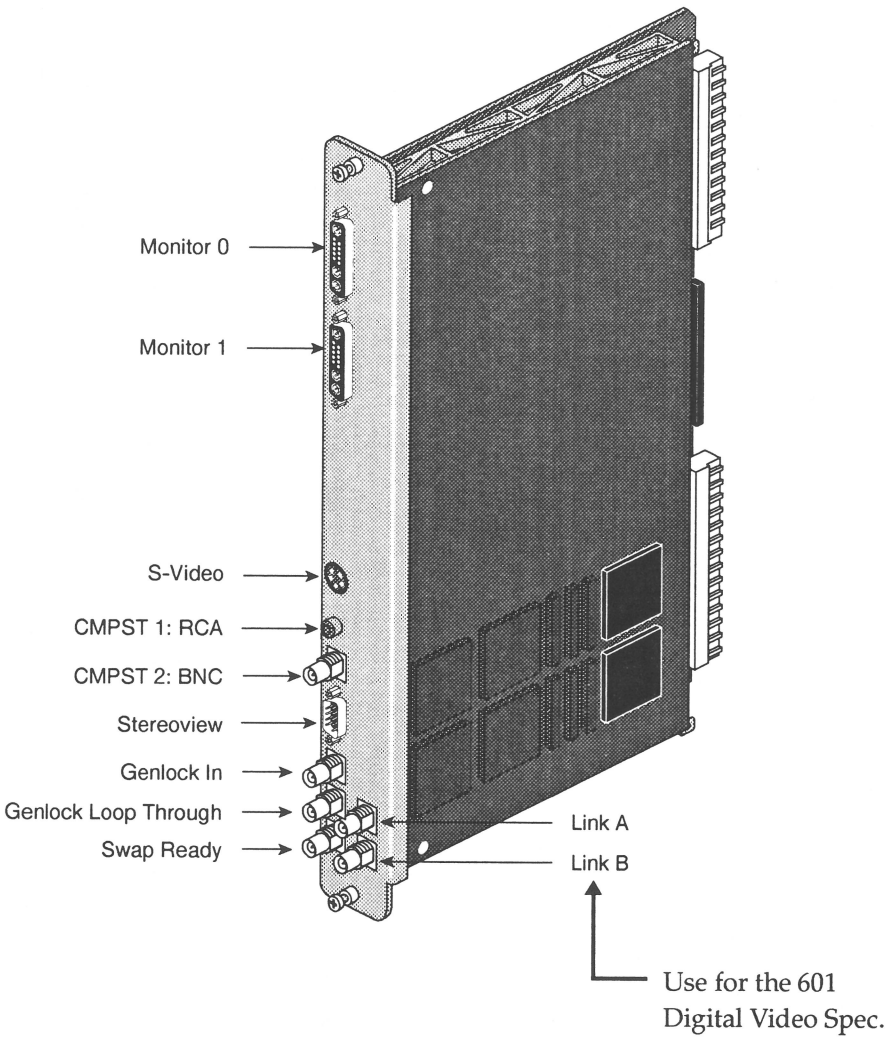
- DG5-2 functions like the DG5-8, but with 6 fewer RGB monitor channels.



Student Notes

13-23 FOIL: DG5-GVO Connections

- Graphics-to-video option.
- **Daughter board** installed on the DG5.
- Captures graphics from framebuffer and outputs to serial digital video.
- Used by post-production houses.
- Compliments the standard S-video of InfiniteReality.



Student Notes

13-24 FOIL: Multi-seat Onyx2 Configurations

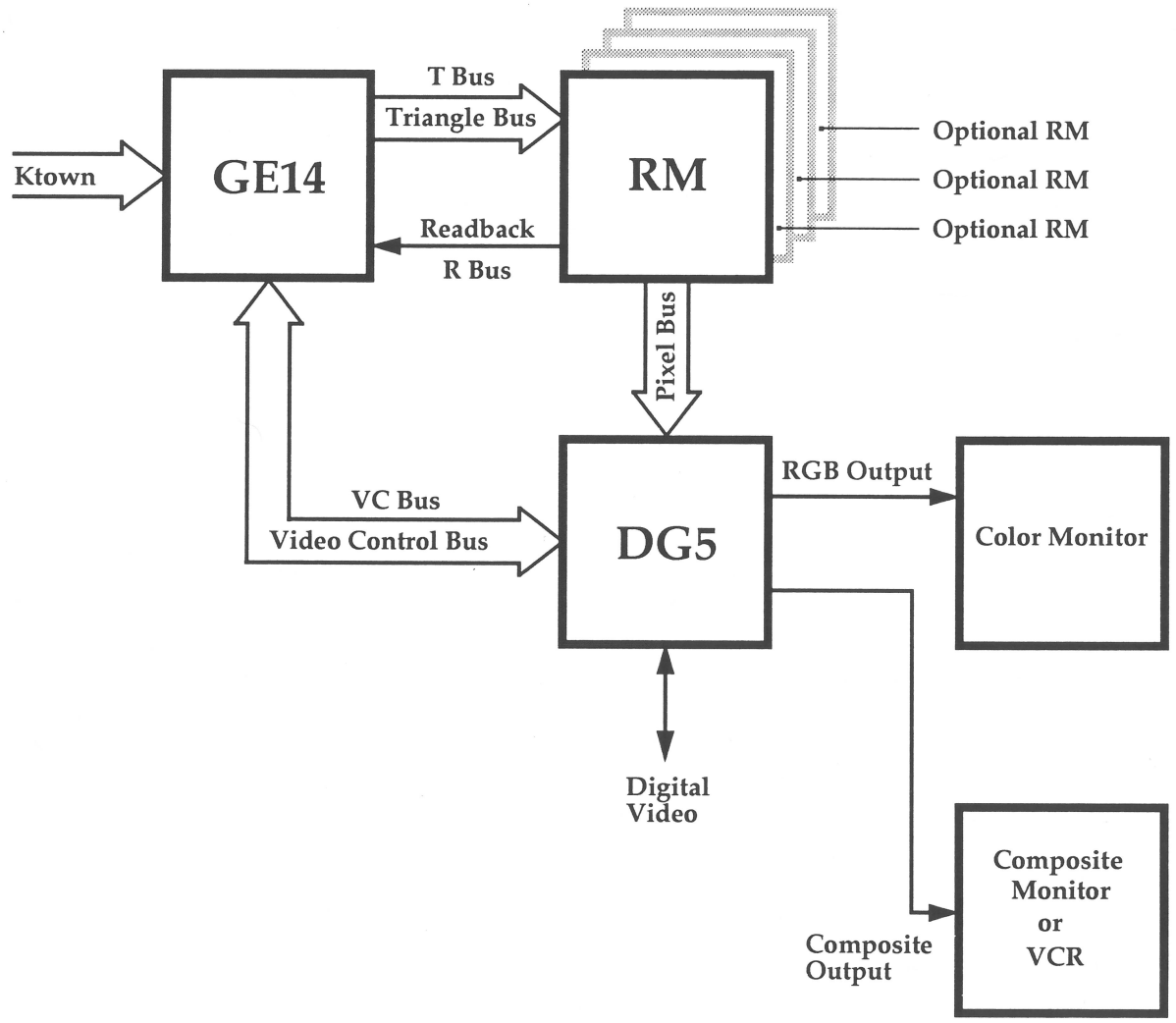
- Each **seat** (person) requires the following:
  - A graphics **keyboard**.
  - A **mouse**.
  - An **X-server**.
  - A graphics **pipe**.
- The above components are **absolute minimums**.
- The user **cannot** use the **multi-channel output** to drive more than one seat.
- An **X-server** runs on **multiple pipes**, but multiple X-servers **cannot** run on **one pipe**.
- Keyboards and mice connect to the system through the **IO6G**.
  - Each IOG6 allows **two** keyboards and mice.
  - Each Origin2000 system provides only **one** IO6G.
  - For every **two seats**, an **Origin2000** module is required.
- At MR (manufacturing release), each **pipe** requires a **separate node** (IP27).

  Note: This may change with future IRIX™ releases.

Student Notes

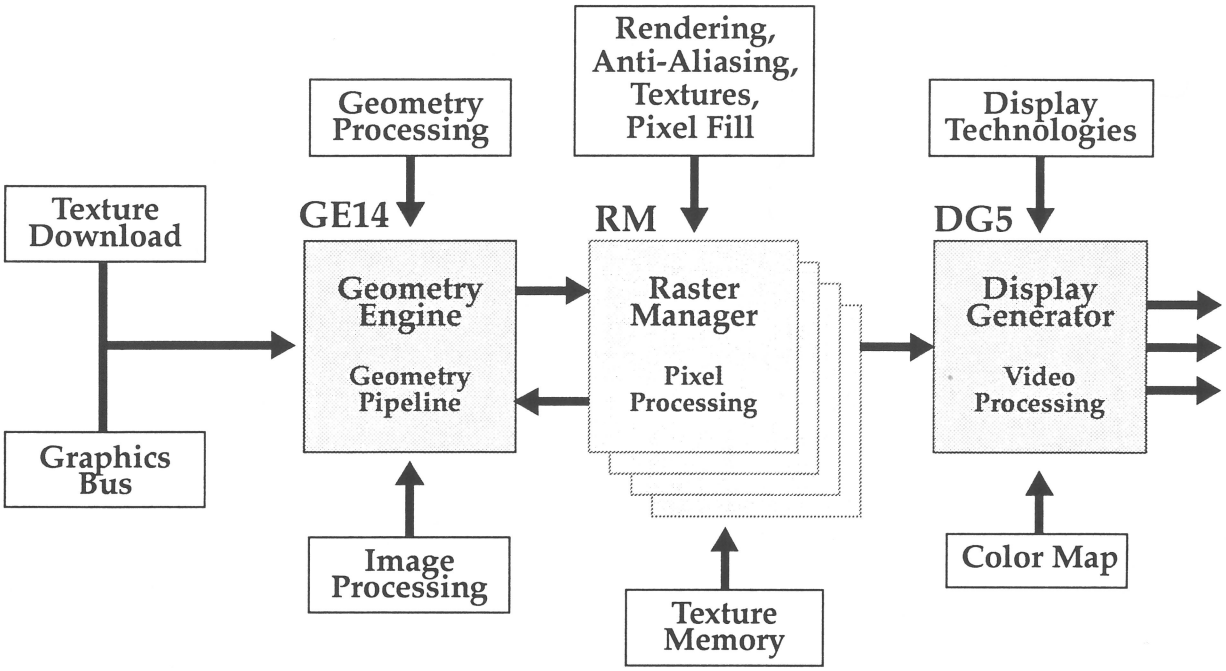
13-25 FOIL: Basic Block Diagram

- **T Bus**: Triangle data to RMs for filling.
- **R Bus**: Readback data from RMs to GE or Host.
- **VC Bus**: Video Control information (cursor, timing, color map filling).
- **Pixel Bus**: Pixels from RMs for display (no edge connector).
- **Jtag Bus**: Not shown, but built into the system. It is used by manufacturing to test the boards for connectivity and run **built-in self-tests** of the RAMs.



Student Notes

13-26 FOIL: Onyx2 Architecture



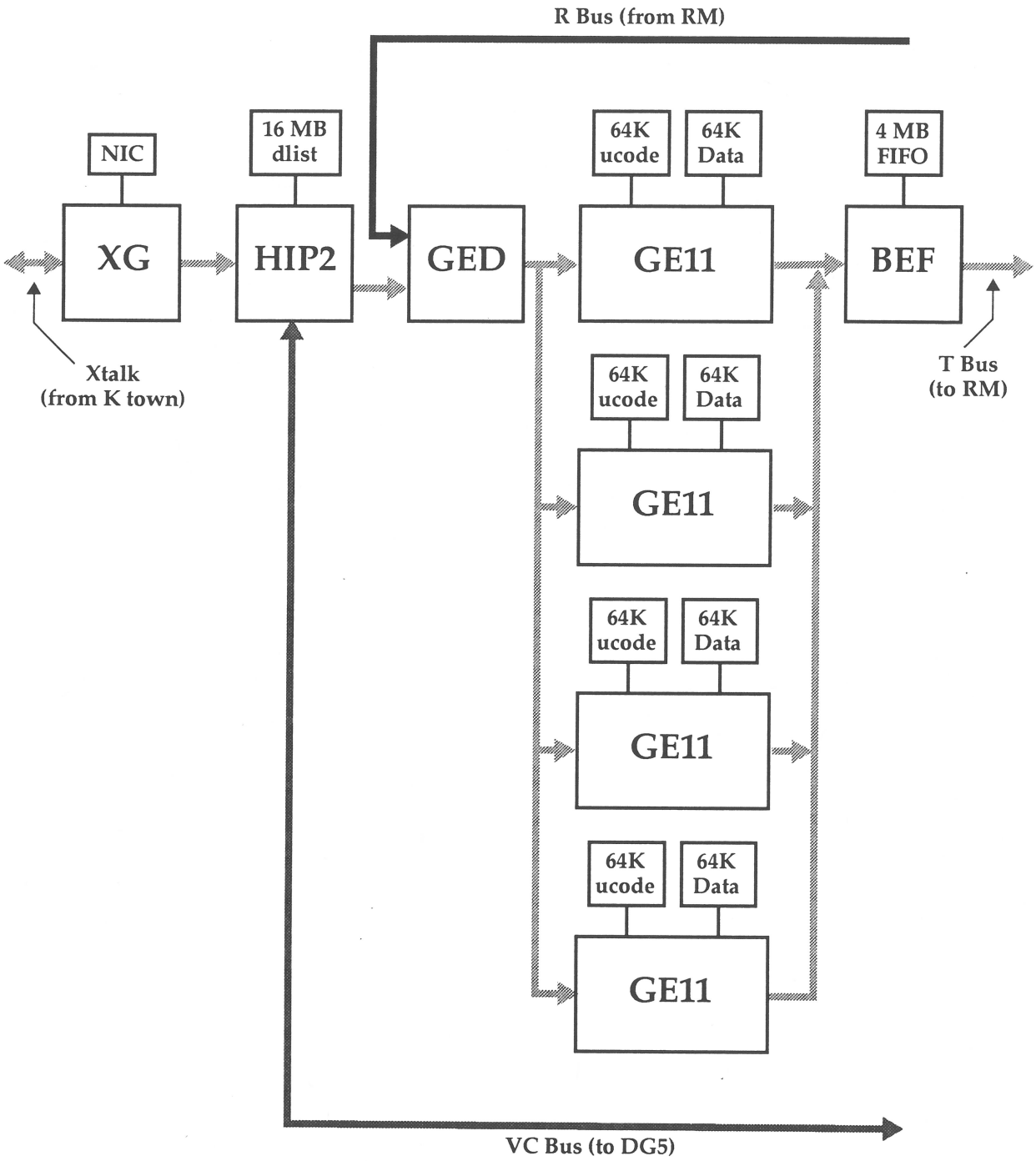
Student Notes

13-27 FOIL: GE14 Functions

- Two configurations:
  - **Reality** - 2 GE ASICs and RM8.
  - **InfiniteReality** - 4 GE ASICs and RM7.
- **GE14** receives vertex data and does the following:
  - **Performs Geometric Transformations**:
    - **Matrix calculations**.
    - **Lighting**.
    - **Clipping**.
    - Transforms **object-space coordinates** to **world-space coordinates**.
    - Maps **world-space coordinates** to **screen pixel coordinates**.
  - Performs **Image Processing** functions.
- **GE14 ASIC** Functions.
  - **XG**: converts Xtalk™ protocol to graphics protocol.
  - **HIP2**: Host Interface Processor:
    - 16 MB **display list** cache.
    - Video Control Bus.
      - Access to color maps on DG5.
      - Window display modes.
      - Cursor control.
  - **GED**: Geometry Engine Distributor:
    - Passes vertex data to idle Geometry Engines.
    - Receives data from framebuffer (R bus).
  - **GE11**: Geometry Engines:
    - Three floating point processors running at 80 MHz.
    - Floating point to fix number conversion for each FPU.
    - 256x32 output FIFO.
  - **BEF**: Back-end FIFO:
    - 4-MB FIFO.
    - Drives the Triangle Bus (T bus) to the RM6.

Student Notes

13-28 FOIL: GE14 Functional Block Diagram



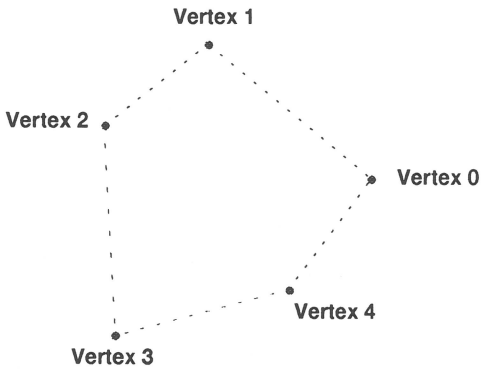
Student Notes

13-29 FOIL: Processing Geometric Vertex Data

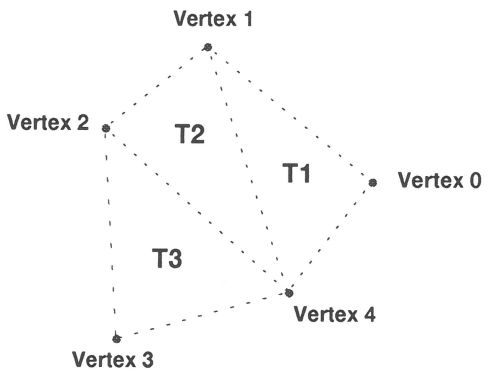
If polygons have more than 3 vertices, they are subdivided into **triangles**.

- Polygons may have up to 256 sides.
- **Meshing** of the polygons is advised to allow best performance.

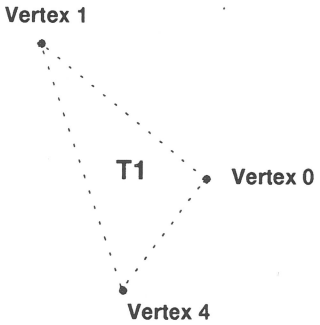
1. Assume a five sided polygon is received by a Geometry Engine, as defined by the five vertex points shown below. Each point is accompanied by location, orientation, color, and texture mapping coordinates.



2. The original polygon is divided into 3 small triangles by the Geometry Engine and then the transformations and lighting calculations are performed on the three vertices of each triangle.



3. The vertices of each triangle are passed to the Raster Manager, via the Triangle Bus, after all calculations are performed by the Geometry Engine. The other parameters that are used by the RM are also passed.



Each Vertex is accompanied by:

World space coord:	X, Y, Z
Texture Coord:	S, T
Color:	R, G, B
Transparency:	A
Normal:	N

→ **To Raster Manager**

The vertex points that leave the GE14 are mapped to screen pixel coordinates.

Student Notes

13-30 FOIL: RM Functions

- Scans and converts **triangle** data into **pixel data.**
- Performs **texture mapping.**
- Organizes data into **vertical spans** and hands it off to **Image Processors.**
- Fills the polygons and stores **screen pixel data** in **Image Buffer.**
- Contains the **FrameBuffer** Memory:
  - **80 MB** for **RM7** (FRU is a pair of boards).
  - **40 MB** for **RM8** (FRU is a single board).
- RM **ASIC** Functions:
  - **PG**: Pixel Generator.

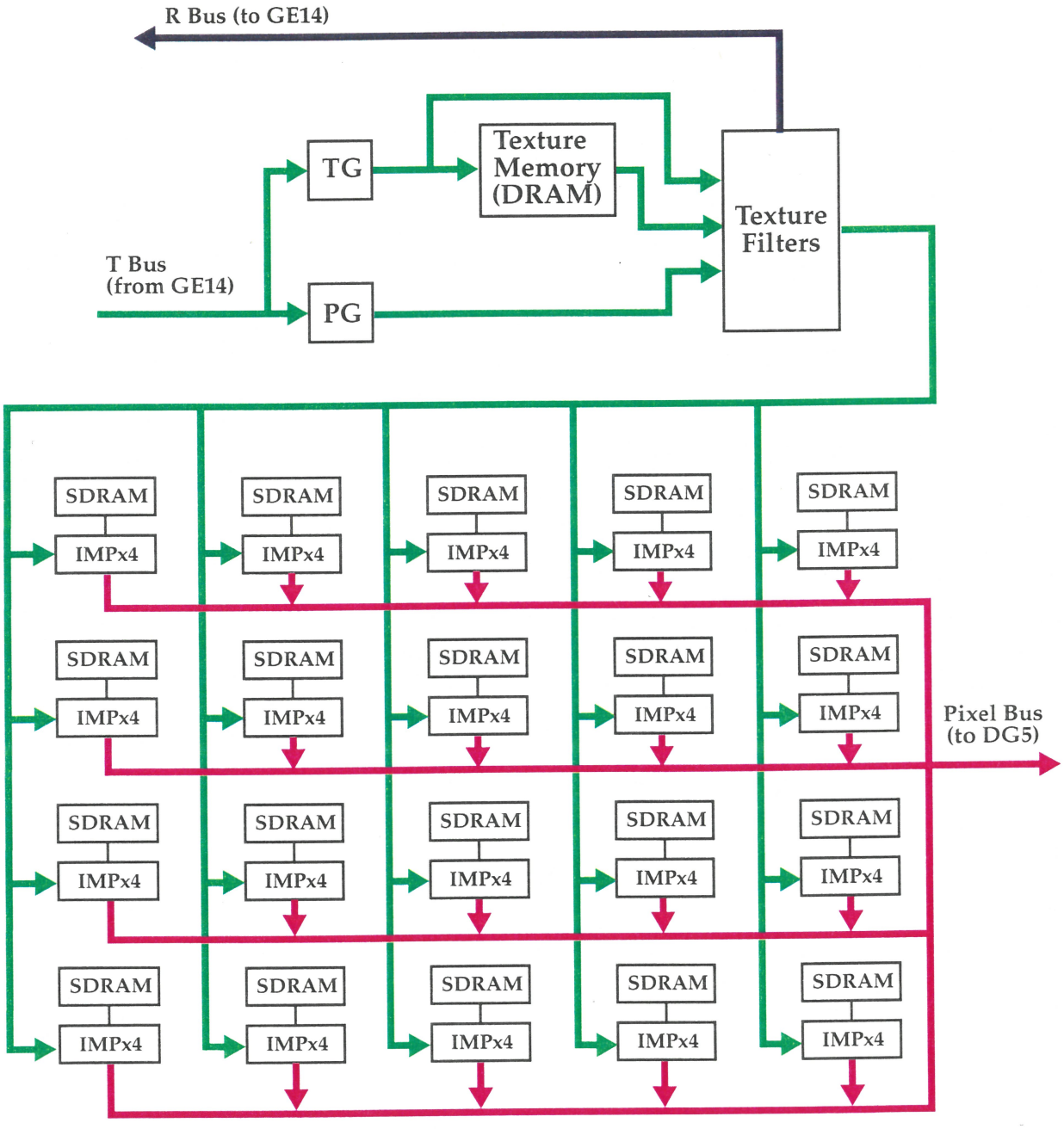
    Identifies **outside edges** of **triangle** and calculates the **top** and **bottom** pixel in a **vertical** span and passes data to **texture filter.**
  - **TG**: Texel Generator.
    - Performs **texture mapping** and passes data to **texture filter.**
    - **16 MB** or **64 MB** texture memory.
      - Textures replicated across boards for speed.
      - Additional RM boards do not add more texture memory; they just make it go **faster.**
  - **Texture Filter.**

    Combines the output of **PG** and **TG** and passes data to **IMP.**
  - **IMP**: Image Memory Processor.
    - **RM7** contains **20 IMP8** ASICs, which contain 4 IMPs (20 x 4 = 80 IMPs total per RM).
    - **RM8** contains **10 IMP9** ASICs, which contain 4 IMPs (10 x 4 = 40 IMPs total per RM).

      Gives a slower fill rate than RM7.
    - Perform **Z-buffering**.
    - **Alpha blending, fog**, and **anti-aliasing.**
- **RM7** and **RM8** boards **cannot** be mixed.

Student Notes

13-31 FOIL: RM7 Block Diagram

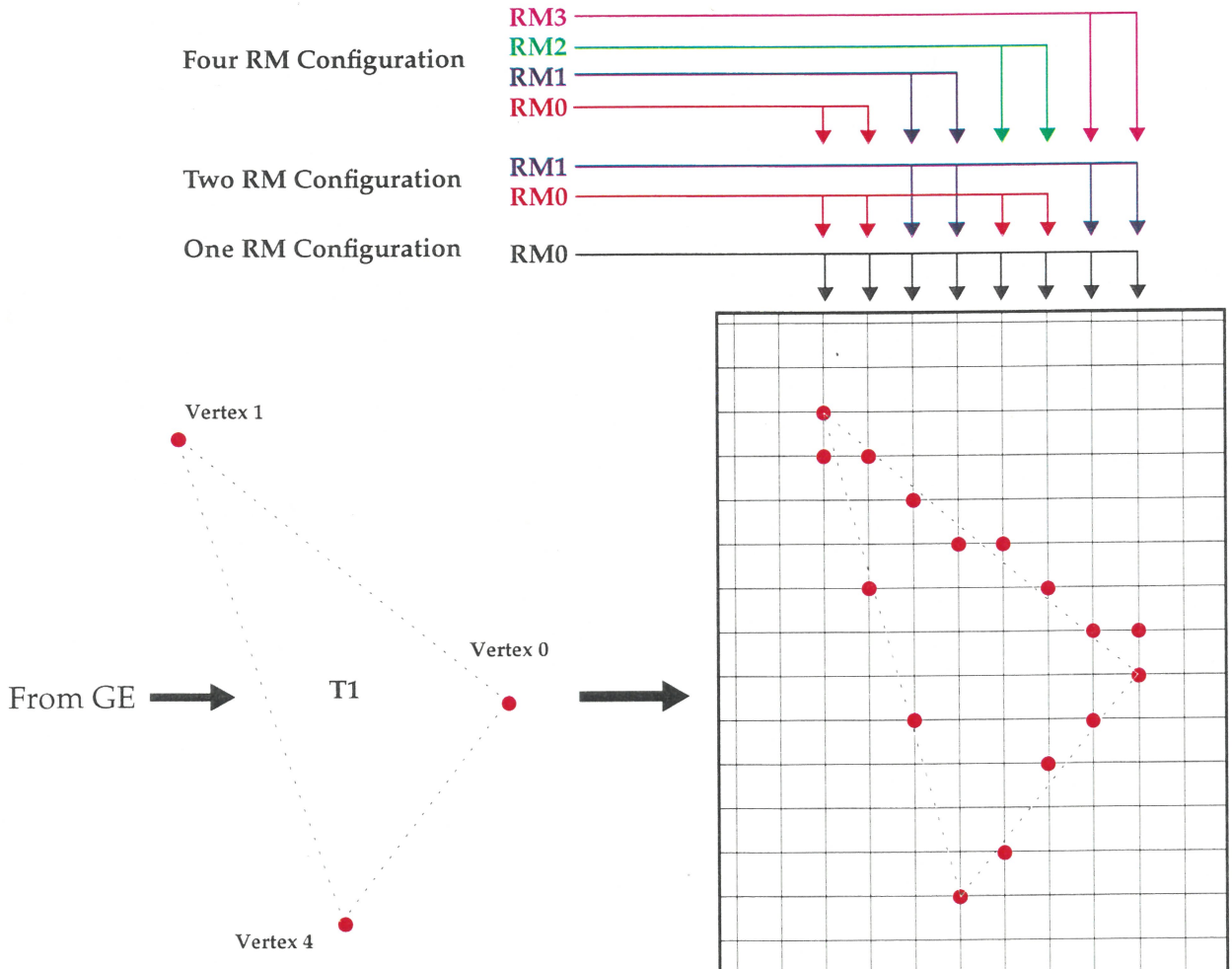


Student Notes

13-32 FOIL: Edge Interpolation

- **Triangles** are fed to the Pixel Generator (**PG**) from the Geometry Engines.
- The **PG** walks the edges of the triangle, identifying which pixels lie along those edges.
- The **PG** then identifies **pixel pairs** on the **vertical spans** of the outside edges of the triangle.

  Note: Each RM board controls **two vertical spans**, interleaved with the other RM boards, beginning at **X0 and X1** and spanning the screen to **X1278 and X1279**.



Each Vertex is accompanied by:

World space coord:	X, Y, Z
Texture Coord:	S, T
Color:	R, G, B
Transparency:	A
Normal:	N

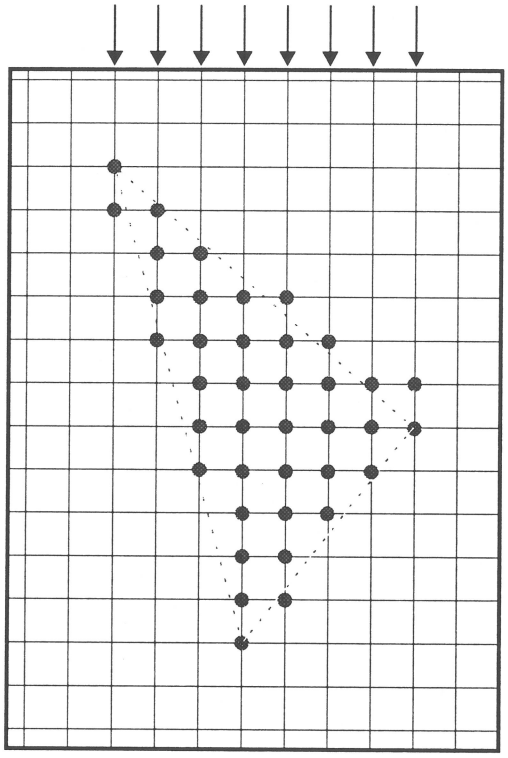
The PG identifies pixel pairs of each vertical span along the edges of the triangle.

Student Notes

13-33 FOIL: Fill the Triangle

- The **PG** selects all the pixels along the vertical spans.

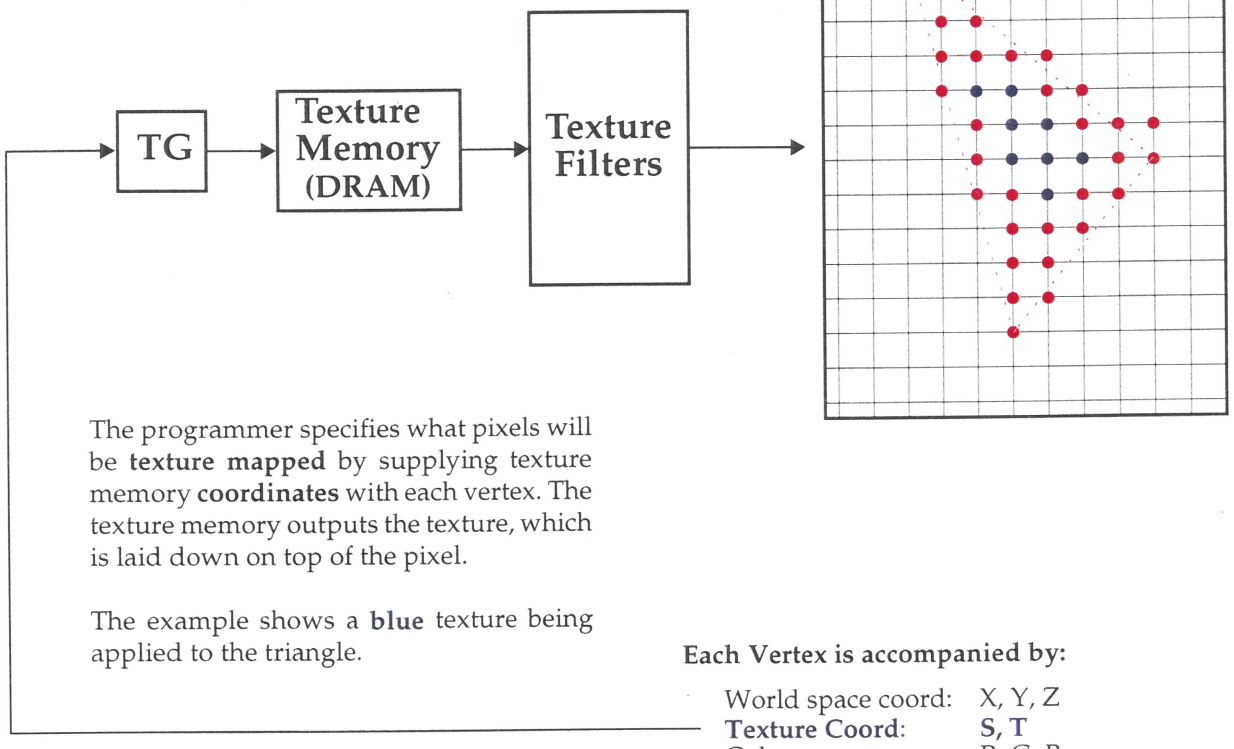
The PG identifies pixels along the spans, from the top pixel to the bottom pixel and calculates all the parameters for each pixel.



Student Notes

13-34 FOIL: Texture Mapping

- To facilitate performance, there are two paths from the PG to the next stage:
  - One path is utilized when the polygons are **texture mapped**, and the texture memory must be accessed.
  - The other path **bypasses** the **texture generator** and **memory** when texturing is not specified on a pixel.
- **Texturing mapping** lays a texture over specified pixels.



Student Notes

13-35 FOIL: Image Memory Processor

- Each **IMP** controls 4-megabits of associated SDRAM (**image memory** and all the associated bit planes).
- The **IMPs** support both **8-bit** and **12-bit** color components for pixels.
- The main responsibilities of the **IMPs** are to:
  - Decide whether to write the pixel based upon the **Z-buffer** value (**hidden surface removal**).
  - Perform the **alpha** blending of each new pixel with the pixel value that has already been rendered at the same location.
  - Perform **fog** function.
  - Perform **anti-aliasing**.
  - Send the digital pixels over the **Pixel Bus** to the DG5.

Student Notes

13-36 FOIL: DG5 Functions

- Takes **digital output from the FrameBuffer**, performs **gamma correction**, and converts it to an **analog** signal for display.
- Supplies timing and support for the various video formats: **NTSC**, **PAL**, **HDTV**, **Genlock**, etc.
- Performs **Dynamic FrameBuffer Resizing.**

  This allows the programmer to draw a frame using low resolution; the DG5 will blow it up and output it to a monitor with a higher resolution.

  The output becomes grainy but does not detract from the scene.
- Two **RGB** outputs are standard.
- Up to six additional **RGB** outputs are optional.
- DG5 total pixel output at **300-million** pixels per second.

  Example calculations:

  - With (2 monitors) x 1280 x 1024 x 72 = **188,743,680** pixels per second.
  - With (8 monitors) x 1280 x 1024 x 72 = **754,974,720** pixels per second.

    This exceeds 300 million; therefore, to use up to 8 monitors, a combination of lower-resolution and higher-resolution monitors must be used to keep within the 300-million pixel limit.
- DG5 ASIC Functions:
  - **FM2** ASIC.
    - Acts as the functional manager of the DG5.
    - Interface for the Video Control bus from GE14.
  - **XMAP** ASIC.
    - Takes framebuffer pixels and streams them onto the video packet bus.
    - Controls cursor functions.
    - **Color-map** and Gamma-correction look-up-tables (**LUT**).

Student Notes

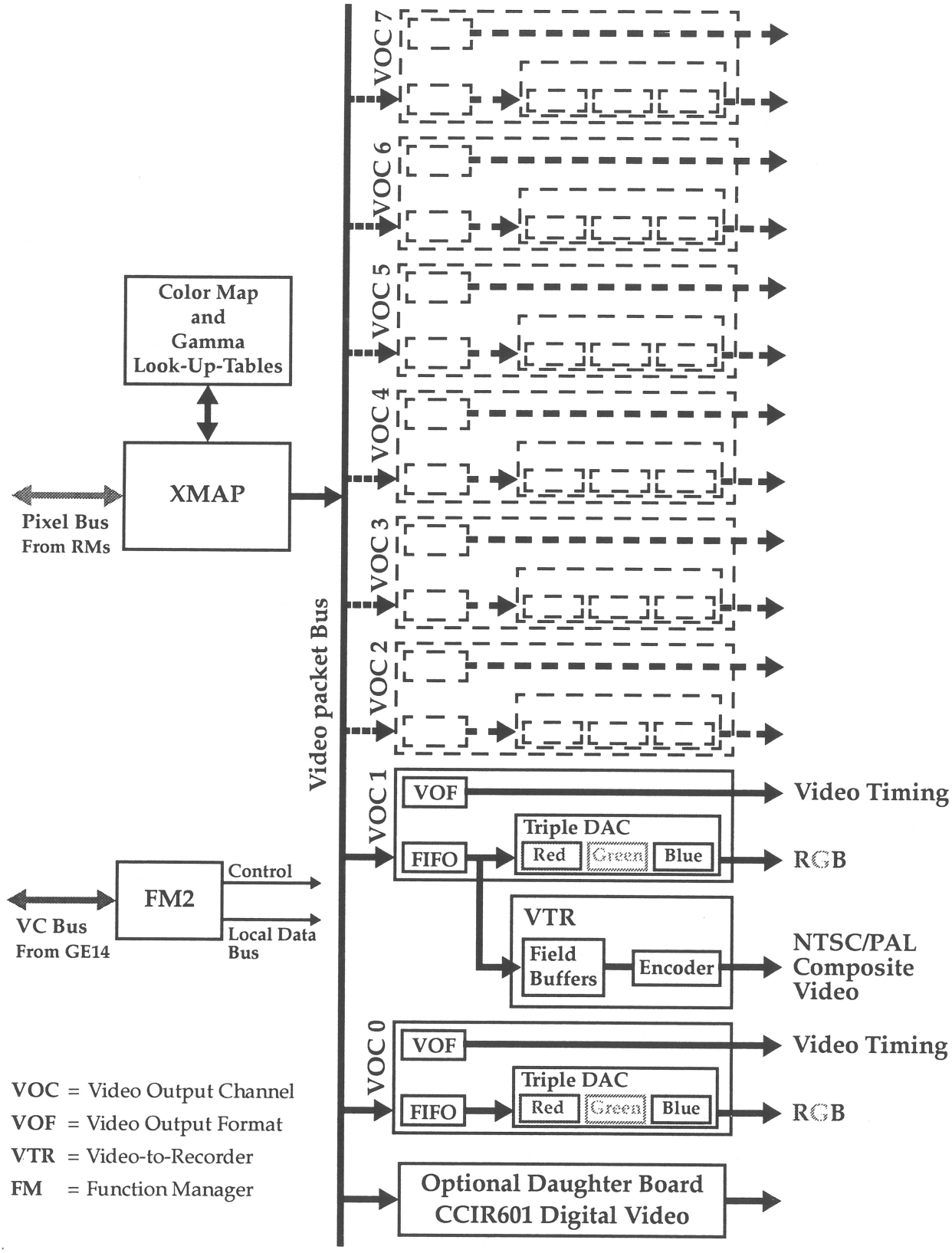
13-37 FOIL: DG5 Functions (continued)

- **XMAP** ASIC (continued).
  - Outputs to:
    - Video Output Channel (**VOC**).
    - NTSC or PAL encoder (**VTR** channel).
- Video Output Channel (**VOC**).
  - Two channels are standard for dual-screen applications or high-resolution head display (virtual reality head gear).
  - Performs the **Dynamic FrameBuffer Resizing** function.
  - Three-channel and eight-channel versions are optional, replacing the need for MCO (multi-channel option).

    The 3-channel version gives the **CCIR601 Digital Video** output.
    - **VOC** channels **0** and **2 - 7** contain the following functions:
      - Video timing (**VOF**).
      - FIFO buffer.
      - Triple DACs.
    - **VOC** channel **1** contains the following functions:
      - Video timing (**VOF**).
      - FIFO buffer.
      - Triple DACs.
      - Video-to-Recorder (**VTR**) channel.
    - Video-to-Recorder Channel (**VTR**).
      - Encodes video output into **NTSC** or **PAL** format.
      - Is part of the data path of **VOC** channel **1**.
      - VOC #1 can output in RGB or Encoded format, **but not both.**
      - User selects which function via the **ircombine** tool.

Student Notes

13-38 FOIL: DG5 Block Diagram

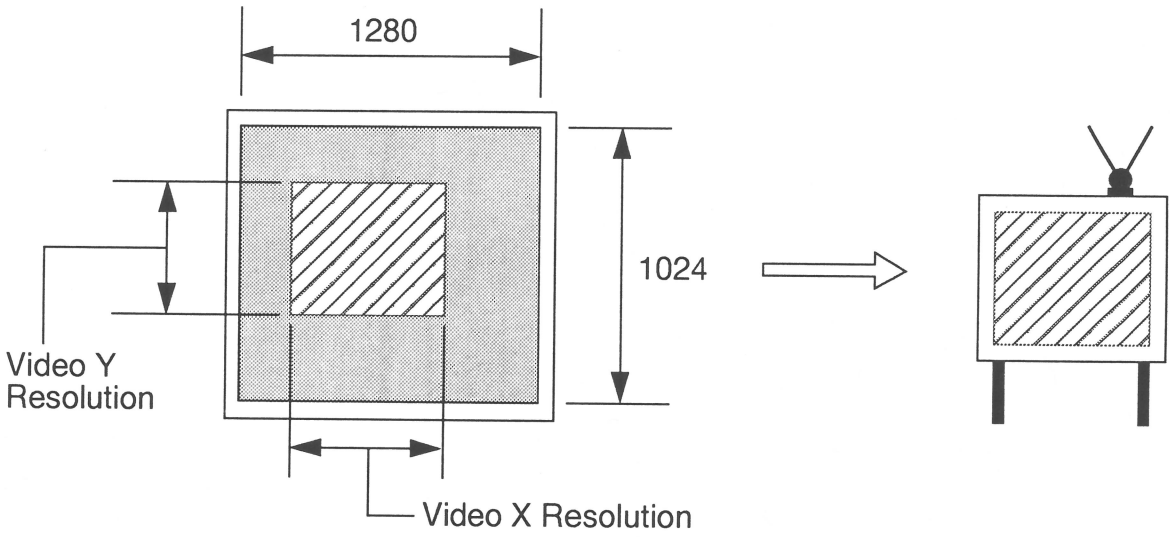


Student Notes

13-39 FOIL: Composite Video

Ircombine allows a roamable NTSC window that may be placed anywhere on the screen.

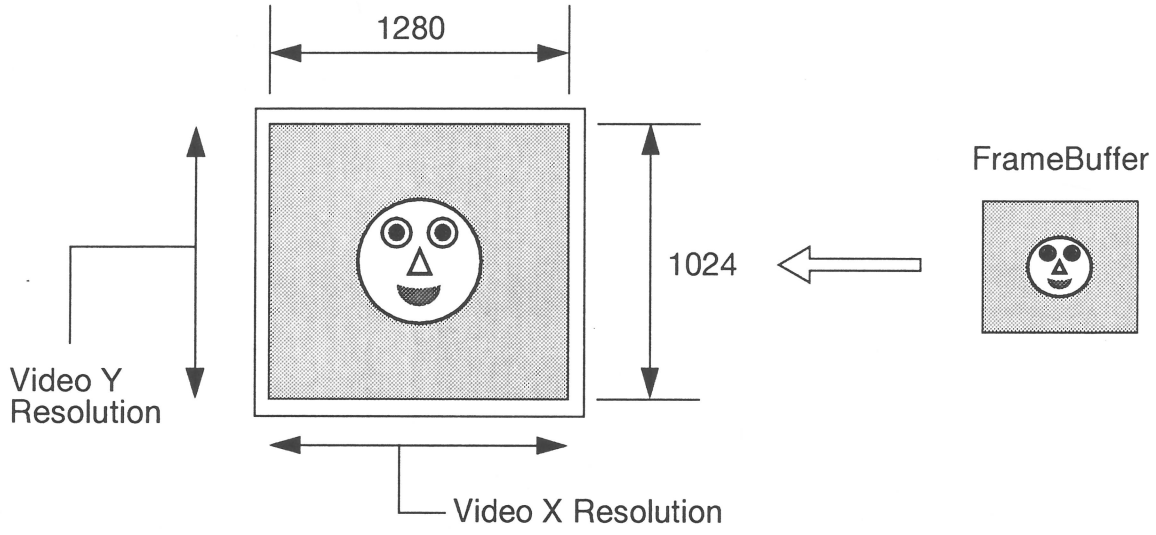
Whatever is drawn into the FrameBuffer within this roamable window is converted from its component (**RGB**) signal to a **composite** (TV) signal in real-time and may be connected directly from the system to a VCR.



Student Notes

13-40 FOIL: Dynamic Framebuffer Resizing

- This allows the programmer to draw a frame using low resolution; the DG5 will blow it up and output it to a monitor with a higher resolution.
- The output becomes grainy but does not detract from the scene.

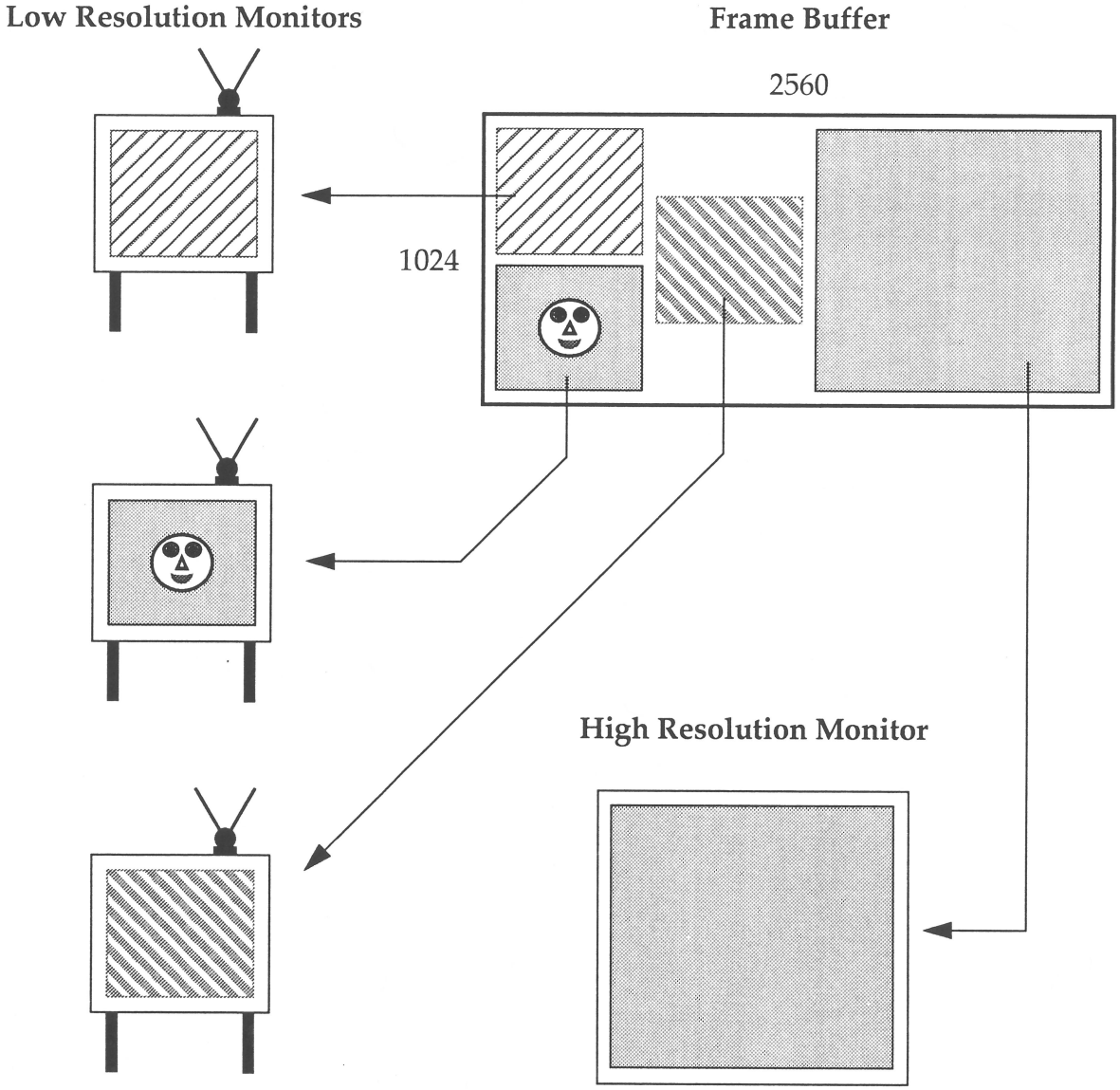


Student Notes

13-41 FOIL: Multiple Output Channels

- Allows the pipeline to provide up to **eight** independent channels of video output.
- Takes the digital output coming off the pixel bus from the XMAPs.
- A mix of **high**-resolution and multiple **lower**-resolution displays are possible.

Note: Make sure not to exceed the **300-million** pixel transfer rate (see page 71).



Student Notes

13-42 FOIL: Software Tools

- Requires IRIX 6.4.
- Collect **status** of graphics system:
  - **hinv**
  - **/usr/gfx/gfxinfo**
- Manual **starting** and **stopping** of the graphics system:
  - Stop gfx: **/usr/gfx/stopgfx**
  - Load microcode: **/usr/gfx/gfxinit -v**
  - Start gfx: **/usr/gfx/startgfx**
- Use **ircombine** to configure the DG5 output channels.
- Use **irsaudit** to diagnose the graphics subsystem.

  IRIEEPROM

  Note: Do not forget the **man** pages.

Student Notes

13-43 FOIL: System Boot Process

From **power on** to **multi-user mode**, several processes are involved during the boot-up and initialization of the system:

- IP27 power-on diagnostics.
- PROM monitor.
- SASH.
- Kernel boot process.
- Auto-configuration process.
- Init process.

Student Notes

13-44 FOIL: Init Process

- The IRIX operating system is always at a specific **run** level.
  - The **init** process controls what run level is entered.
  - **init** is the second process started at boot-time (**sched** is the first).
  - **init** brings up the IRIX operating system by running command scripts (or processes) that are specified in **/etc/inittab**.
  - **init** continues to run in the background after the system boots up.

Entries in inittab are in the form:

id:rstate:action:process

Field	Description
id	Name of the entry.
rstate	The run state. When the value is equal to the run state being entered, init runs the specified command.
action	A statement that qualifies or limits the action of the specified command.
process	The specified command.

For example: **s2:23:wait:/etc/rc2**

- See man page INIT(1M) for a full description of **init**.
- See man page INITTAB(4) for a full description of **inittab**.

Student Notes

The lecture only discusses the portion of the boot process that performs the **initialization** of the graphics subsystem that begins when the **/etc/rc2** process runs the **/etc/rc2.d/S98xdm** process.

Detailed information concerning the boot process is covered in the **SSE System Administration** course which is a **prerequisite** for this course.

13-45 FOIL: /etc/inittab

- When **init** is initially started, it scans **/etc/inittab** for the statement **initdefault**.
  - If **initdefault** exists, the specified run level will be entered.
  - If **initdefault** does not exist, the boot process stops and prompts the user to supply the initial run level.
- By **default**, all SGI® systems are set to enter **run level 2**.
- The **init** process scans **/etc/inittab** and runs all the commands whose specified run states (**rstate** field) correspond to the run level being entered.
  - The **order** of the entries in **inittab** defines the order of execution.
  - Commands with **rstate** fields that are empty are run when a new run level is entered unless the **action** field prohibits it.
- The following commands are run when **run level 2** is initially entered: **/etc/bcheckrc**, **/etc/brc**, **/etc/lnsyscon**, **/etc/rc2**, and any specified **getty** process.

/etc/inittab

Initial run level = 2

```
is:2:initdefault:
fs::sysinit:/etc/bcheckrc </dev/console >/dev/console 2>&1
mt::sysinit:/etc/brc </dev/console >/dev/console 2>&1
link::wait:/etc/lnsyscon > /dev/console 2>&1 < /dev/null
s0:06s:wait:/etc/rc0 >/dev/console 2>&1 </dev/console
s1:1:wait:/etc/shutdown -y -iS -g0 >/dev/console 2>&1 </dev/console
s2:23:wait:/etc/rc2 >/dev/console 2>&1 </dev/console
s3:3:wait:/etc/rc3 >/dev/console 2>&1 </dev/console
s4:06:wait:/etc/umountfs > /dev/console 2>&1
of:0:wait:/etc/uadmin 2 0 >/dev/console 2>&1 </dev/console
RB:6:wait:/etc/init.d/announce restart
rb:6:wait:/etc/uadmin 2 1 >/dev/console 2>&1 </dev/console
tp:23:off:/sbin/getty tport co_9600          # textport
t1:23:respawn:/sbin/getty ttyd1 co_9600      # alt console
t2:23:off:/sbin/getty -N ttyd2 co_9600       # port 2
```

Note: For illustration, the **inittab** file is compressed: non-essential information has been excluded.

Student Notes

13-46 FOIL: Starting the Graphics Subsystem

- The last process run by **/etc/rc2** is **/etc/rc2.d/S98xdm**.
- The **S98xdm** process performs the following operations:

  1. Checks if **/usr/bin/X11/xdm** is present.
  2. If **xdm** is present, checks if the configuration state variable **windowsystem** is **on** and if **/usr/bin/X11/Xsgi** is present,

     or

     If the configuration state variable **xdm** is **on**.
  3. If **1** and **2** are both true, **S98xdm** starts the **/usr/bin/X11/xdm** process as a daemon.

     Note: If **1** and **2** fail, the **operating system will come up**, but the graphics system functional microcode will not be loaded. You could log in at an ASCII terminal and attempt to isolate the problem.

- **xdm** is the **X display manager**.
- **Xsgi** is the **X window server**.

Student Notes

13-47 FOIL: Configuration State Variables

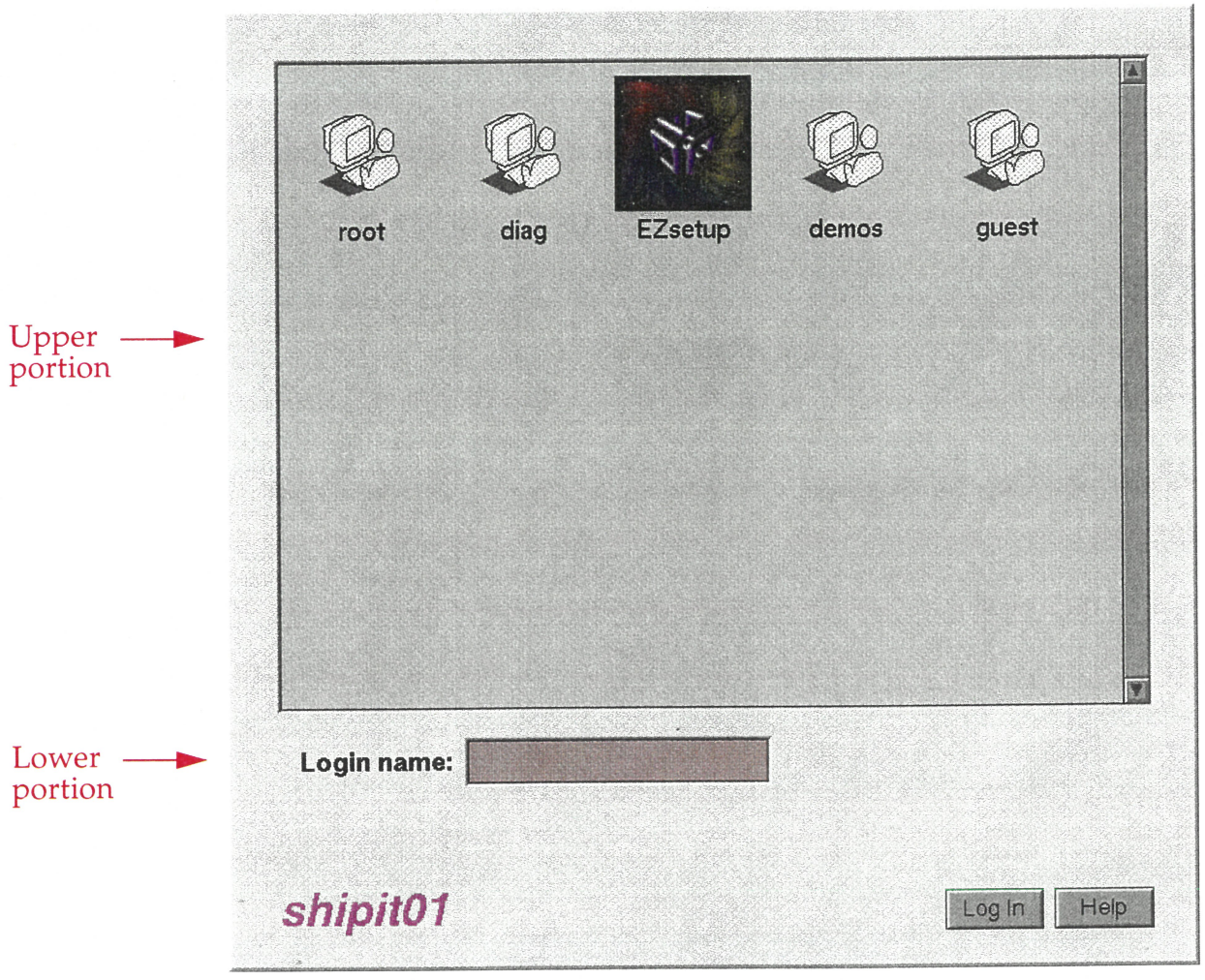
The **configuration state variables** are displayed using the *chkconfig* command.

Variable	Default	Definition
windowsystem	**on**	If **on**, allows the **init** process to start **Xsgi**.
xdm	**on**	If **on**, allows the **init** process to start **xdm**.
visuallogin	**on**	If **on**, the **Clogin** window is displayed. If **off**, a small **Welcome to IRIX** login window is displayed.
noiconlogin	**off**	If **off**, the **user icons** are displayed in the upper portion of the **Clogin** window. If **on**, a large SGI logo is displayed in the upper portion of the **Clogin** window.

Student Notes

13-48 FOIL: Clogin Window

- The **Clogin** window has an **upper** and **lower** portion:
  - The upper portion is used to display **user icons** or the **SGI logo**.
  - The lower portion contains the **Login name** box.
- The configuration state variable **noiconlogin** determines what is displayed:
  - If **noiconlogin** is **off**, the **user icons** are displayed in the upper portion of the **Clogin** window.
  - If **noiconlogin** is **on**, a large **SGI logo** is displayed in the upper portion of the **Clogin** window.



Student Notes

13-49 FOIL: X Display Manager (xdm)

/usr/bin/X11/xdm

- The **X display manager** (**xdm**) provides services similar to that provided by the **init, getty**, and ***login*** processes at an ASCII terminal: prompting for a user login name, authenticating the user, and starting a session.

  Note: If the system is experiencing trouble starting **xdm**, the following files may contain hints that identify the problem.

  - **/usr/lib/X11/xdm/xdm-errors**
  - **/var/adm/SYSLOG**

- **xdm** is a binary file and you can display its ASCII strings using the following command:

  strings /usr/bin/X11/xdm | more

- **xdm** runs the **/usr/bin/X11/X** process which performs the following functions:

  - If the configuration state variable **windowsystem** is **on**, runs the process **/usr/gfx/gfxinit**.

    This process downloads the graphics microcode.

  - When **gfxinit** completes the download, the **X** process terminates and the process **Xsgi** is started as a daemon. The same process ID (**PID**) is assigned to the **Xsgi** process that had been assigned to the **X** process.

    Xsgi is the **X Window Server**.

Student Notes

13-50 FOIL: X Display Manager (continued)

- After **gfxinit** completes, **xdm** runs the **/usr/lib/X11/xdm/Xlogin** process which performs the following functions:
  - Sets the default gamma value (**1.7**) for the graphics monitor.
  - Defines the screen background color (**light blue**).
  - List what screens (monitors) are available: usually only one monitor is present and the variable **DISPLAY** is set to **DISPLAY:0.0**.
    - The **second digit** is the pipe number.
    - Use the **env** command to print the environmentals.
  - Test the condition of the configuration state variables **visuallogin** and **noiconlogin**.

    If **visuallogin** is **on**:

    Then start **Clogin**

    If **noiconlogin** is **on**, display the **SGI logo**.

    If **noiconlogin** is **off**, display the **user icons**.

    Else the **Welcome to IRIX** login window is displayed by **xdm** using the process **/usr/lib/X11/xdm/Xresources**.
- After **Xlogin** completes, a user can login.

Student Notes

13-51 FOIL: Login Process

- At the end of the boot process, the following processes should be running:
  - If **visuallogin = on**: **xdm**, **Xsgi**, & **Clogin**.
  - If **visuallogin = off**: **xdm** and **Xsgi**.
- When a user logs in, **xdm** runs the **/usr/lib/X11/Xsession** process which performs the following functions:
  - If the argument **failsafe** is used during login, the **Xsession** process opens a **text-port window** and terminates **Xsession** without any further login configuration. For example: **root failsafe**, or **guest failsafe**.
  - Checks for the presence of the files **.xsession** or **.sgisession** in the users home directory.

    These files are used to set user environment variables and customize the users display. These files perform functions similar to the **.login** or **.profile** files.
  - Runs the process **/usr/bin/X11/userenv** which starts the user login shell.
  - Starts the **motif** window manager **/usr/bin/X11/4Dwm** as a daemon.
  - Runs the process **/usr/bin/X11/wait4wm**.

    This process insures that the **X window manager** (**Xsgi**) is started before allowing other windows to be opened.
  - Starts the **console** window as an **icon**.
  - Starts the **toolchest** as an **icon**.
- After the login process is complete, the following processes should be running: **xdm**, **4Dwm**, **toolchest**, **xwsh -console**, and **Xsgi**.

Student Notes

13-52 FOIL: Diagnostics: irsaudit

- The **irsaudit** program is used for hardware diagnosis and component fault isolation.
  - Unlike previous graphics hardware diagnostics, **irsaudit** does not use the graphics Integrated Diagnostic Environment (IDE).
  - **irsaudit** does not have a **prompting** interface.
  - **irsaudit** is driven by **command-line** options.
- The following conditions are required to run **irsaudit**:
  - The IRIX operating system must be running.
  - The **X window server should not be running**, because **irsaudit** takes direct control of the graphics hardware.
  - If the **X window server** is running, use */usr/gfx/stopgfx* to stop it.
  - The user must have **root privilege** to run **irsaudit**.
  - The diagnostic **cannot be run from the graphics console** being tested: it should be run from either a remote shell or the diagnostic terminal (the ASCII terminal).
- **irsaudit** uses **JTAG** technology to verify that chips are connected to each other and not open or shorted.
- The **JTAG** controller is in the **HIP2** ASIC.
- **irsaudit** is shipped with every system.

Student Notes

13-53 FOIL: Troubleshooting Tips

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

  1. **GE14**
  2. **DG5**

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

  1. **GE14**
  2. **DG5**

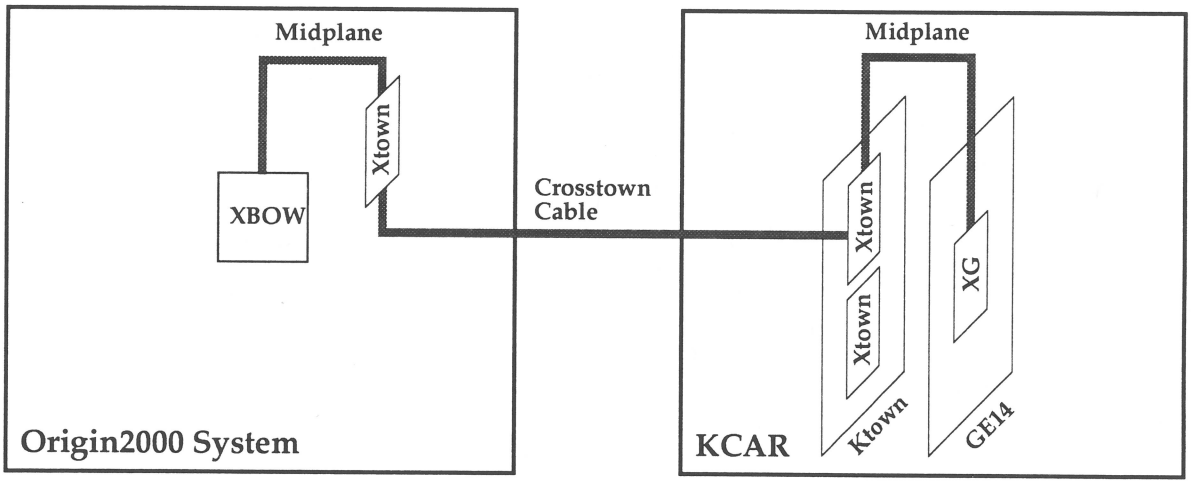
- If garbage is displayed or a window cannot be opened, the suspect board is: **GE14.**
- If a window opens and it has **vertical lines** (solid or dashed) or evenly-spaced pixel drop out (**holes**), the suspect board is: **RM.**

  For **2**-RM or **4**-RM board configurations, back out **1** or **3** boards.

- When there is no monitor display or when the display cannot sync, the suspects are:

  1. **DG5**
  2. **Video output** parameters

- When colors are incorrect, the suspect board is: **GE14.**
- **irsaudit** cannot diagnose a wire.



This complete **path** is seen as a **wire** to **irsaudit**.

The **Xtalk link LED** could be off if any of the indicated components (XBOW, midplane, Xtown, cable, Ktown, midplane, or XG) are bad.

Student Notes

13-54 FOIL: Video Output Format

- Each video output channel is controlled by a **video output format** file (**VOF**) that specifies one or more of the following:
  - The **size of the framebuffer** that is managed by the **X window server** (the resolution size).
  - The **sweep rate** of the graphics monitor (called the **swap** rate).
  - Type of output:
    - Separate **Red**, **Green**, and **Blue** components (**RGB**).
    - **Composite** component.
  - What color or colors will carry the monitor **sync** signal.
  - Set **genlock** parameters.
- The **VOF** can be written into two storage devices on the **DG** board:
  - The **EEPROM** holds the **default** format. - WRONG

    It takes effect during the power-on boot process when the graphics subsystem is initialized.
  - The VOF **RAM** holds the **current** format. - ONLY PLACE INFO IS KEPT USE INCOMBINE
    - Contains the format read from the **EEPROM** during initialization.
    - A new format can be created and written into the RAM while the graphics subsystem is running.

Student Notes

13-55 FOIL: ircombine

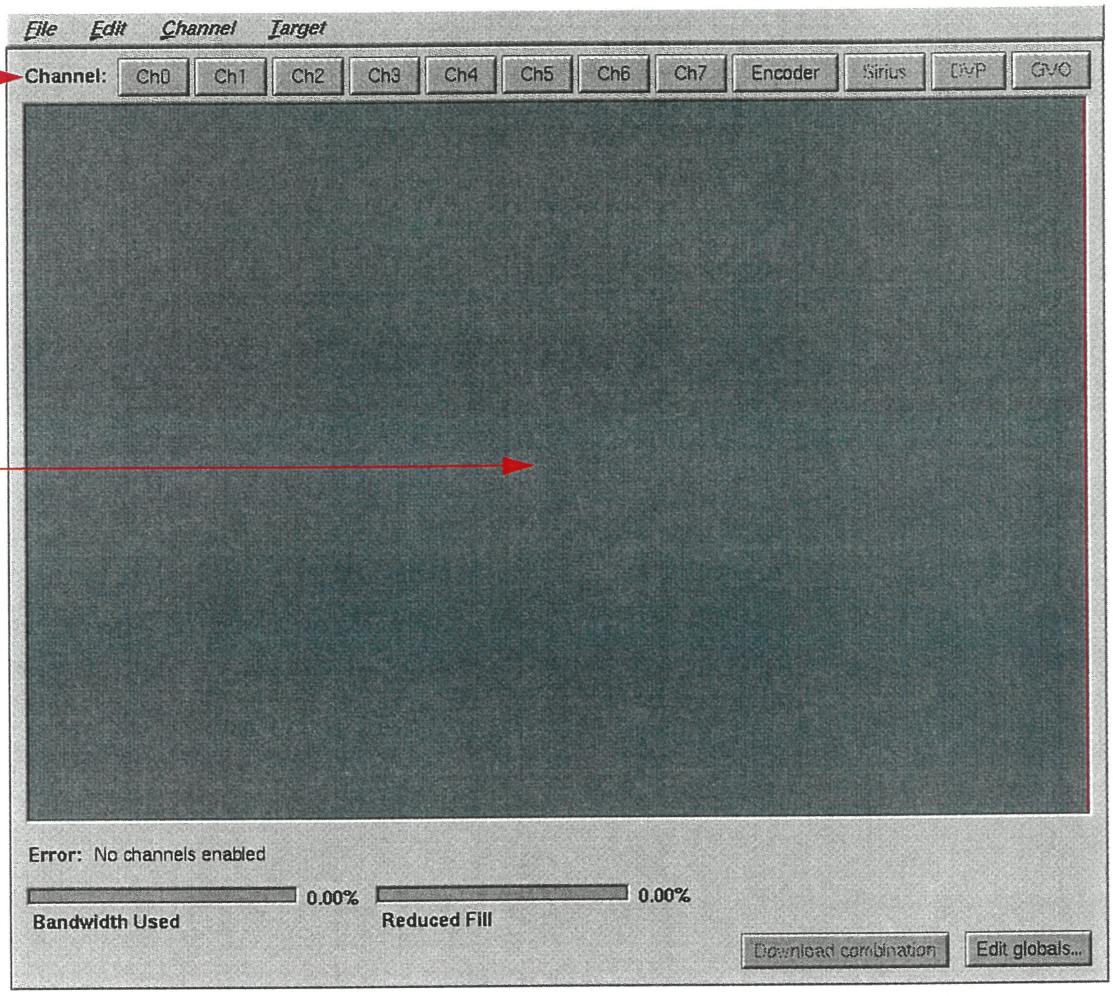
- The ***ircombine*** command is used to create, modify, query, load and save video format combinations.
- The ***ircombine*** command allows the user to specify a **combination** of video formats; one for **each channel** of the DG board.
  - The new combination can be saved in a file; this allows the changing of video formats very quickly when needed.
  - The new combination can be loaded into the VOF **RAM**, which then becomes the **current** format.
  - The new combination can be loaded into the **EEPROM**, which then becomes the **default** format.
- For graphics subsystems other than **InfiniteReality**, the current or default video format is created using the ***setmon*** command.
- The ***setmon*** command is available for **InfiniteReality** graphics subsystems, but use of the ***ircombine*** command is recommended: it is a **gui** (graphics user interface) for the ***setmon*** command.

Student Notes

13-56 FOIL: Initial ircombine window

```
# /usr/gfx/ircombine&
```

Click on a button to select a **channel** or **function**.

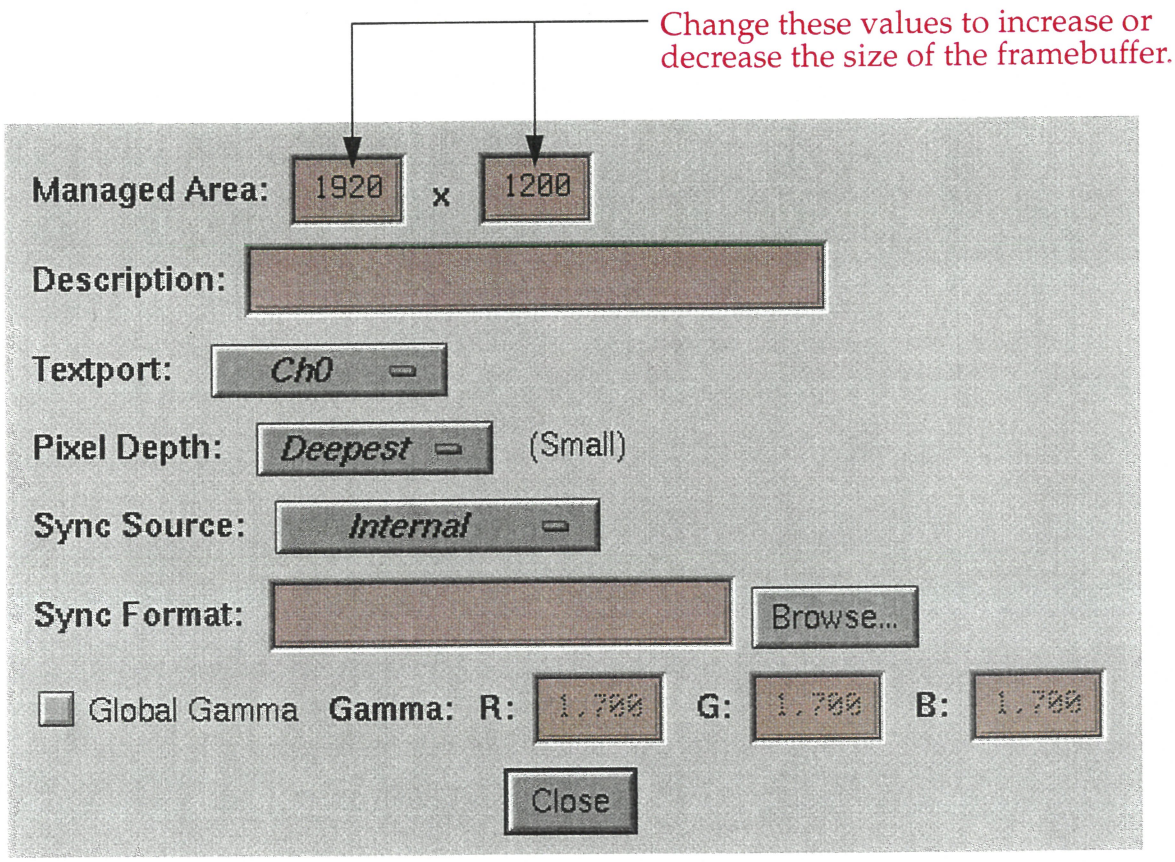


This area represents the **total size** of the **framebuffer** that is managed by the **X window server**.

To adjust the size of the managed area, click on the **Edit globals** button (see the next drawing for the Edit globals window).

Student Notes

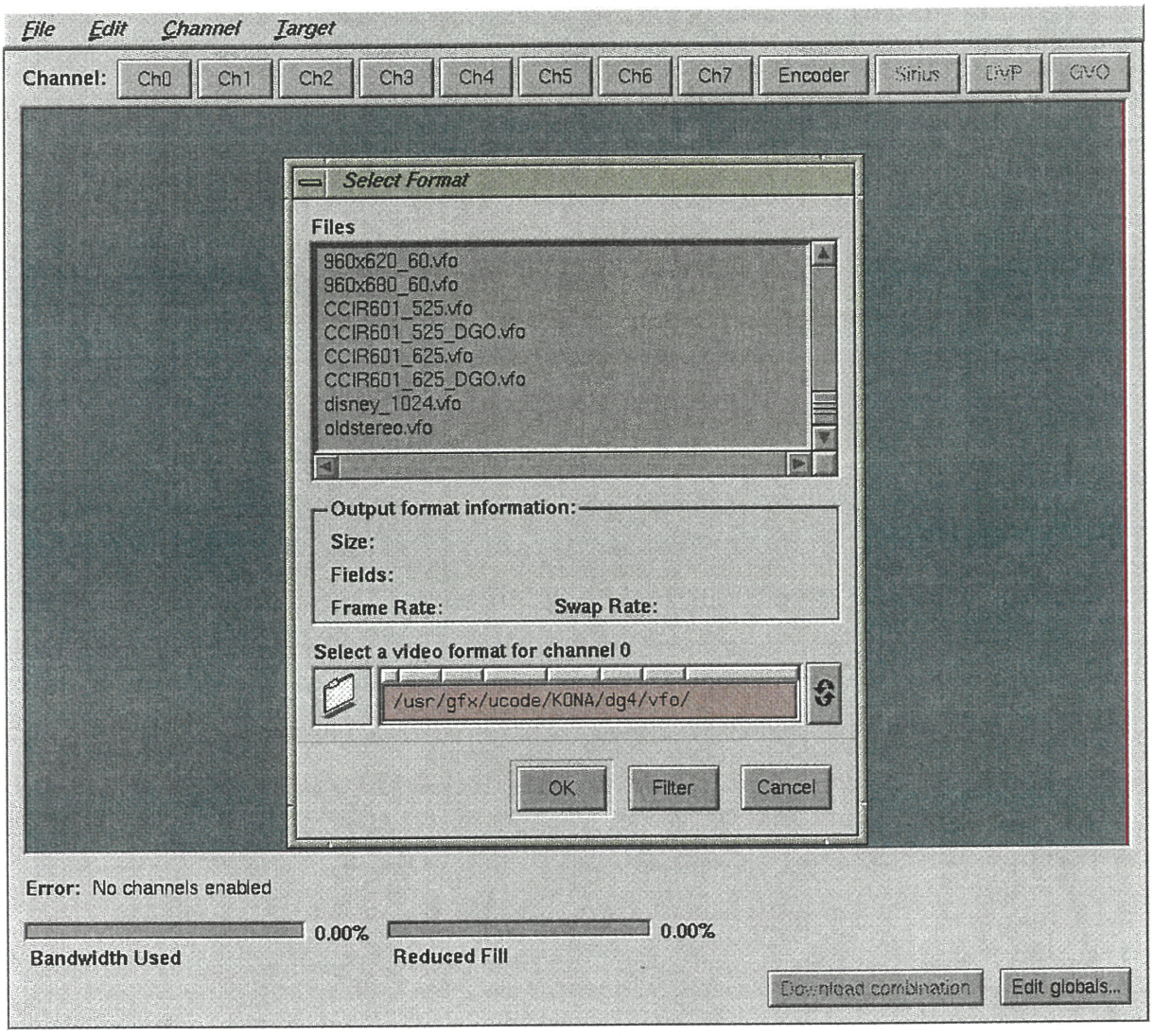
13-57 FOIL: Edit globals window



Student Notes

13-58 FOIL: Select channel 0

- To create a new format, no channels can currently be selected.
- Selected channels are identified when the letters of the **Channel** select buttons are **blue**.
- To select **channel 0**, use the **right** mouse button and **click** on the **Ch0** button.

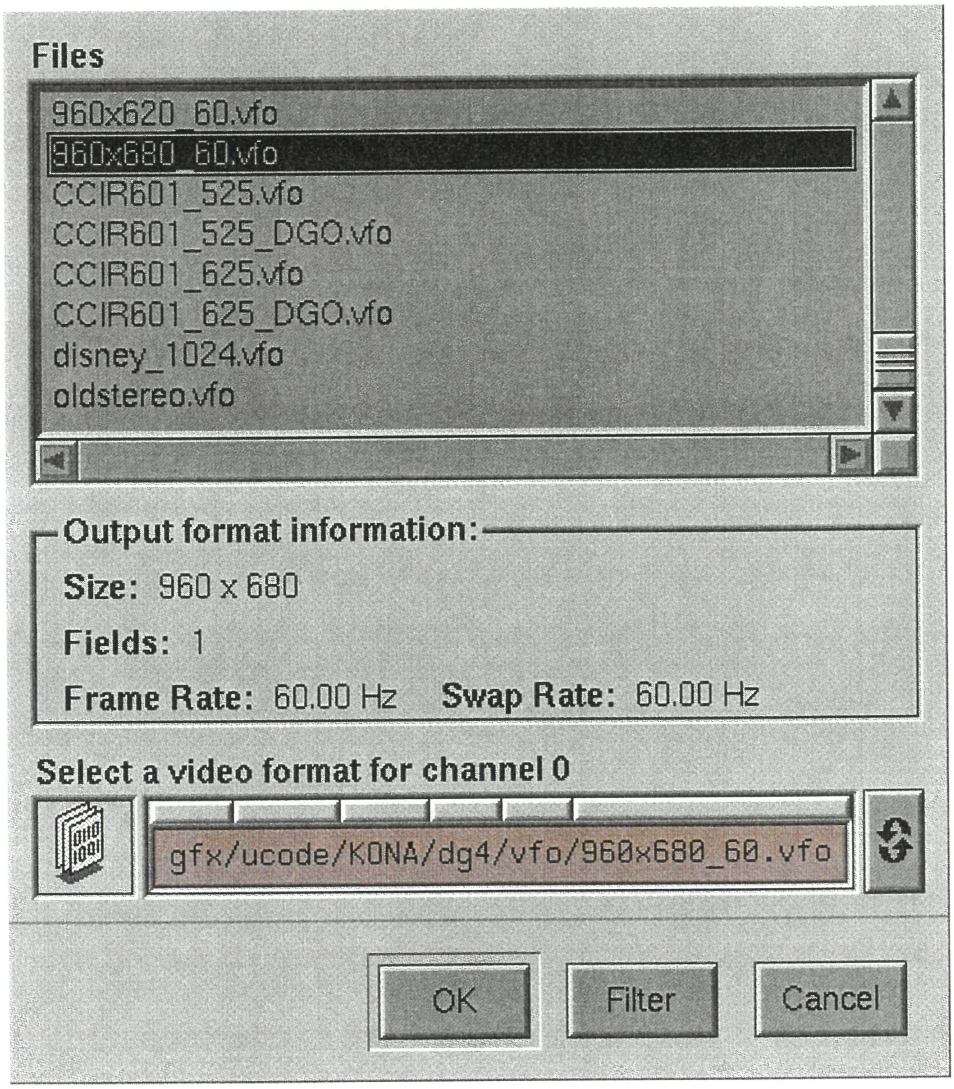


- Select a pre-defined video format from the supplied list (the next drawing illustrates the selection of the format **960x680 pixel resolution** at **60 Hz**).

Student Notes

13-59 FOIL: Select a video format

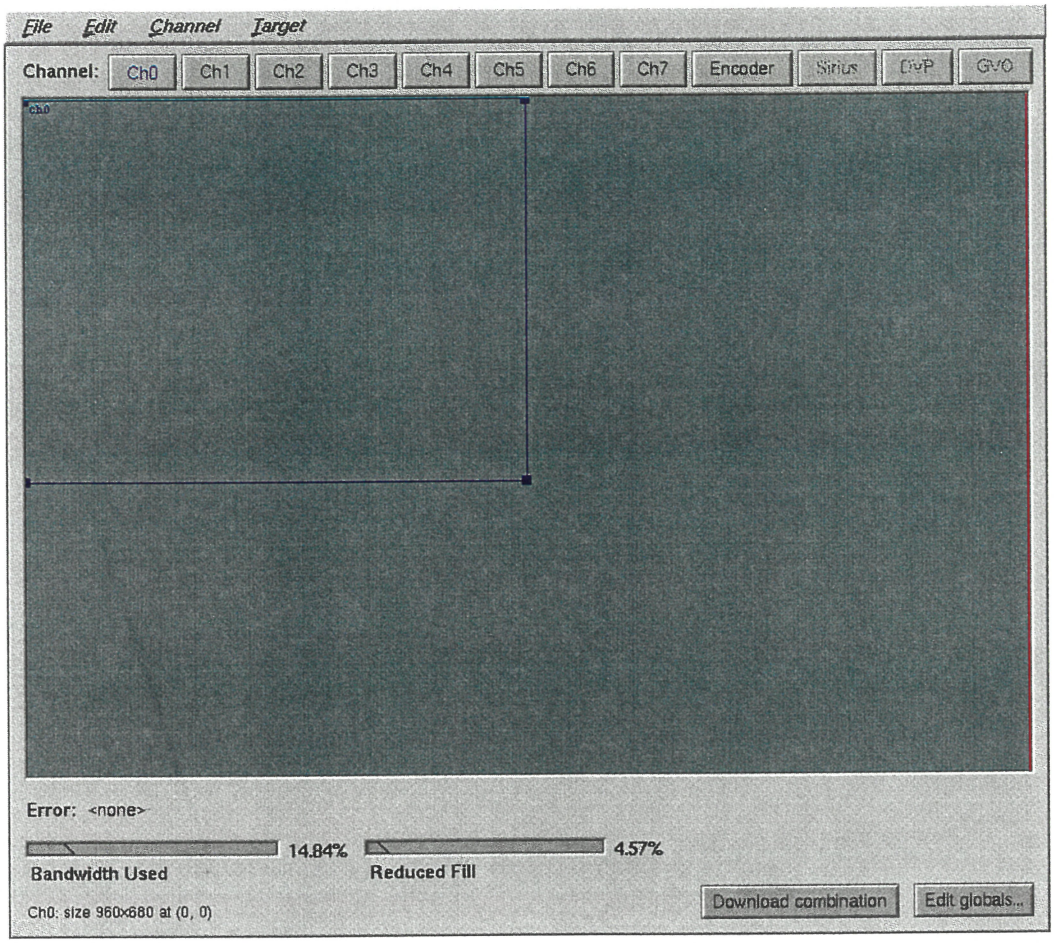
Select a video fromat from a predetermined list of valid formats and **click** on the **OK** button.



Student Notes

13-60 FOIL: A selected channel

- The letters of the **Ch0** button change to **blue**, and a **blue** outlined box for **channel 0** is displayed in the managed area of the framebuffer.
- The **blue** outlined box represents the total amount of the available space of the framebuffer used by **channel 0** due to the selected format.
- Click on **Download combination** and the new combination becomes the **current** format (it loads the format into the **DG RAM**).
- Select the **Save As** function from the **File** pull-down menu to save the combination for future use.
- Select the **Save to EEPROM** function from the **File** pull-down menu to download the new combination to the **DG EEPROM** (the format will become the **default** format anytime the graphics subsystem is **initialized**).
- Before saving or downloading the format, select the **Edit attributes** function from the **Channel** pull-down menu to change any of the channel attributes (see the next illustration).

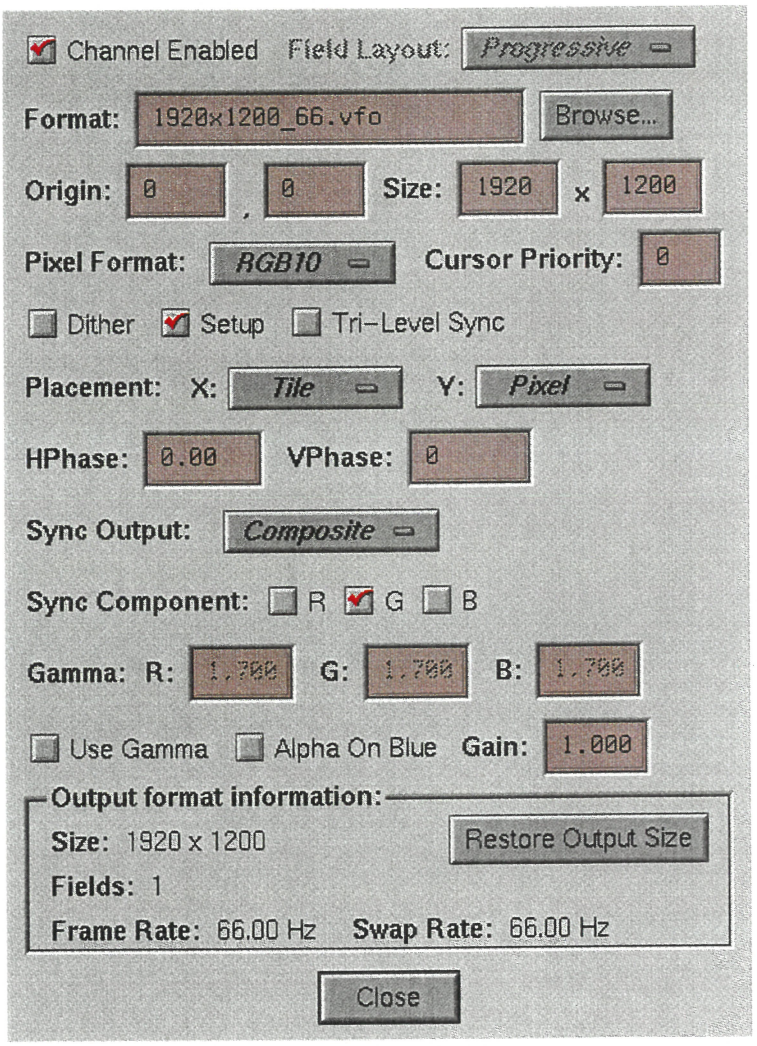


Student Notes

13-61 FOIL: Edit attributes window

- These are the **attribute** (parameter) settings for the selected channel.
- See **ircombine** man page for descriptions.

Window parameter name	Corresponding man page parameter name
Channel Enabled	enable=
Field Layout	scan=
Format	formate=
Origin	sourceloc=
Size	sourcesize=
Pixel Format	pixelformat=
Cursor Priority	cursorpriority=
Dither	dither=
Setup	pedestal=
Tri-level	synctrilevel=
Placement	panstep=
HPhase	hphase=
Vphase	vphase=
Sync Output	syncport=
Sync Component	sync=
Gamma	gamma=
Use Gamma	globalgamma=
Alpha On Blue	alpha=
Gain	gain=



Student Notes

13-62 FOIL: Lab Project 6: irsaudit and ircombine

- Two or three students are assigned a classroom system.
- Each team begins the lab project at the Base IO console (the ASCII terminal).
- Two color monitors are required to complete **Section 2** of this lab project. One monitor should be connected to **channel 0** of the **DG5** board, and the other monitor should be connected to **channel 1**.
- Time allotted to complete project: **2 hours**.
- Time allotted to go over the review questions: **30 minutes**.
- After completion of the lab project, the student should be able to:
  - Verify that the microcode can see and communicate with the graphics subsystem using the PROM monitor *hinv* command.
  - Switch control of the console between the Base IO board and the graphics subsystem using the PROM monitor *setenv* command.
  - Manually start and stop the graphics ***X window server*** (Xsgi) while the IRIX operating system is running.
  - Test the complete graphics subsystem using the InfiniteReality system diagnostics program (***irsaudit***).
  - Test an individual graphics board using the InfiniteReality system diagnostics program (***irsaudit***).
  - With the IRIX operating system running and using the *gfxinfo* command, display graphics subsystems configuration information.
  - Force console control to the Base IO console keyboard from the graphics keyboard when the PROM monitor environmental ***console*** is set to *g* and the graphics keyboard is not responding.
  - Cause a restart of the graphics ***X window server*** (Xsgi) using the special graphics keyboard **4-keystroke** procedure.
  - Log in at the graphics keyboard without starting the X window system, but also gaining control of the system using a ***text port window.***
  - Using ***ircombine***, create a new ***video format combination file*** that will enable video output channels **0** and **1**. The video resolution of each channel will be **960x680 pixels** at **60 Hz**.

Instructor Note: Before students move to the systems, configure each rack system per the ***Lab Project Set-up Instructions*** for ***Lab Project 6*** in the ***Instructor Manual***.

0-1 FOIL: Onyx2 Multi-Pipe Graphics Configurations

This section provides an overview of multi-pipe graphics configurations for the Onyx2 system. The topics presented include:

- Definition of a multi-pipe system
- Multi-pipe vs. multi-channel configurations
- X window system concepts
- X display configurations
- Configuring Xtown boards on Onyx2 systems
- Basic configuration guidelines and limitations
- Xtown/IO6G board placement
- Pipe numbering
- IO6G keyboard and mouse port numbering
- KCAR module pipe placement
- Configuring multiple pipes and X Servers
- Configuring multiple X displays
- Configuring multiple X servers
- Screen positioning examples

0-2 FOIL: Multi-pipe Definitions

Graphics Pipeline

The term ***graphics pipeline*** (or ***pipe***) refers to the dedicated graphics rendering hardware that supports a single graphics seat and 1 to 8 monitors. Up to 8 pipes are currently supported in an Onyx2 InfiniteReality configuration. Each pipe must be supported by an Origin2000 node board. The following components support a minimally configured pipe in an Onyx2 rack system:

- Origin2000 node board
- IO6G (BaseIO with MIO option)
- Xtown board
- Xtown cable
- Ktown board
- GE-14 board (geometry engine)
- RM7 board (raster/texture memory)
- DG5-2 board (display generator)

Graphics Seat

A minimally configured graphics ***seat*** or ***head*** comprises the following components:

- 1 graphics pipe
- 1 keyboard/mouse
- 1 monitor
- 1 X server

Channel

Each ***channel*** drives a single monitor from the output of a display generator (DG) board. A single DG board can drive a maximum of 2 (DG5-2) or 8 (DG5-8) channels, depending on which board you install.

0-3 FOIL: Multi-Pipe Configurations

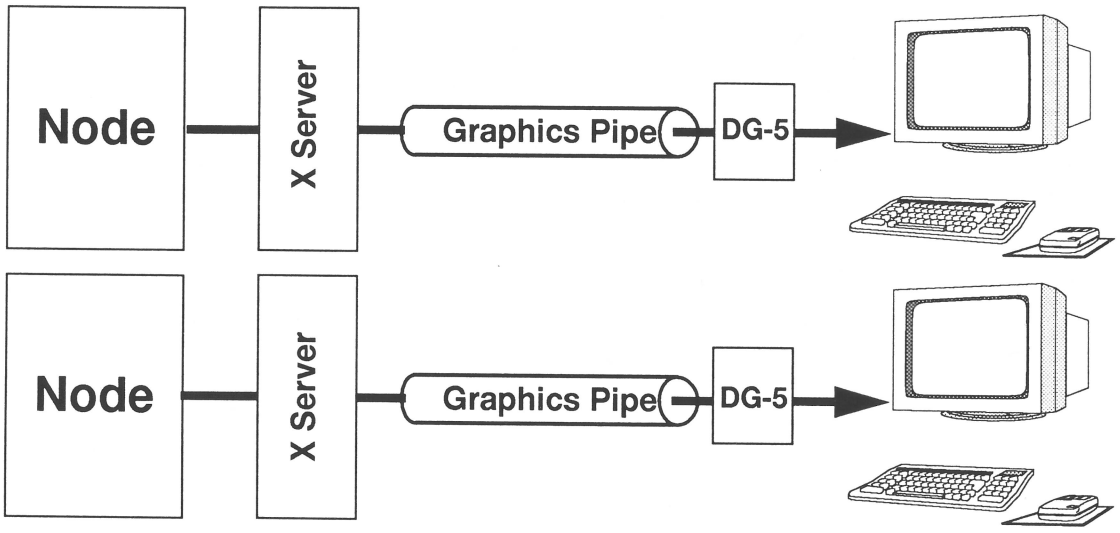
Multi-pipe configurations provide higher performance, more available screen area than single-pipe systems, and allow multiple users to access a single system dataset. Some specific applications of multi-pipe configurations include:

- Driving PowerWall, Reality Center, dome projection, or other multi-screen configurations
- Volume rendering of extremely large datasets that exceed the raster memory capacity of a single pipe
- Dividing the geometry calculation workload for a single image across multiple pipes
- Supporting multiple users sharing a common dataset. This arrangement is common in the Oil, Gas, CAD, and Nuclear market segments
- "Workstations by Day, Supercomputer by Night." This configuration allows a multi-pipe system to function as a cluster of individual workstations or as a single, computational unit, depending on the customer's requirements

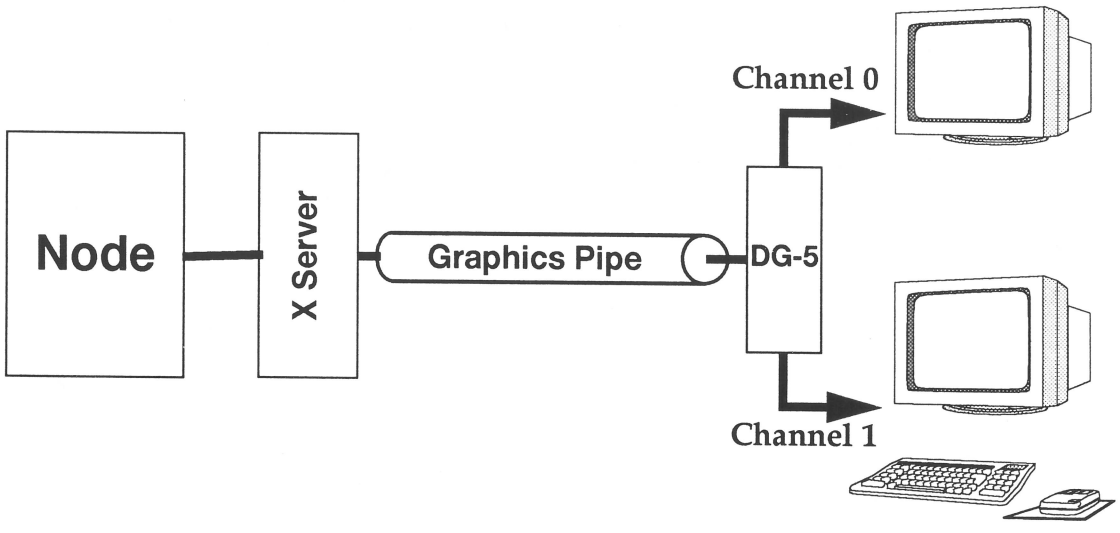
0-4 FOIL: Multi-pipe vs. Multi-channel Configurations

The term ***multi-pipe*** is not interchangeable with ***multi-channel***. Multi-pipe configurations have a dedicated graphics pipe and frame buffer for each graphics seat in the system. Multi-channel configurations have multiple monitors configured on a single graphics pipe. Each monitor is user-configured to display a different part of the same frame buffer and is attached to the same DG-5 board. Refer to the illustrations below:

Multi-Pipe Configuration



Multi-Channel Configuration



0-5 FOIL: X Window System Concepts

SGI's implementation of the X Window System server (**Xsgi**) places constraints on the configuration of screens, displays, and windows. It is necessary for you to understand the definition of these terms and their impact on Onyx2 configurations.

- **X Screen**

  Xsgi defines an X screen as a single pipe and all devices attached to that pipe. For example: a single screen could comprise a keyboard/mouse and from 1 to 8 monitors. Each screen displays a single frame buffer's contents.

- **X Display**

  An X display is a collection of one or more screens together with a keyboard and mouse. Typically, a single X server manages one X display and one X screen. However, in dual-pipe display configurations, one keyboard and mouse set control two X screens and a single X display.

- **X Server**

  The X Window System display server (or **X)** is a binary that drives the server process for each display configured on your system. The **xdm** program normally starts the X server utility.

- **X Window**

  An X window is a rectangular subsection of a screen. Within its borders, all X rendering is clipped and X events are registered. An X window cannot be moved between screens in multi-pipe systems.

0-6 FOIL: DISPLAY Variable

The DISPLAY variable is a shell environment variable that directs the output of a process to a particular display and screen. You can set the DISPLAY variable several different ways:

- **C shell**: `setenv DISPLAY` *`hostname:display#.screen#`*

  Example: `setenv DISPLAY bushido:0.1`

- **Korn shell**: `export DISPLAY=`*`hostname:display#.screen#`*

  Example: `export DISPLAY= bushido:0.1`

- **-display** option of an X program (passed by the `DisplayManager.servers` file. See the `X(1)` man page for further details)

You can use the `xlistscrns` command to obtain a complete list of all the screens configured for a particular display:

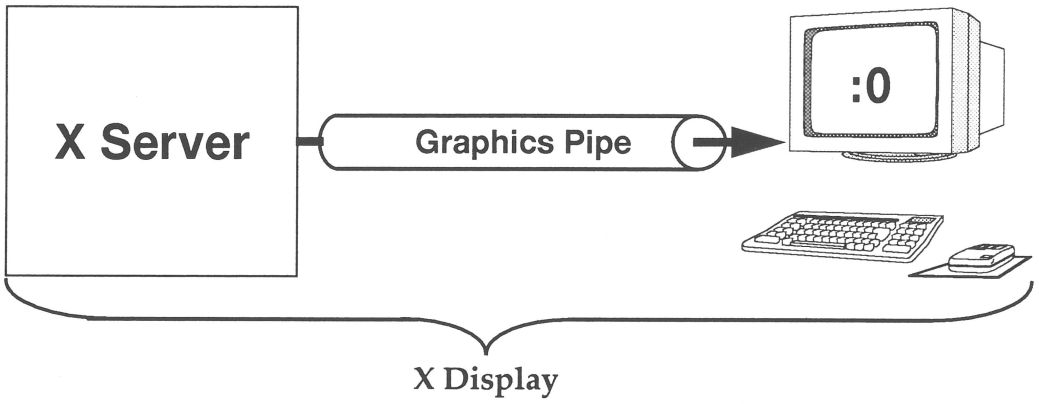
Example:

```
# /usr/bin/X11/xlistscrns
# :0.0 :0.1 :0.2
```

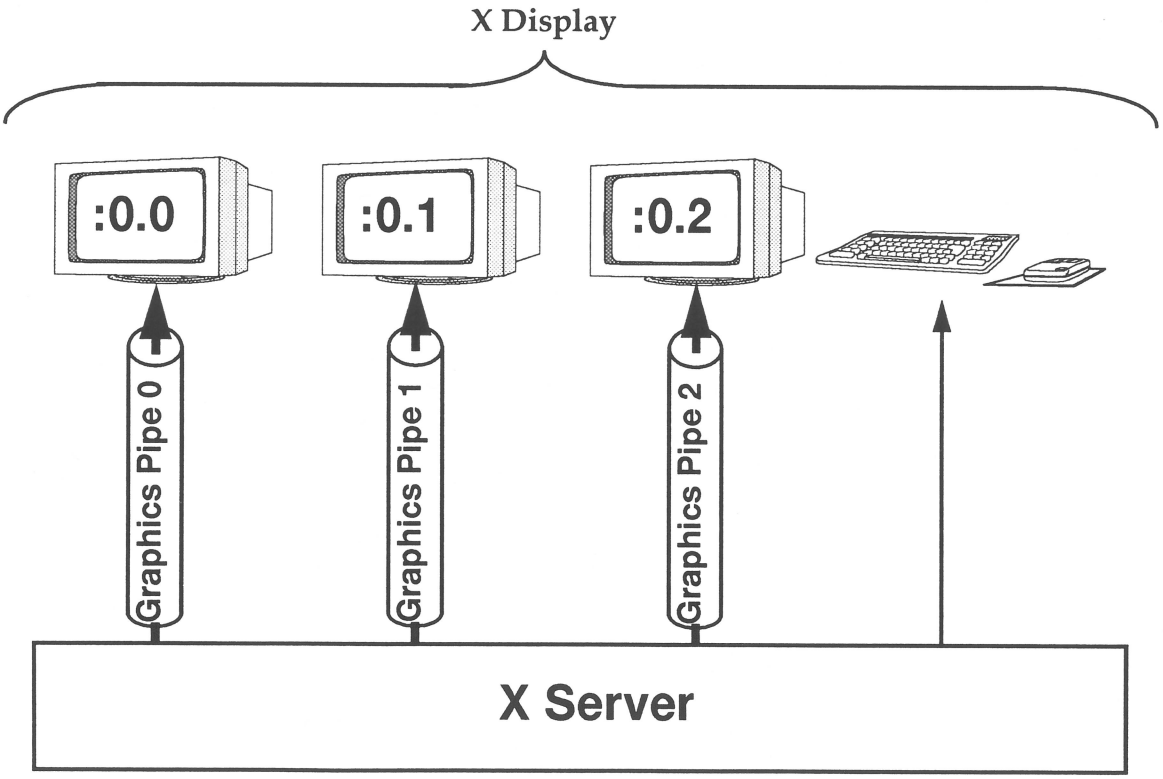
0-7 FOIL: X Display Configurations

The illustrations below provide an example of single and multi-pipe X display configurations:

Basic X Display Configuration



Multi-Screen X Display Configuration



0-8 FOIL: Configuring Xtown Boards on Onyx2 Systems

The following section describes some basic rules that you can follow to configure a multi-pipe Onyx2 rackmount system.

Xtown/IO6G Placement

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

As shown in the table below, 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, Each Origin2000 server module (8P/12) must contain an IO6G in XIO slot 1 to support the keyboard and mouse connections required for each graphics seat.

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

0-9 FOIL: Configuring Xtown Boards on Onyx2 Systems (Cont.)

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

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

0-10 FOIL: Basic Configuration Guidelines and Limitations

A single 8P/12 module with 4 nodes can support a maximum configuration of two graphics seats with 2 pipes configured for each seat. An Onyx2 graphics insert module (KCAR) can support a maximum of 2 pipes. Each graphics seat in an Onyx2 rack system requires the following minimum hardware configuration:

- 1 keyboard
- 1 mouse
- 1 monitor
- 1 8P/12 module that contains:
  - 1 IP27 node board
  - 1 IO6G
  - 1 Xtown board
  - 1 Crosstown cable
- 1 KCAR module containing:
  - 1 RM board
  - 1 GE board
  - 1 DG5 board
  - 1 Ktown board

Configuration Limitations

You must observe the following limitations when configuring Onyx2 systems:

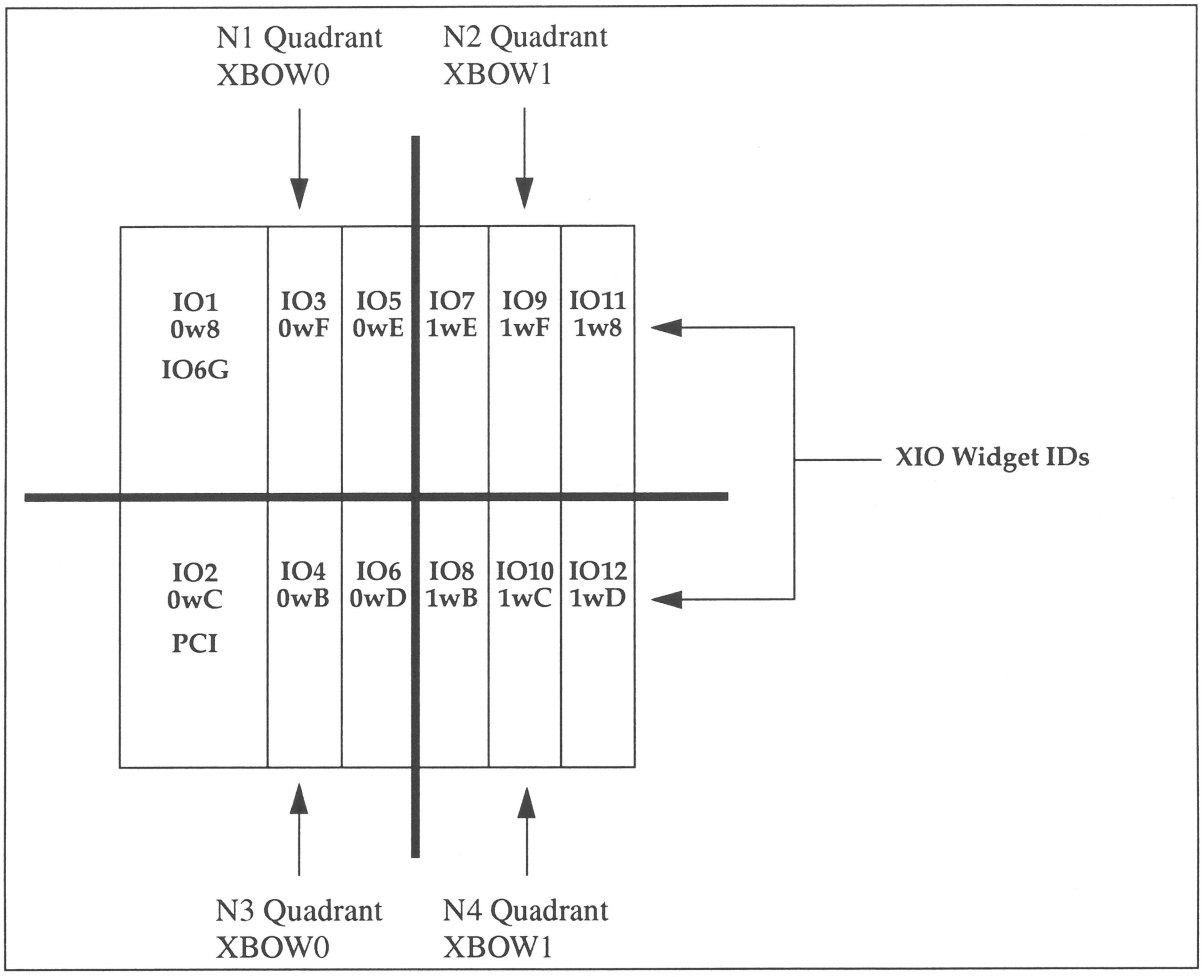
- One node board supports a single pipe. You cannot configure multiple pipes on a single node
- One DG-5 board drives monitors on a single pipe
- Two keyboards cannot be used to control a single X display
- One IO6G supports 2 keyboard/mouse sets
- Multiple output channels on a single pipe cannot be configured as separate X screens

0-11 FOIL: Pipe Numbering

The IRIX kernel numbers the pipes using the following global ordering:

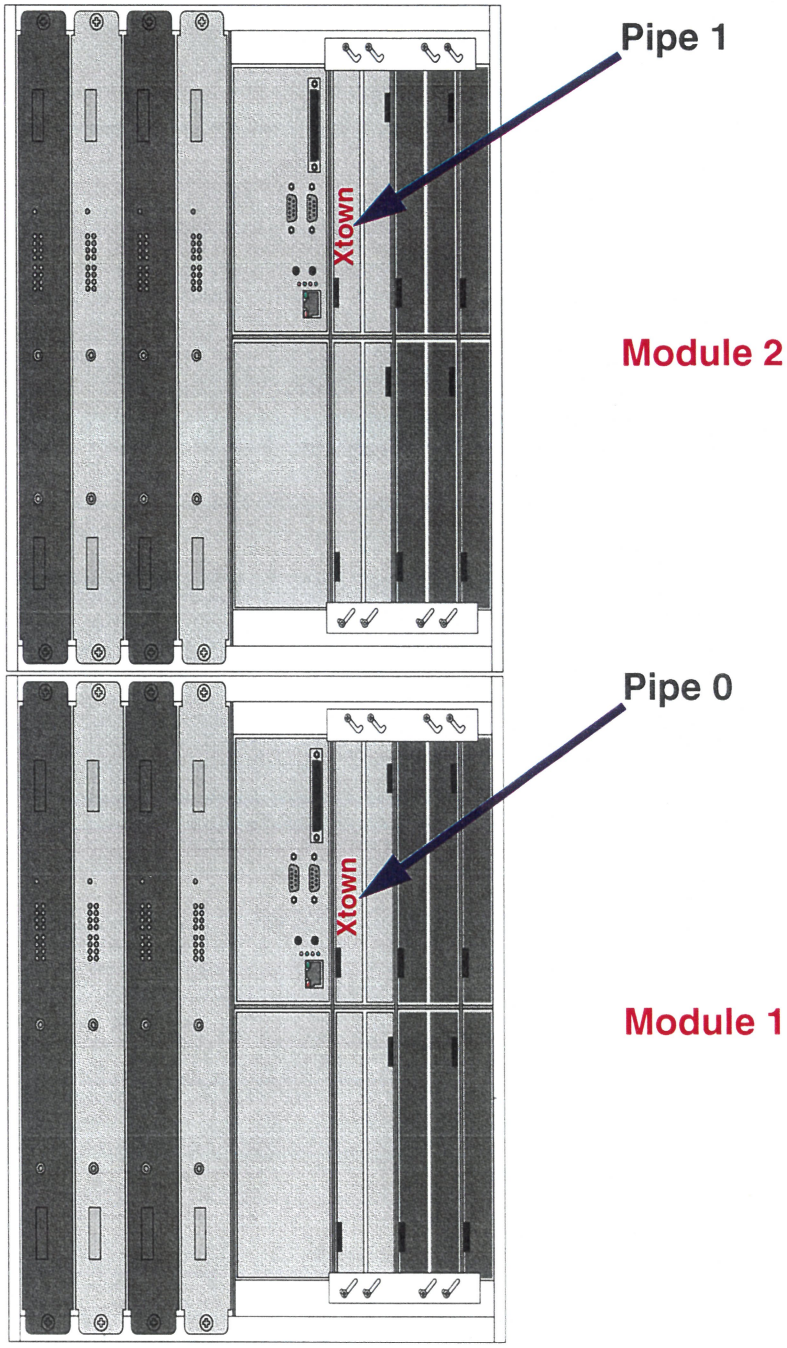
- module ID number
- XBOW number
- XIO widget ID number

The figure below shows the XIO widget IDs for each XIO slot:



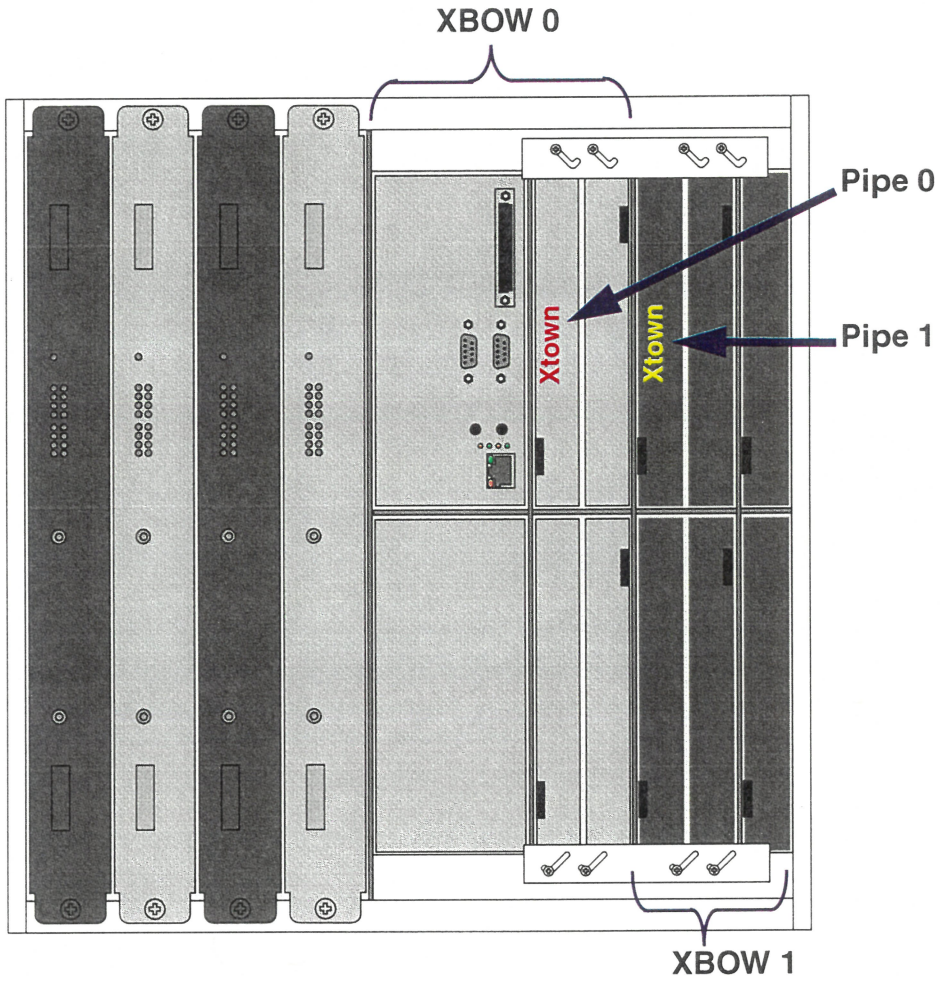
0-12 FOIL: Pipe Numbering Examples

The following examples demonstrate the pipe numbering algorithm. One Xtown board is located in XIO slot 3 of each 8P/12 module. In this case, IRIX assigns pipe 0 to the Xtown board in the module with the lowest number.



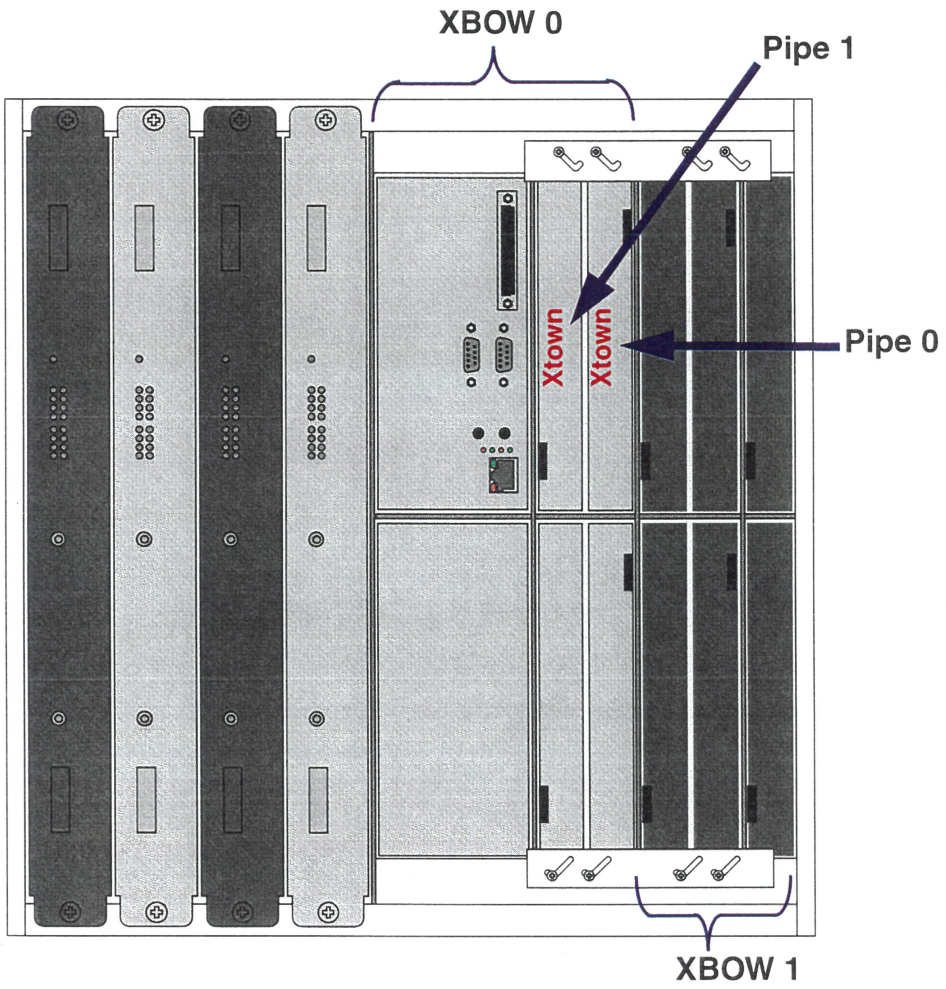
0-13 FOIL: Pipe Numbering Examples (Cont.)

If an 8P/12 module contains more than one Xtown board, the pipe numbers are assigned by lowest XBOW and XIO widget number. In the example below, XBOW 0 controls one Xtown board and XBOW 1 controls a second Xtown. Therefore, IRIX assigns pipe 0 to XBOW 0's board:



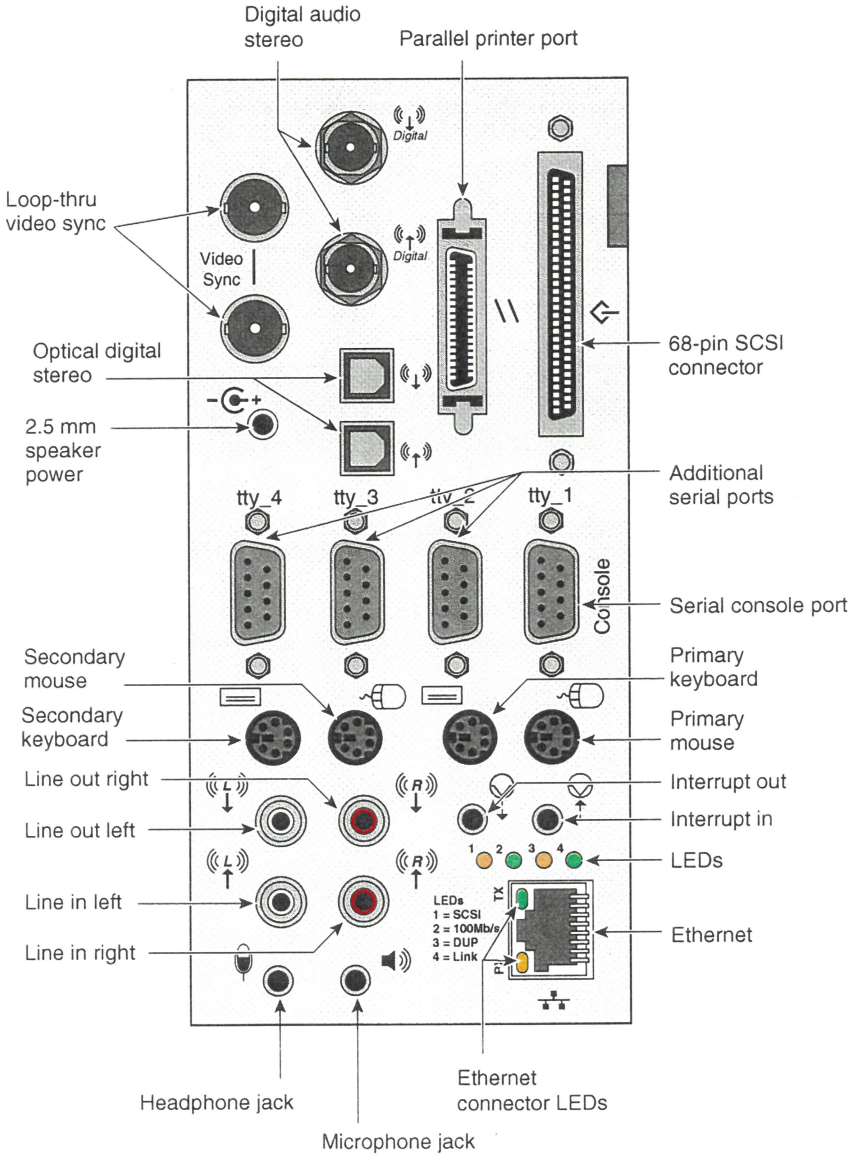
0-14 FOIL: Pipe Numbering Examples (Cont.)

If a single XBOW controls more than one Xtown board, IRIX assigns pipe numbers by widget IDs. In the example below, one Xtown board resides in io3 and the other in io5. XBOW 0 controls both boards. The Xtown board in io5 receives the lowest pipe number because its widget ID (0wE) is lower than io3's (0wF):



0-15 FOIL: IO6G Keyboard and Mouse Port Numbering

Each IO6G provides 2 sets of keyboard and mouse ports labeled primary and secondary, as shown in the following diagram. The port that you connect a keyboard and mouse set to determines which **hwgraph** file that set is linked to.



0-16 FOIL: IO6G Keyboard and Mouse Port Numbering (Cont.)

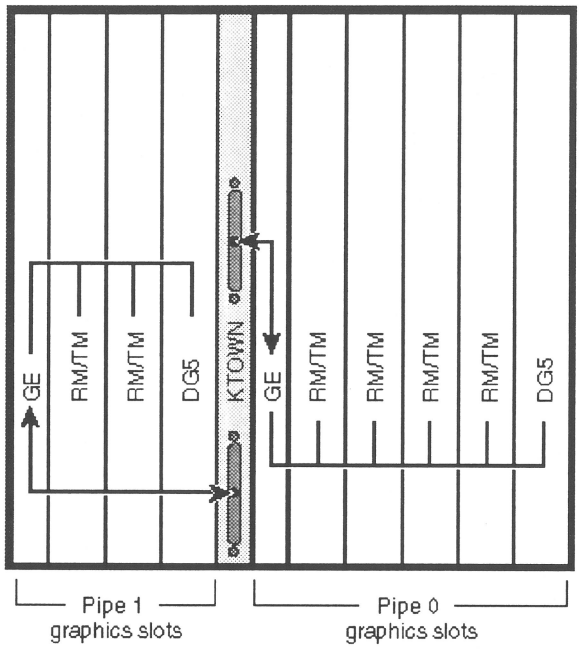
Starting with the IO6G located in the lowest numbered module, **/hw/input** is assigned to the primary (right) port and **/hw/input1** is assigned to the secondary (left) port. The IRIX operating system links these files to character special files in the **/dev** directory. You can use these files to assign a specific keyboard/mouse set to a particular X display.

If your system contains a second IO6G, the system reverses the port order and assigns **/hw/input2** to the ***secondary*** (left) port and **/hw/input3** to the ***primary*** (right)port. The system will assign **/hw/input** numbers to subsequent IO6Gs in the same manner (refer to the table below). The Onyx2 system supports a maximum of 8 keyboard/mouse sets.

IO6G	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

0-17 FOIL: KCAR module Pipe Placement

You should connect the lowest numbered pipe in each 8P/12 module to the 4RM (right) side of the KCAR module as shown in the figure below. Therefore, you must connect the Crosstown cable from pipe 0 to the upper connector of the Ktown board. If the 8P/12 is configured with more than 2 pipes, then you should connect pipe 3 to the 4RM side of a second KCAR module and so on.



0-18 FOIL: Gfxinfo

Once you have attached Crosstown cables to the Ktown board connectors, you can use the **gfxinfo** command to determine which pipe is connected to each side of the KCAR module. If the pipe with the lowest number is not connected to the 4-RM side of the KCAR, power down the system and swap the Crosstown cables where they connect to the Ktown board. Power up the system again and use **gfxinfo** to ensure that the pipes are correctly configured. The sample output below shows that pipe 0 is attached to the 4-RM side of the KCAR. You can also use **gfxinfo -vv** to obtain more detailed graphics information.

Note: The ouput of **gfxinfo** indentifies a graphics pipe as a ***board***.

Example:

```
# /usr/gfx/gfxinfo
Graphics board 0 is "KONAL" graphics.          <-- Pipe 0
    Managed (":0.0") 1280x1024
    Display has 2 channels
    4 GEs (of 4), occmask = 0x0f
    4MB external BEF ram, 32bit path
    4 RM7 boards (of 4) 1/1/0/0                <-- RM Boards
    Texture Memory: 16MB/16MB/16MB/16MB
    Medium pixel depth
    32K cmap
    Channel 0:
     Origin = (0,0)
     Video Output: 1280 pixels, 1024 lines, 60.00Hz (1280x1024_60.vfo)
    Channel 1:
     Channel's Source Size: 1280 pixels, 1024 lines; Origin = (0,0)
     Video Output: 640 pixels, 486 lines, 29.97Hz (640x486_30i.vfo)
Graphics board 1 is "KONAL" graphics.          <-- Pipe 1
    Managed (":0.1") 1280x1024
    Display has 2 channels
    4 GEs (of 4), occmask = 0x0f
    4MB external BEF ram, 32bit path
    2 RM7 boards (of 2) 1/1/0/0                <-- RM Boards
    Texture Memory: 16MB/16MB/-/-
    Medium pixel depth
    32K cmap
    Channel 0:
     Origin = (0,0)
     Video Output: 1280 pixels, 1024 lines, 60.00Hz (1280x1024_60.vfo)
    Channel 1:
     Channel's Source Size: 1280 pixels, 1024 lines; Origin = (0,0)
     Video Output: 640 pixels, 486 lines, 29.97Hz (640x486_30i.vfo)
```

0-19 FOIL: Configuring Multiple Pipes and X Servers

All rackmount Onyx2 systems ship from the factory with a single X Server configured to use all of the available pipes. You must reconfigure the system during install to match the customer's required configuration.

All systems with graphics capabilities come preconfigured with 4 `shmiq` drivers, 1 `input` directory containing character special files that point to a keyboard/mouse set, and 3 `qcntl` nodes.

Complete the following steps on all multi-pipe systems. Typically, you should configure *n*+1 `shmiq`s, where *n* is the number of pipes in your system. You should configure one `qcntl` files for each pipe that your system supports.

1. Edit the `/var/sysgen/master.d/shmiq` file and change the number of shared memory input queue (`shmiq`) drivers defined in the kernel:

   Change

   ```
   #define NSHMIQS 4
   ```
   to
   ```
   #define NSHMIQS 9
   ```

   Each `shmiq` driver handles input events from user hardware devices through a duart port; in this case, a keyboard and mouse.

2. In the `/dev` directory enter the following command lines at the prompt to create additional `qcntl` nodes:

   ```
   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
   ```

   The `qcntl` nodes are associated with the `shmiq` driver. The `qcntl` device allows the X Server to process character input from the `shmiq` driver.

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

3. Enter the command, `autoconfig -fv` to create a new kernel, then reboot the system.
4. To create additional `/dev/input` directories, plug keyboard/mouse sets into the appropriate IO6G ports and reset the system from the PROM monitor prompt.

0-20 FOIL: Configuring Multiple X Displays

The X display manager (**xdm**) manages a collection of X display servers. Xdm provides various services to the user such as: prompting for a login name and password, authenticating the user, and running a **session**. The session is similar to the user's login shell process that is created on a traditional character-based terminal. Terminating the session process causes the users session to exit.

The `xdm-config` file allows you to modify the behavior of specific displays. When an action is directed towards a particular display, the display number is inserted into the X resource name as shown below:

```
DisplayManager._display#.startup: /var/X11/xdm/Xstartup
```

Add additional lines to the `/usr/lib/X11/xdm-config` file to configure more X window display servers. If you add the additional lines below to your default **Xstartup** file, a total of 8 X displays will be created (0-7):

```
DisplayManager._4.startup: /var/X11/xdm/Xstartup
DisplayManager._5.startup: /var/X11/xdm/Xstartup
DisplayManager._6.startup: /var/X11/xdm/Xstartup
DisplayManager._7.startup: /var/X11/xdm/Xstartup

DisplayManager._4.session: /var/X11/xdm/Xsession
DisplayManager._5.session: /var/X11/xdm/Xsession
DisplayManager._6.session: /var/X11/xdm/Xsession
DisplayManager._7.session: /var/X11/xdm/Xsession

DisplayManager._4.authfile: /var/X11/xdm/xdm-auth-4
DisplayManager._5.authfile: /var/X11/xdm/xdm-auth-5
DisplayManager._6.authfile: /var/X11/xdm/xdm-auth-6
DisplayManager._7.authfile: /var/X11/xdm/xdm-auth-6

DisplayManager._4.loginProgram: /var/X11/xdm/Xlogin
DisplayManager._5.loginProgram: /var/X11/xdm/Xlogin
DisplayManager._6.loginProgram: /var/X11/xdm/Xlogin
DisplayManager._7.loginProgram: /var/X11/xdm/Xlogin

DisplayManager._4.terminateServer: False
DisplayManager._5.terminateServer: False
DisplayManager._6.terminateServer: False
DisplayManager._7.terminateServer: False
```

0-21 FOIL: Configuring Multiple X Servers

You must configure an X display server for each graphics seat that your system supports; modify the `/usr/lib/X11/xdm/Xservers` file to accomplish this. Each command line in the file configures a different display server. Typically, each display server corresponds to a single keyboard/mouse set and one or more X screens. The example below shows a system configured for 3 graphics seats. The third seat contains 2 pipes (pipe 2 and 3). If you configure the **Xservers** file for a single X display server, all available pipes will be assigned to that server by default.

Note: When configuring X servers, you must not exceed the available number of pipes.

You must enter each line in the following format where:

- *display#* = X display number
- *pipe#* = screen (pipe) number
- *input#* = directory containing character special files that point to the specific keyboard and mouse set that each display uses

```
:display# secure /usr/bin/X11/X :display# -boards pipe# -devdir \
/dev/input# -bs -nobitscale -c -pseudomap 4sight -solidroot sgilightblue \
-cursorFG red -cursorBG white
```

```
:0 secure /usr/bin/X11/X :0 -boards 0 -devdir /dev/input -bs \
-nobitscale -c -pseudomap 4sight -solidroot sgilightblue \
-cursorFG red -cursorBG white
:1 secure /usr/bin/X11/X :1 -boards 1 -devdir /dev/input1 -bs \
-nobitscale -c -pseudomap 4sight -solidroot sgilightblue \
-cursorFG red -cursorBG white
:2 secure /usr/bin/X11/X :2 -boards 2 -devdir /dev/input2 -bs \
-nobitscale -c -pseudomap 4sight -solidroot sgilightblue \
-cursorFG red -cursorBG white
:2 secure /usr/bin/X11/X :2 -boards 3 -devdir /dev/input2 -bs \
-nobitscale -c -pseudomap 4sight -solidroot sgilightblue \
-cursorFG red -cursorBG white
```

0-22 FOIL: Configuring Multiple X Servers (Cont.)

If your system requires additional displays, add another line to the `Xservers` file and increment *`display#`* by one. Remember that the maximum number of pipes currently supported in an Onyx2 configuration is 8.

Enter the following commands to restart the graphics subsystem:

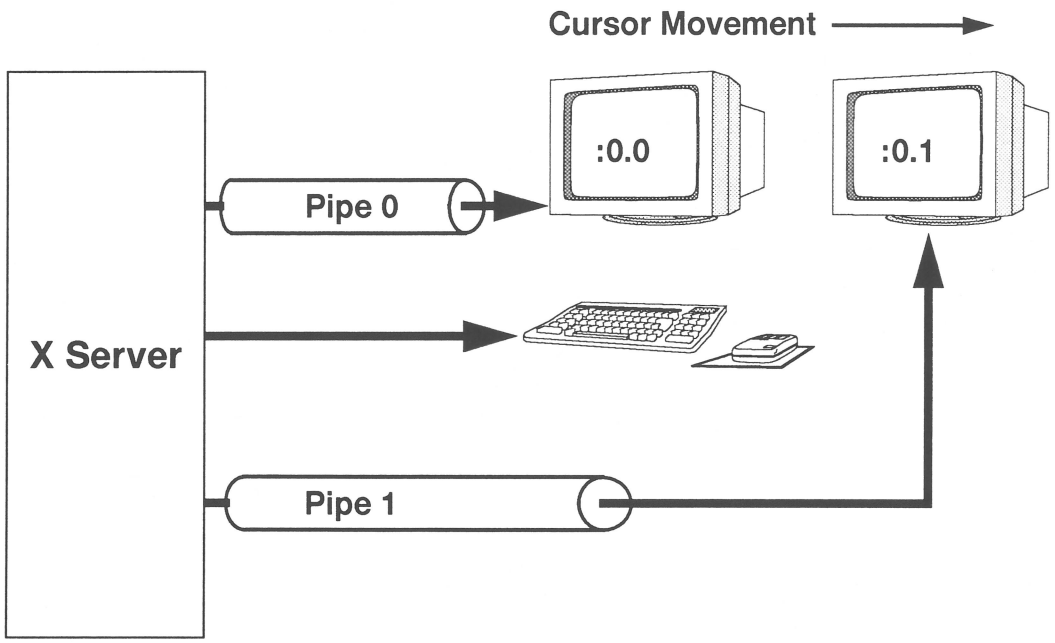
```
(/usr/gfx/stopgfx;/etc/killall xdm;/usr/gfx/startgfx) &
```

Note: The parentheses and ampersand are necessary to ensure that the command line completes correctly.

Use the `ircombine` GUI to configure your channel attributes:

0-23 FOIL: Screen Positioning

The screen 0 monitor of a multi-pipe system is normally configured in the leftmost position with each subsequent screen's monitor is physically positioned to the right of the previous monitor. This configuration allows the cursor to leave a monitor at its right edge and enter the next sequentially numbered screen at its left edge. See the illustration below:



Screen Positioning Options

You can configure the way that the X server sees the pipe's logical position by editing the **/usr/lib/X11/xdm/Xservers** file. There are two options that affect the placement of screens:

- **-stacked**
- **-hw**

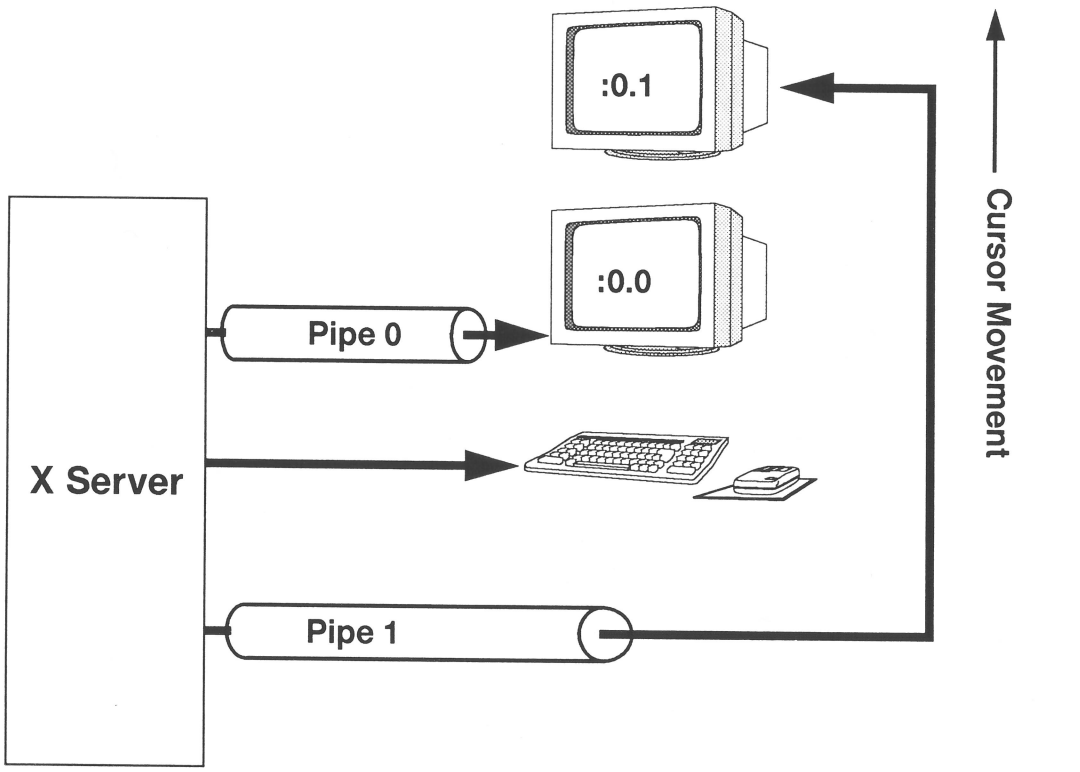
0-24 FOIL: Screen Positioning (Cont.)

-stacked Option

The **-stacked** option positions screen 0 as the bottom-most screen with additional screens positioned above it. In this configuration, the cursor will leave the topmost edge of screen 0's monitor and enter at the bottom edge of screen 1's monitor. Refer to the illustration below:

Xserver file:

```
:0 secure /usr/bin/X11/X -bs -nobitscale -c -pseudomap \
4sight -solidroot sgilightblue -cursorFG red -cursorBG \
white -stacked
```



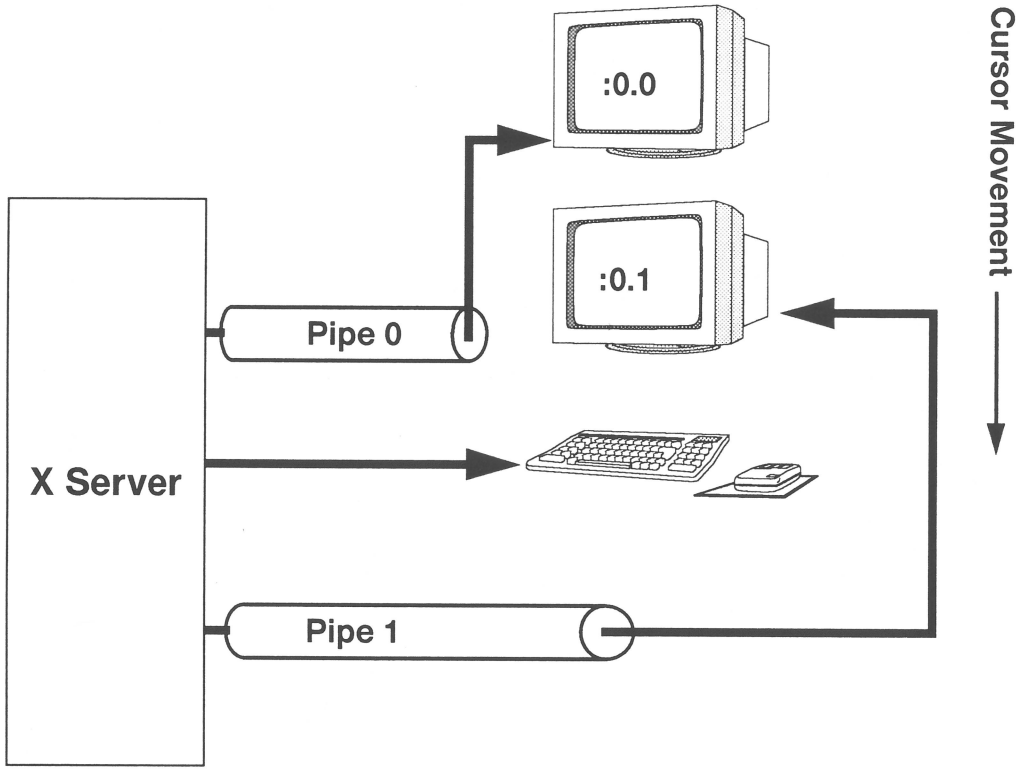
0-25 FOIL: Screen Positioning (Cont.)

-hw Option

The **-hw** option allows you to create a customized screen configuration. You can use -hw to position the screens in any order. In the following example, screen 0's monitor has been logically placed above screen 1. The cursor still moves from screen 0 to screen 1. Refer to the illustration below:

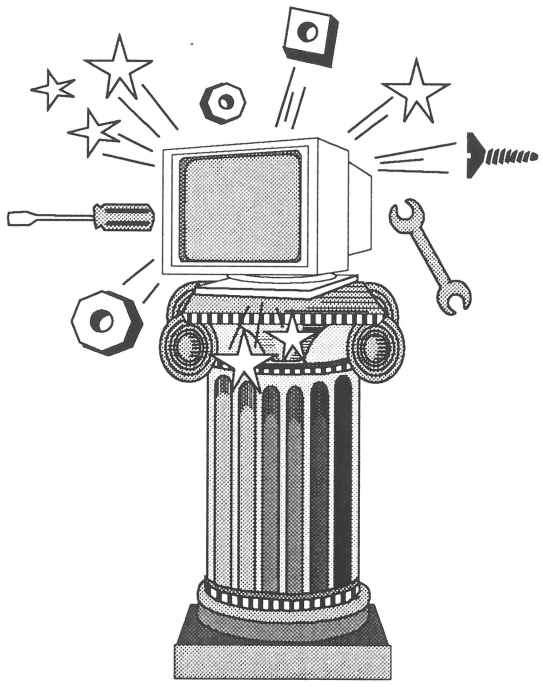
Xserver file:

```
:0 secure /usr/bin/X11/X -bs -nobitscale -c -pseudomap \
4sight -solidroot sgilightblue -cursorFG red -cursorBG \
white -hw board=0,below=1 -hw board=1,above=0
```



SiliconGraphics
Computer Systems

Module 14: Fault Isolation: Graphics



Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

14-2 FOIL: Module Objectives

The objective of this module allows each student time to apply the knowledge learned during modules 12 and 13, and reinforce that learning by troubleshooting instructor-induced faults within the Onyx2™ graphics system.

- Each student must demonstrate that a majority of the instructor-induced faults can be identified using any available documentation or tool.
- Each student must demonstrate that diagnostics and special troubleshooting procedures can be used to locate induced failures.

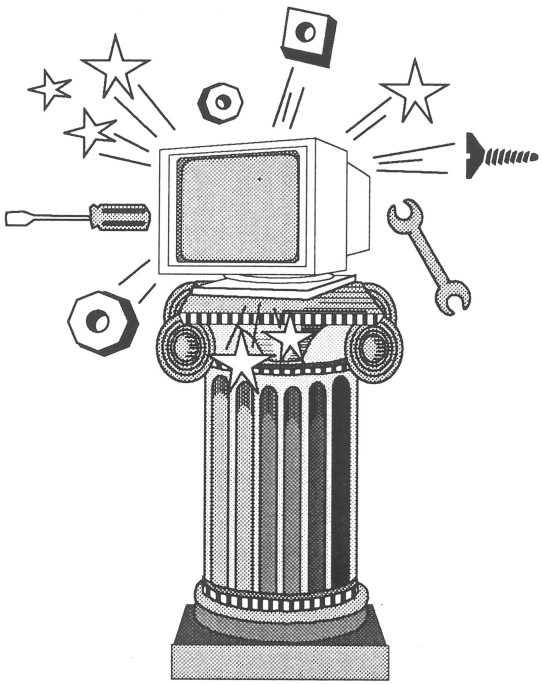
Student Notes

14-3 FOIL: Fault Isolation: Onyx2 Graphics System

- Student teams troubleshoot each fault together, and it is recommended that only **one team member make the keyboard entries.**
- Each lab system has a different instructor-induced fault, and as the team begins to evaluate a new fault, a **different team member** should be selected to **make the keyboard entries.**
- The team must leave a system in the **same** condition that it was in when they took control of the system.
- Time allotted to complete this lab: **4 hours.**
- Logic boards can be backed out, but **do not remove** or **replace** any components from the system. If a team wants to determine what would happen if a board or component is replaced, ask the lab instructor how the replacement would effect the system: since lab time is important, this allows the troubleshooting evaluation to proceed at a faster pace, and it also reduces the risk of another fault being accidentally introduced into the system.

SiliconGraphics
Computer Systems

Module 15: Component Locations & FRUs



Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

15-2 FOIL: Objectives

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

- Take the proper precautions to guard against ESD damage to equipment.
- Prevent damage to the exterior skins of the equipment.
- Handle the system components without damaging them.

Student Notes

15-3 FOIL: Module Topics

- General parts removal procedures.
- Proper handling of FRUs.

Student Notes

15-4 FOIL: General Parts Removal

There are several precautions that must be observed when removing or replacing parts on the system:

- Ensure that power has been removed before you install or remove parts.
- Use proper ESD precautions.
- Do not remove parts from the ESD bag until you are ready to use them.
- Have the proper tools at hand.
- Do not exceed the 1.25-inch bend radius of cables when you handle them.
- Do not touch connectors.
- Place skins that you remove in an area where they will not be scratched or damaged.

It is important that you guard against ESD when you perform the lab procedures. Always wear your smocks, wrist straps, and heel straps in the lab area.

It is also important that you keep the lab area clean. When you remove a FRU it should be, if applicable, stored in an ESD bag and placed in the racks provided.

Do not proceed with any of the lab steps until you are told to do so by the instructor. These procedures should be demonstrated to you before you attempt them. This will help prevent damage to our equipment!

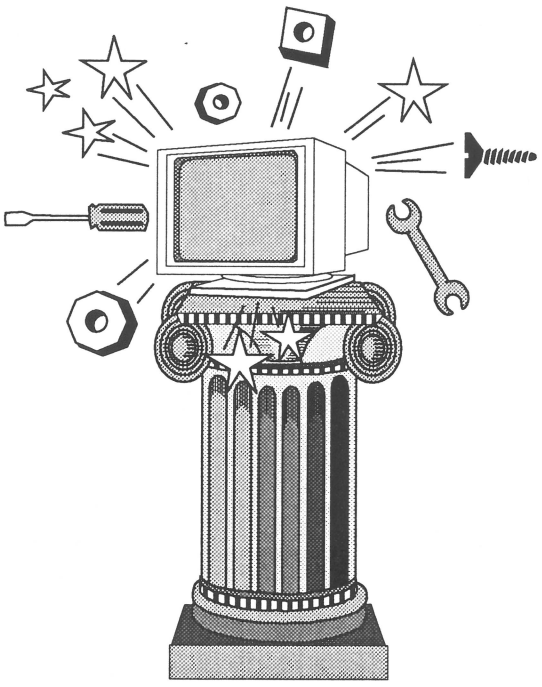
Student Notes

15-5 FOIL: Lab Project 7: Component Locations and FRUs

- Student teams will perform this lab project single Origin2000™ module removed from the rack system. **Modules 1** or **2** will be selected for removal because they contain **Base IO boards**.
- Time allotted to complete project: **3 hours, 30 minutes**.
- After completion of the lab project, the student should be able to:
  - Take the proper precautions to guard against damaging equipment due to ESD.
  - Identify all the FRUs in the Origin™ system module.
  - Properly handle all FRUs to prevent physical damage.
  - Remove all the FRUs.
  - Install all the FRUs.

SiliconGraphics
Computer Systems

Module 16: Origin200™ System



Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

16-2 FOIL: Objectives

Upon completion of this module, using available documentation and the classroom Origin200 systems, you will be able to:

- Identify the system logic carrier and its major subassemblies.
- Describe memory DIMM layout.
- Identify the various control panel switches.
- Describe how to configure a slave module.
- Interpret the MSC status LED display.
- List the system field replaceable units (FRUs).

Student Notes

16-3 FOIL: Module Topics

- System Overview
- Installation
- System Operation and LEDs
- FRUs

Student Notes

16-4 FOIL: System Overview

General Description

The Origin200™ system is a scaled down version of the Origin2000™ system. The Origin200 system uses the same processor chip as the Origin2000 system and is designed to function as a departmental server. The Origin200 system does not have any built-in graphics capability. The system is available in tower and rackmount configurations and can also be cabled to another Origin200 system in a dual-module configuration.

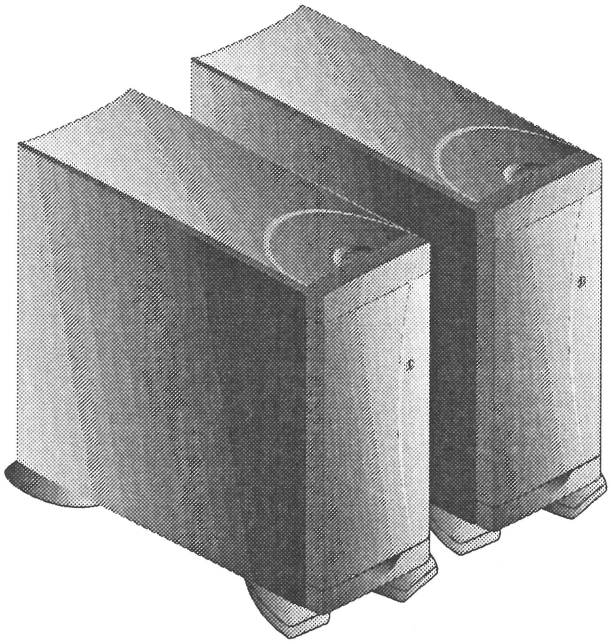
Features

- One or two R10000™ processors (T5 chips).
- On-chip instruction and data cache (32 KB).
- 1 MB secondary Cache per processor.
- 32 MB to 2 GB of memory.
- MSC.
- Two SCSI channels (internal): 1 single-ended Ultra SCSI, and 1 single-ended Fast and Narrow.
- Space for one full height or two half-height 5.25-inch and six 3.5-inch drives.

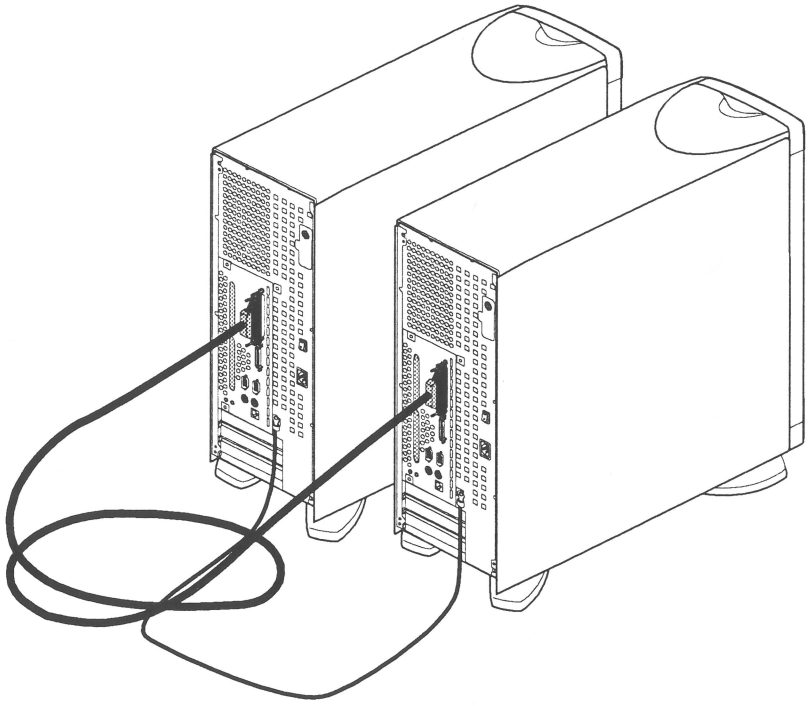
Student Notes

16-5 FOIL: Origin200 Profiles

Dual Origin200 Modules Front View

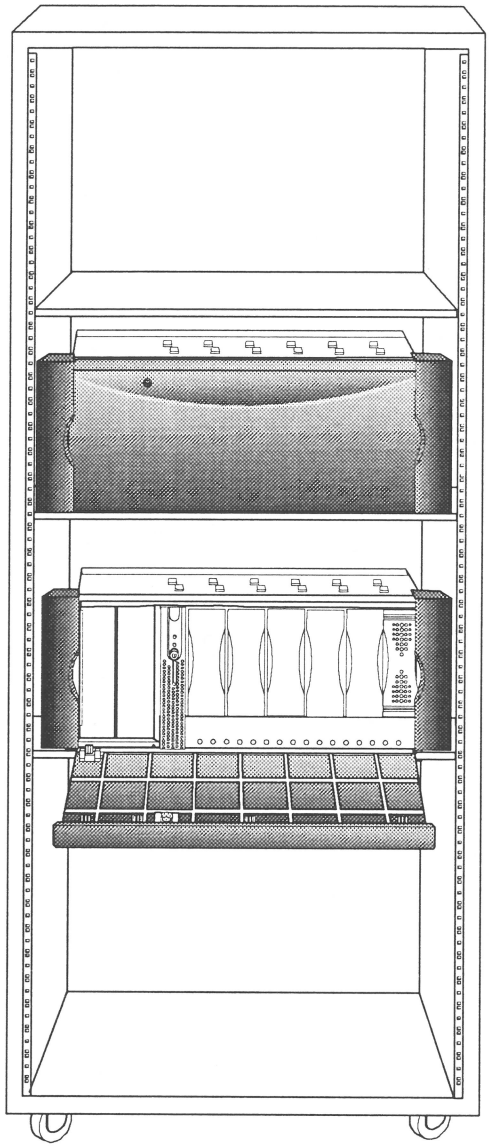


Dual Origin200 Modules Back View



Student Notes

16-6 FOIL: Rackmount Origin200



Student Notes

16-7 FOIL: Logic Carrier

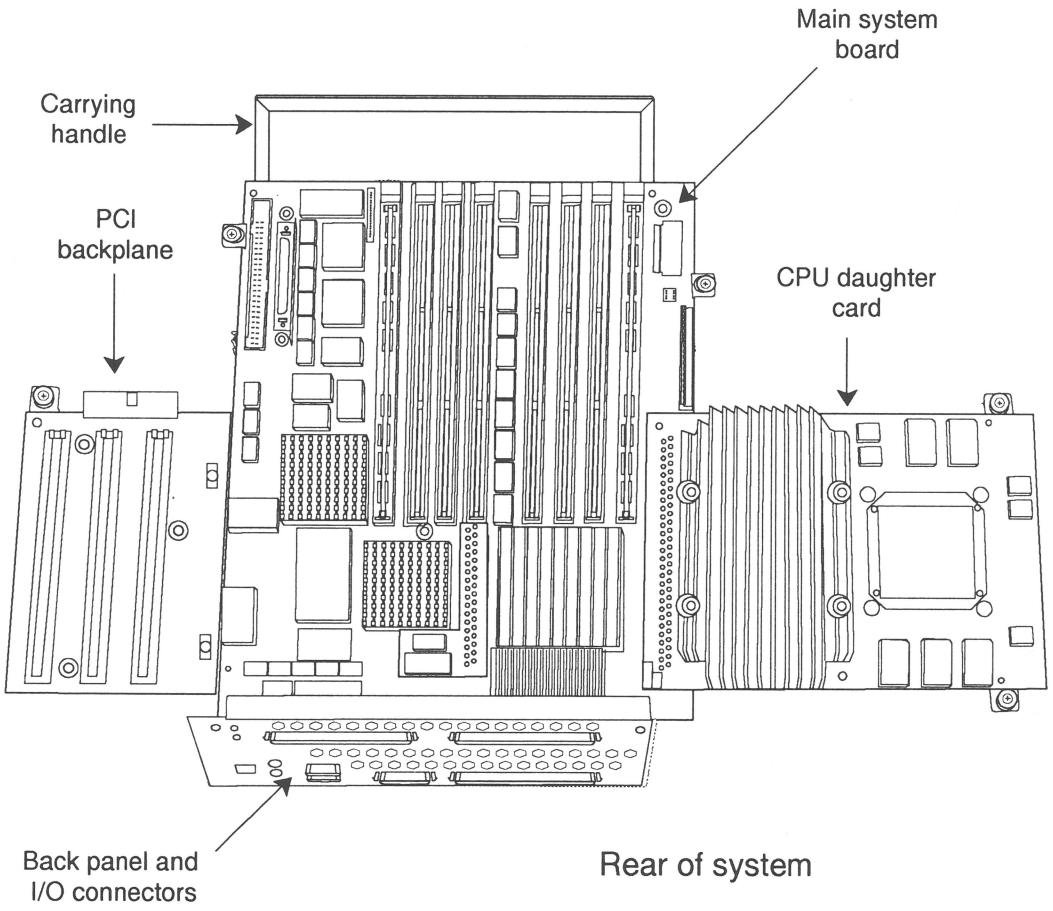
The system boards are mounted to a single piece of sheet metal to form an assembly called the logic carrier.

The logic carrier contains:

- Main system board.
- CPU daughter card.
- PCI backplane.

Note: The Carrier, Logic board, and PCI backplane are a single FRU.

Origin200 Logic Carrier



Student Notes

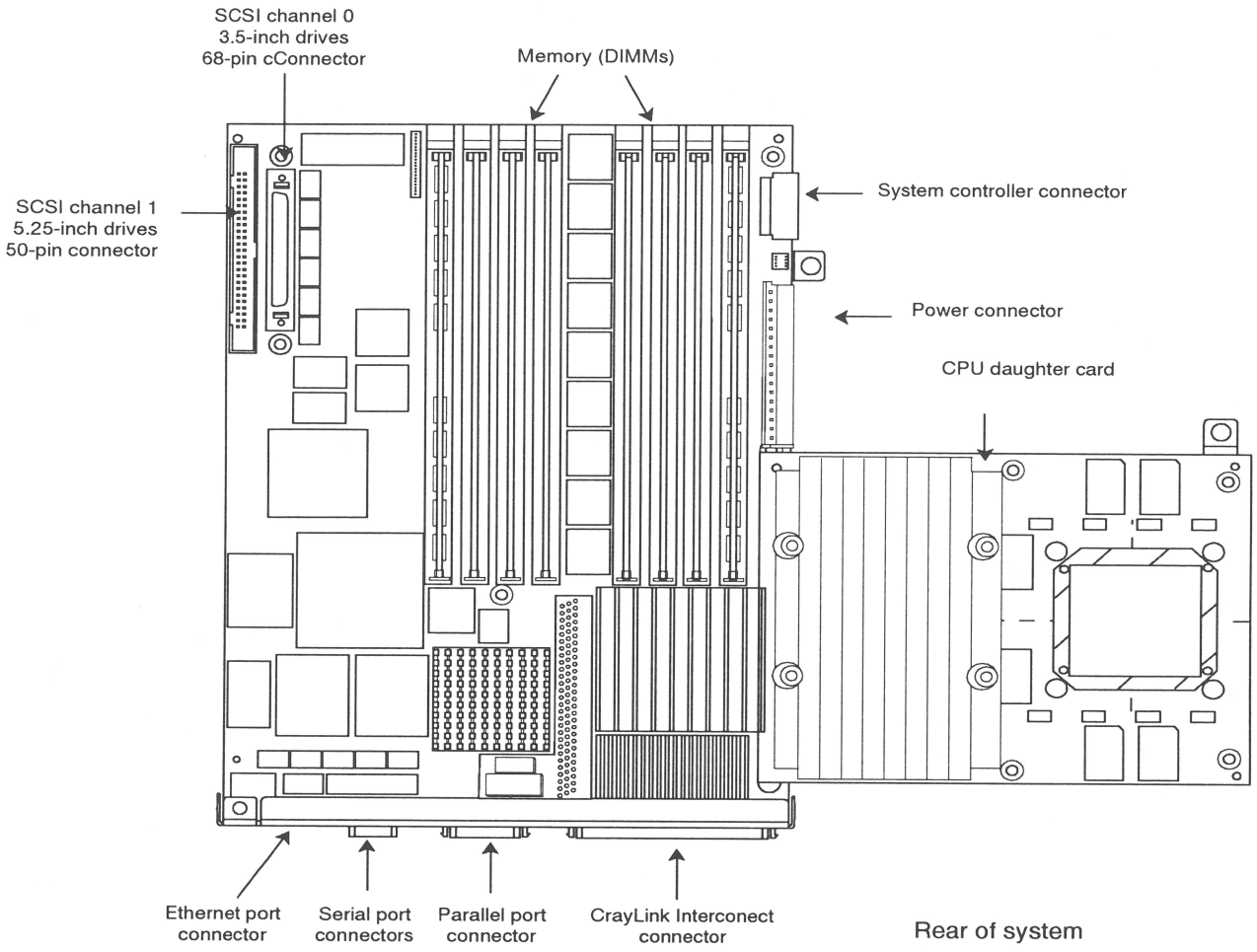
16-8 FOIL: Main System Board

The main system board contains:

- Memory DIMMs.
- 2 SCSI controllers.
  - 1 single-ended Ultra SCSI.
  - 1 single-ended, Fast, Narrow.
- Ethernet controller: 10BaseT or 100BaseT.
- 2 serial port interfaces: RS-232 or RS-422.
- Parallel port interface.
- 3 PCI expansion slots.
- CrayLink™ connector.

Student Notes

16-9 FOIL: Main System Board (continued)



Student Notes

16-10 FOIL: Memory Layout

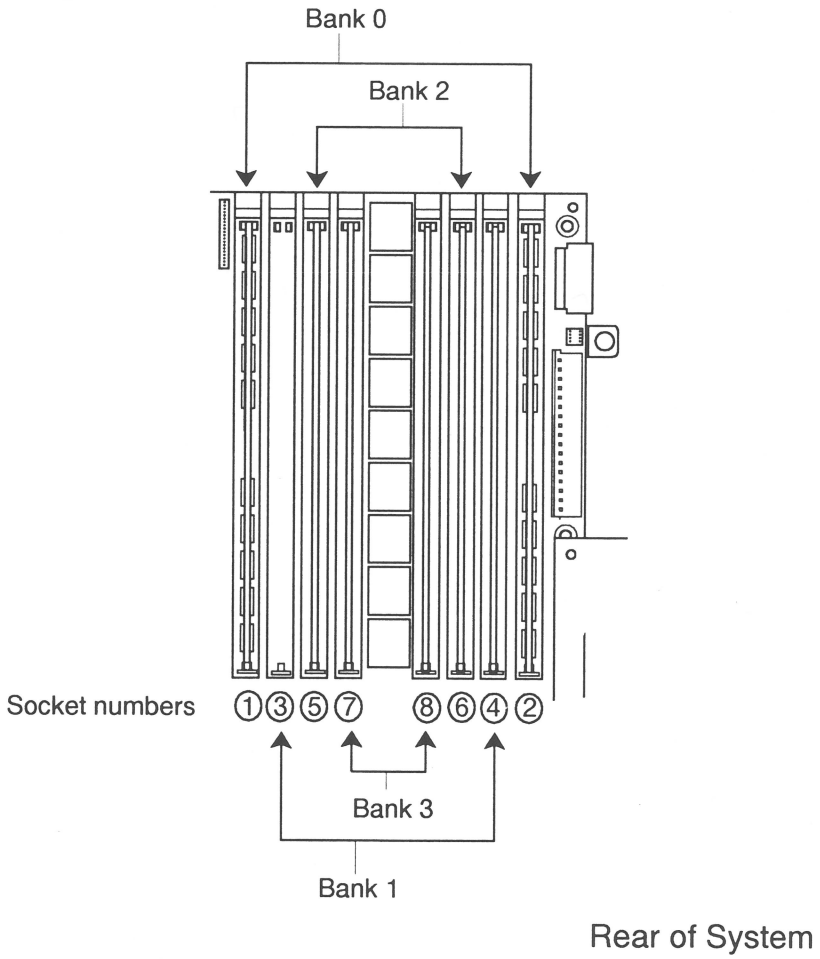
Memory is located on the main system board and has the following characteristics:

- Four banks of memory, eight DIMM sockets (two DIMM per bank).
- 32 MB to 2 GB.

The following rules apply to memory:

- Each module must have memory (32 MB minimum).
- Memory must be installed starting at Bank 0.
- DIMM must be installed as pairs and must be the same size.
- Memory can be mixed between banks.

Note: ***hinv -t memory*** will display the current size of memory.



Student Notes

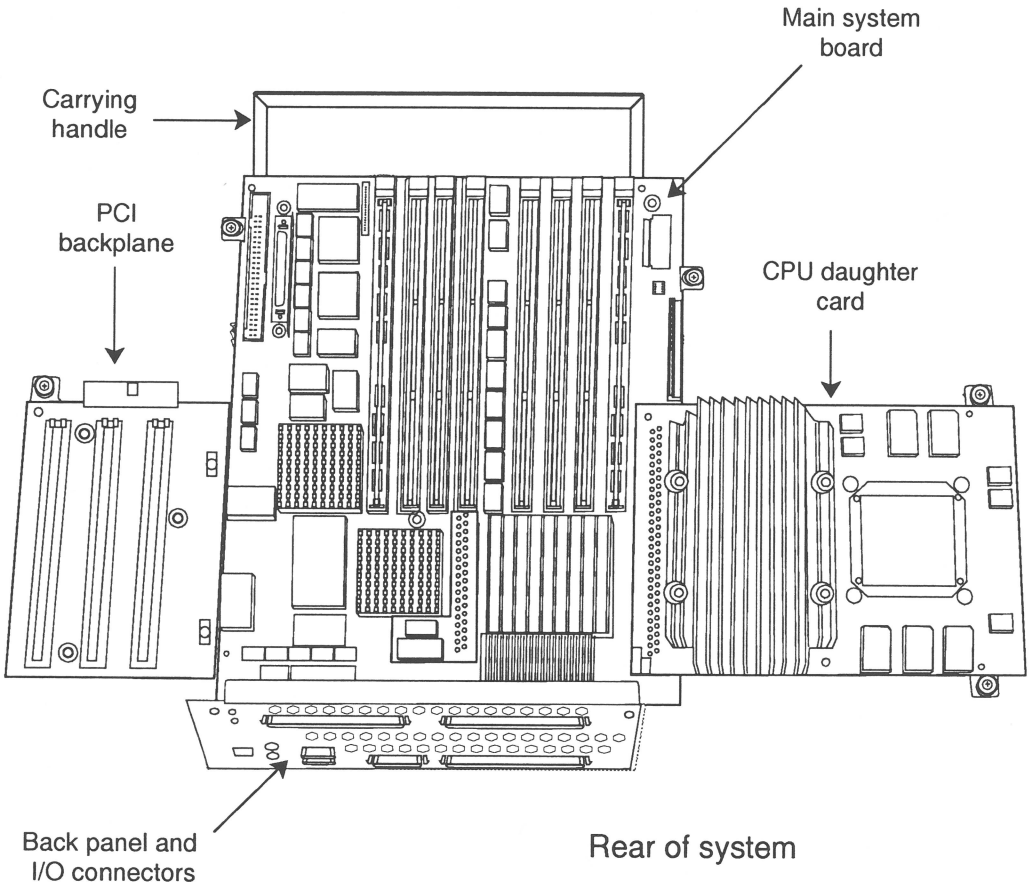
16-11 FOIL: CPU Daughter Card

The CPU daughter card is mounted to the main system board and has the following characteristics:

- Contains one or two R10000 CPU chips.
- Is a separate FRU.

Note: Do not attempt to remove the individual CPUs; the Daughter board is the FRU.

Logic Carrier With CPU Daughter Card



Student Notes

16-12 FOIL: PCI Backplane

The PCI backplane provides alternative controllers for external devices and networks. The PCI backplane has the following characteristics:

- Supplies the 5 volts and 3.3 volts to the PCI cards.
- Provides the only method of attaching external SCSI devices.
- Slots available for three PCI cards: IDs 5,6,7.

Student Notes

16-13 FOIL: Module System Controller (MSC)

The MSC is used to control and monitor the system. The MSC has the following characteristics:

- Used for power functions of system (power on/off, reset, nonmaskable interrupt - NMI).
- Displays system status via LED.
- Contains the NIC that holds the system serial number.
- Functions can be done remotely using the AUX port.

Note: When you replace the MSC, you must transfer the NIC from the old MSC to the replacement MSC.

Student Notes

16-14 FOIL: Drive Bays

Two internal SCSI buses are used to provide two sets of drive bays. The 5.25-inch bay has the following characteristics:

- Can contain any of the following devices: one full-height drive, two half-height drives, or two 1-inch drives.
- Controlled by SCSI channel 1 (configured as channel 3 in a slave module).
- Drive ID set by jumpers on drive (ID is from 1 to 7, ID 0 is reserved for the controller).

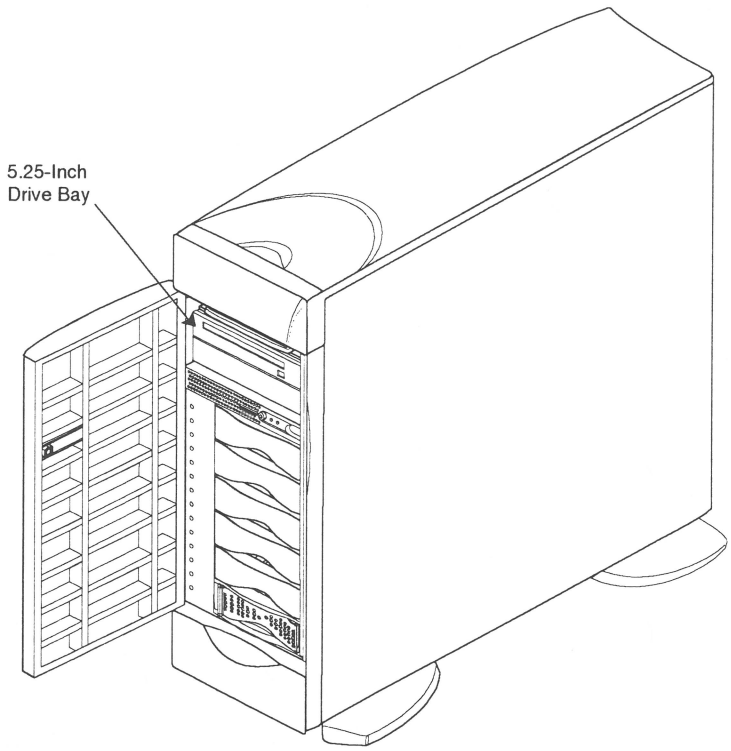
The 3.5-inch bay has the following characteristics:

- Can contain up to six drives.
- Controlled by SCSI channel 0 (channel 2 in a slave module).
- Drive ID set by slot location (ID 1 at bottom to ID 6 at top).

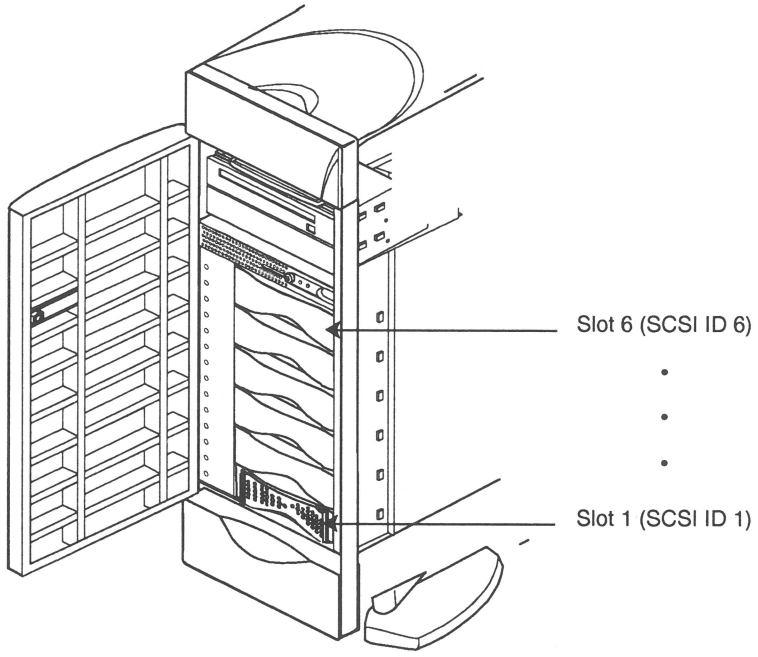
Student Notes

16-15 FOIL: Drive Bays (continued)

5.25-inch Drive Bay



3.5-inch Drive Bays



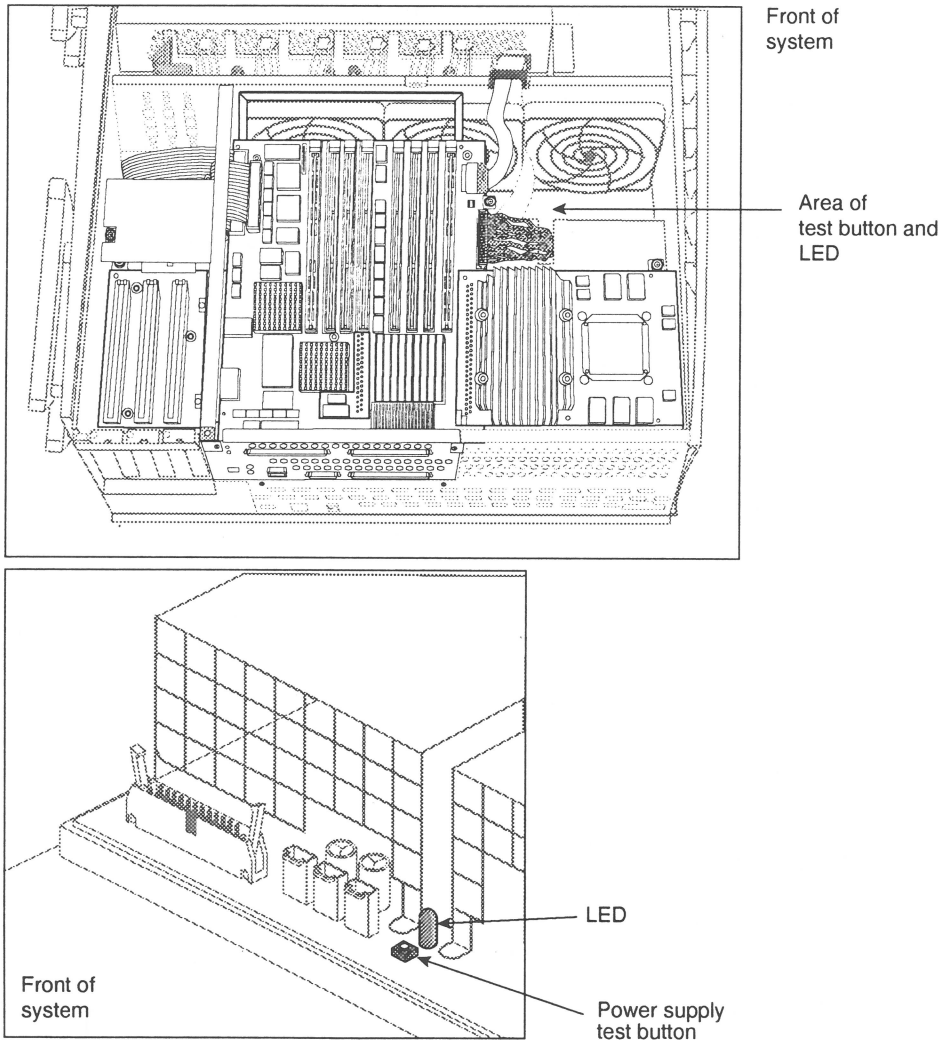
Student Notes

16-16 FOIL: Power Supply

The power supply has the following characteristics:

- Autoranging from 100 Vac to 240 Vac.
- 480-watt output +5 Vdc, +12 Vdc, -12 Vdc.
- 2-kVA power consumption.
- Equipped with a test button to verify the supply.
- Located underneath the logic carrier (carrier must be removed to replace the power supply).

Power Supply Test Button



Student Notes

16-17 FOIL: Installation

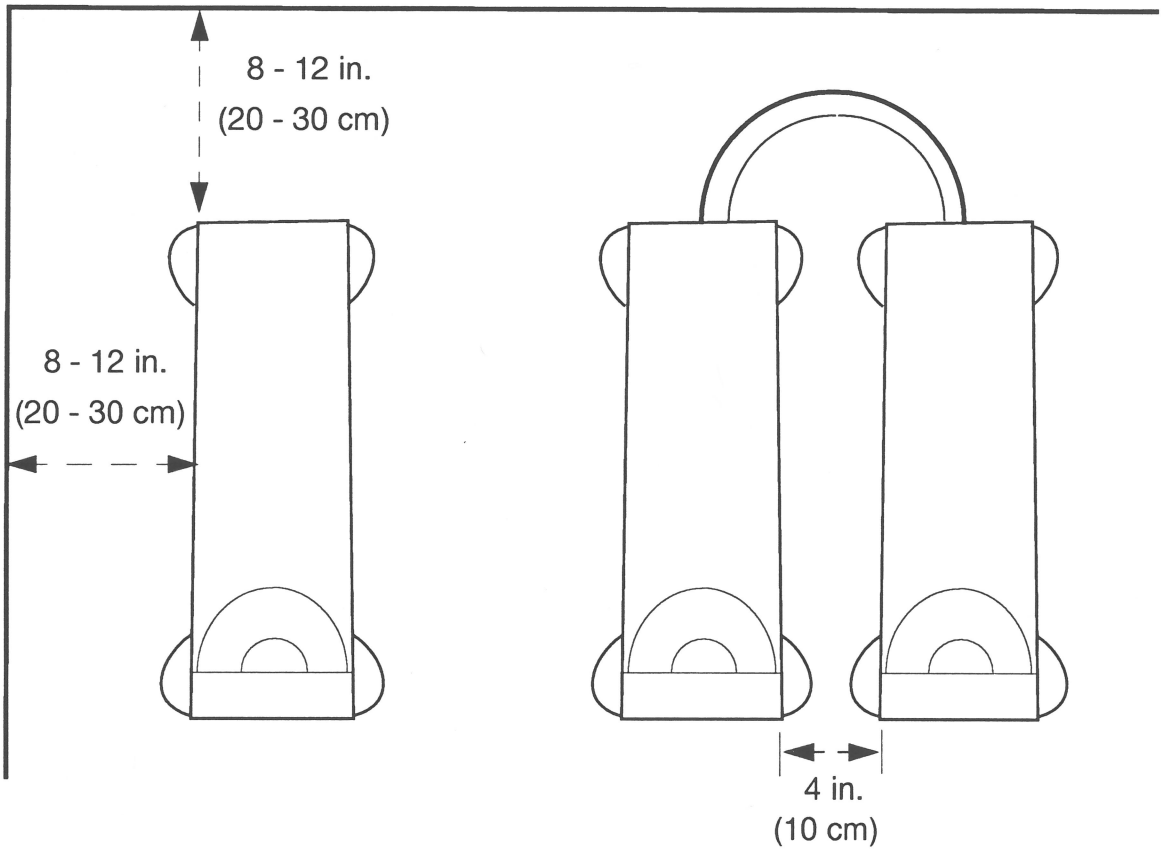
The Origin200 system can be installed in either a tower or rackmount configuration. The system can also be installed in a dual-module configuration (master-slave) and can be connected to an Origin™ Vault peripheral system.

Environmental Considerations

When installing Origin200 systems, follow these guidelines:

- Ensure proper site preparation (see site preparation guide).
- If you install dual modules, you must be connect them to a common ground!
- Ensure proper airflow.
- Observe clearance requirements (see below).

Clearance Requirements



Student Notes

16-18 FOIL: Standalone Installation

Follow these steps when you install a standalone Origin200 system:

1. Unpack the system.
2. Cable the system.
   a. Connect power cable.
   b. Attach Ethernet cable.
   c. Attach system console (or alternate display/computer) to serial port 1.
3. Power on the system.
   a. Turn on the master power switch on the back of the system.
   b. Press the front power switch on the control panel.
4. Boot the system, log on, and run SVP.
5. Shut down the operating system and power off the system.
6. Install any additional PCI interfaces, memory parts, or drives.
7. Boot the system, log on, and run SVP.

Student Notes

16-19 FOIL: Dual-module Installation

Follow the steps above and include these additional steps when you install a dual-module Origin200 system:

1. Unpack both modules and configure one module as slave.

   a. Access MSC through MSC access plate.

   b. Change DIP switch setting for slave operation.

2. Attach ground cable between modules.

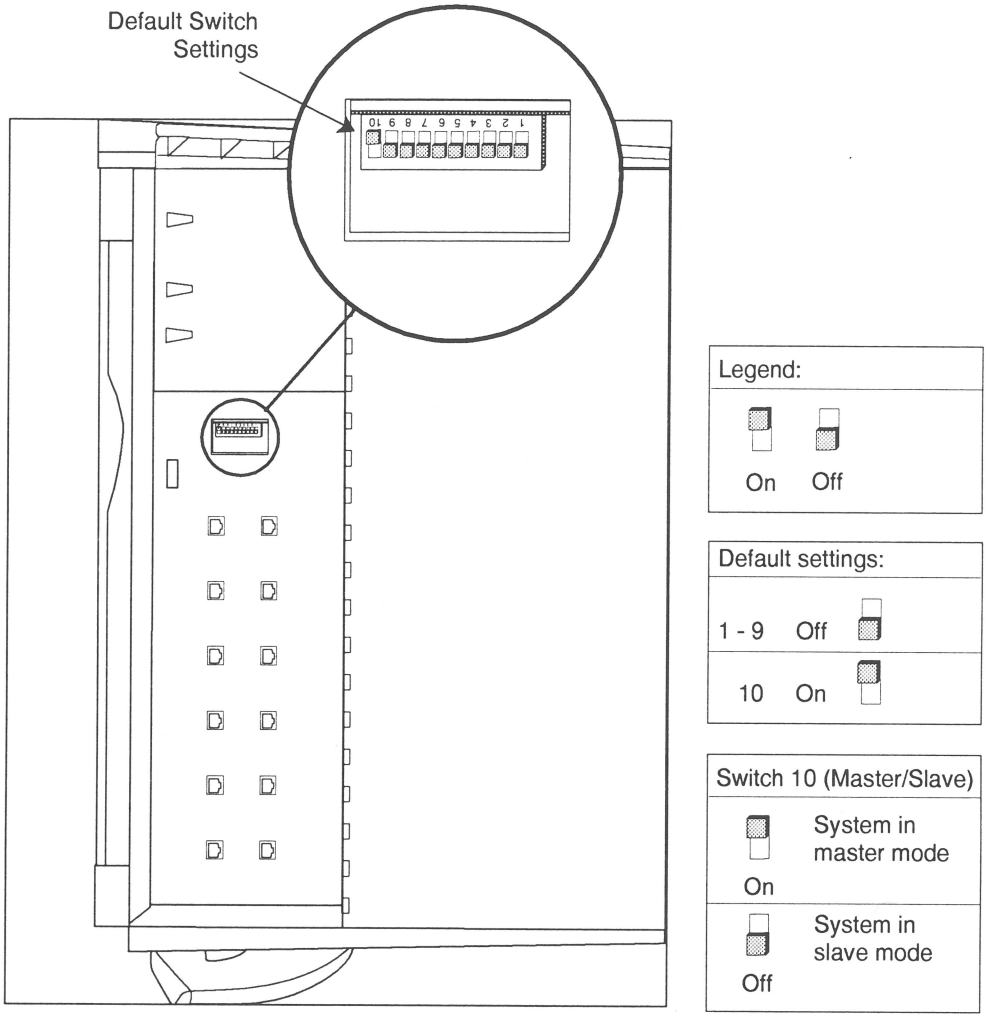
   Note: It is very important that modules have a common ground. Failure to do this will result in damage to the system.

3. Connect CrayLink cables.

4. Power on system from Master MSC.

Student Notes

16-20 FOIL: MSC DIP Switch Settings



Student Notes

16-21 FOIL: Rackmount Installation

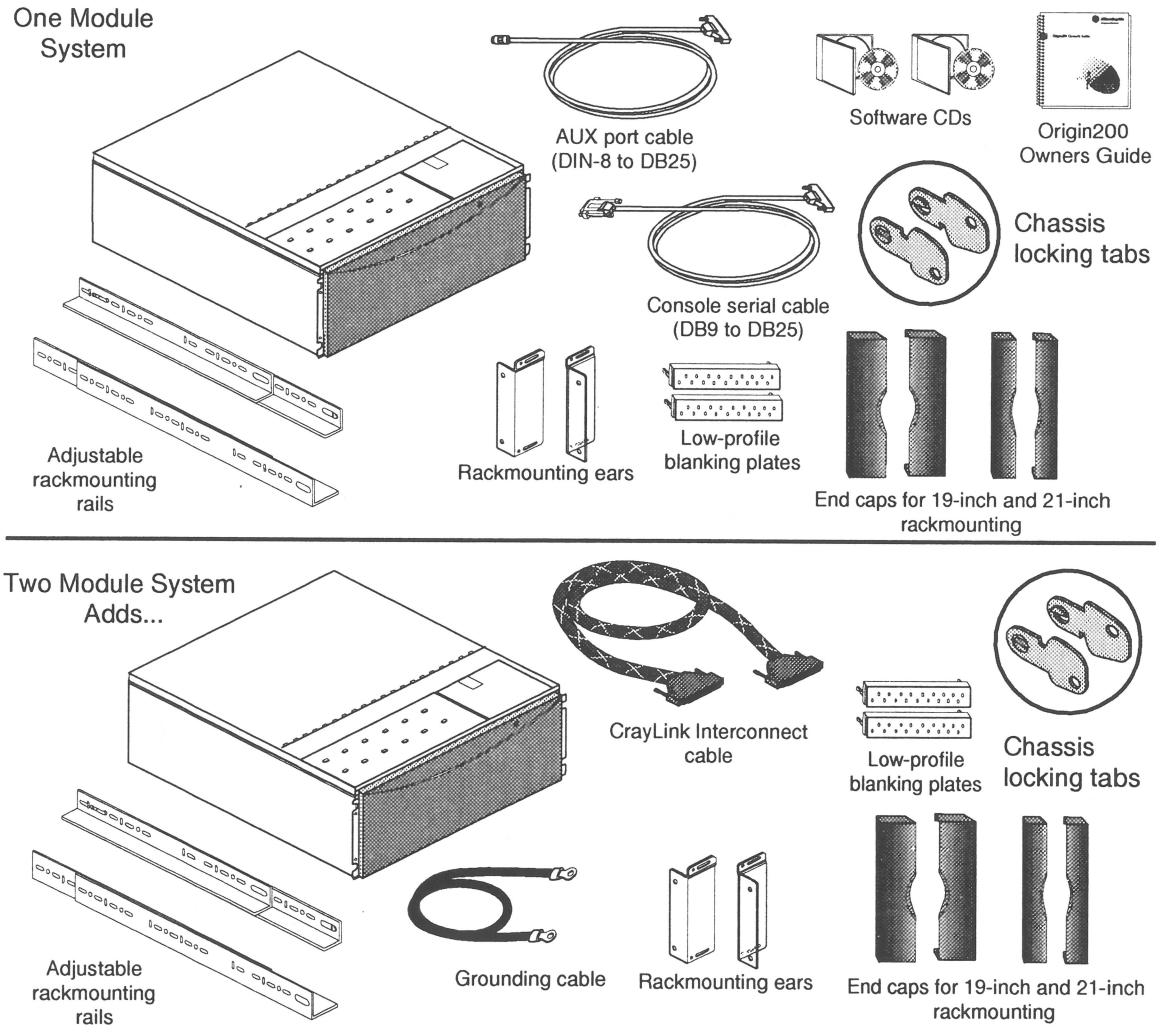
Follow these steps when you install a rackmount Origin200 system:

- Attach mounting ears to side of module.
- Install mounting rails in rack.
- Install module in rack.

  Note: To avoid injury, always use two people to lift and install the system into the equipment rack.

- Secure module in place.
- Attach plastic end caps.
- Cable system.

Rackmount Components



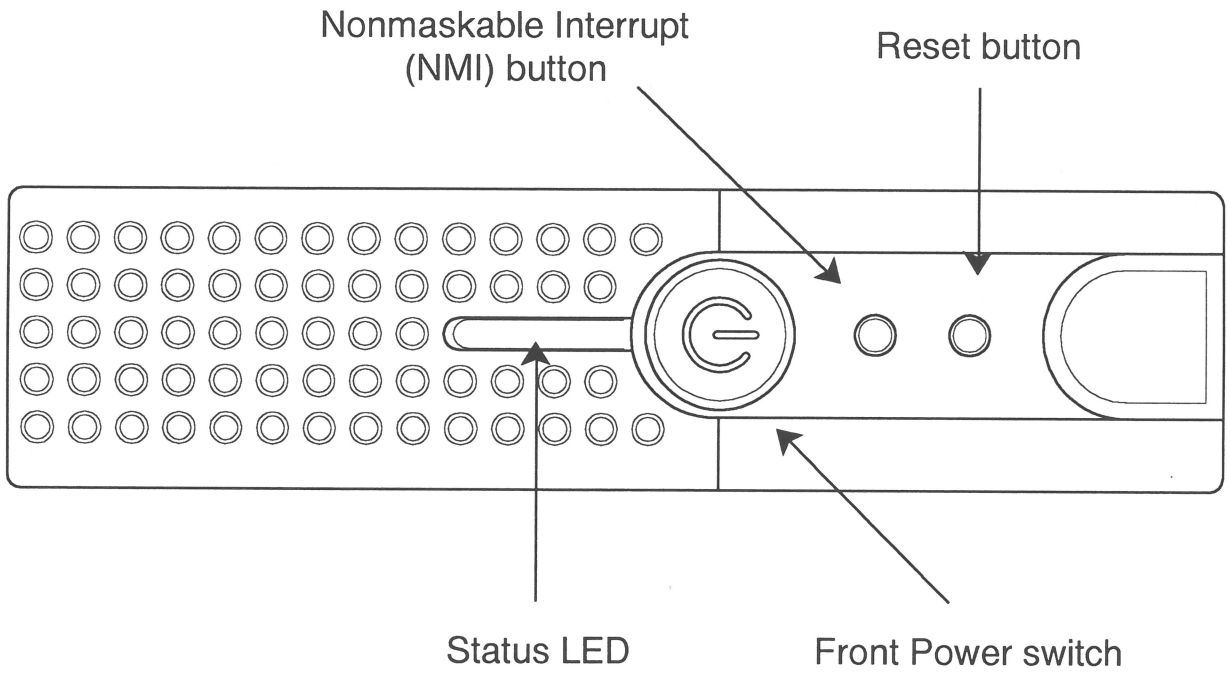
Student Notes

16-22 FOIL: System Operation and LEDs

The front panel includes the following items:

- System status LED (tri-color).
- Power button.
- Reset button.
- NMI button.

Front Panel



Student Notes

16-23 FOIL: Module System Controller

The MSC is used to control the Origin200 system. When using the MSC, note the following items:

- The MSC is accessed using the AUX port on the back of the system (with a terminal, modem, or another computer).
- All commands must be proceeded by a **<ctrl> t**.
- The MSC uses many of the commands that are used on the Origin2000 MSC.

A general listing of the Origin200 MSC command set is listed below; for more detail, refer to the *Origin200 Owner's Guide* document number 007-3415-002.

Command	Description
aut	Display current setting or change automatic power-on feature
clr	Reset all MSC options
dbg	Set or display virtual and physical debug switches
ech	Turn on or off command echo
fan	Display or change fan speed setting
nmi	Send a nonmaskable interrupt to the system
pas *xxxx*	Enter supervisor mode (provide password)
pas s *xxxx*	Set the password
pwr	Display or change system power setting
rst	Send a hardware reset to the system
rsw	Returns current virtual and physical debug switch settings
sel	Select CPU to receive MSC input
see	Select a CPU's output for viewing
tmp	Display system operating temperature
ver	Report MSC firmware revision

Student Notes

16-24 FOIL: Diagnostics

The Origin200 system uses the same set of diagnostics that the Origin2000 system uses:

- Power-on diagnostics (POD).
- Micro-diagnostic kernel (MDK).
- IRIX™ based system verification program (SVP).
- Field diagnostics.
- Field stress tests.

Student Notes

16-25 FOIL: LEDs

The status LED on the front panel can be used to identify power and cooling problems.

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 fail (system still running)
Red - blinking during power-down	Power supply problem
Red - blinking then shut-down	Multiple fan fail, and/or over temperature condition
Red - solid then shut-down	Power supply failure during boot, or overtemperature condition

Other LEDs

There are two Ethernet LEDs on the back panel of the system:

- Green transmit LED.
- Yellow Packet detect LED.

A test LED is also located on the system power supply along with a test button. The test LED should illuminate when the button is pressed; if it does not illuminate, the power supply is probably defective.

Student Notes

16-26 FOIL: FRUs

The Origin200 FRUs are as follows:

- Logic carrier.
- CPU daughter card.
- SCSI backplane.
- DIMM.
- MSC.
- Drives.
- Power supply.
- Fans.
- Skins.
- Cables.

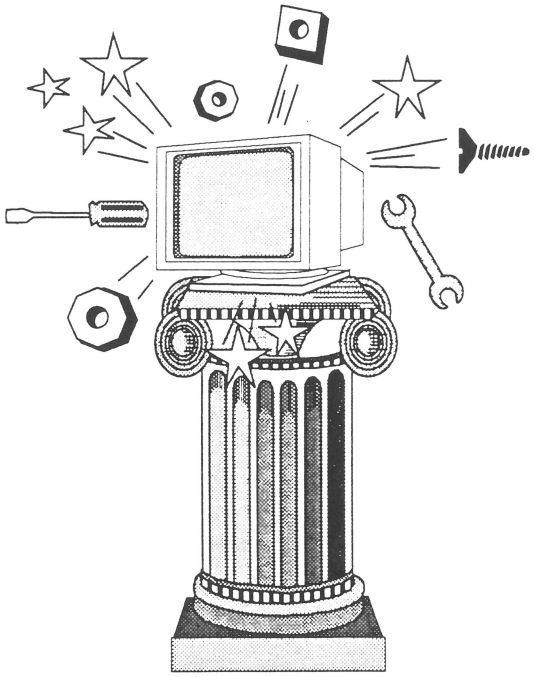
Student Notes

16-27 FOIL: Lab Project 8: Origin200

- Student teams will perform this lab project at a classroom rack system.
- Time allotted to complete project: **2 hours.**
- Time allotted to complete review questions: **30 minutes.**
- Time allotted to go over the review questions: **30 minutes.**
- After completion of the lab project, the student should be able to:
  - Unpack and install the system.
  - Identify Origin200 MSC command differences when compared to an Origin2000™ MSC.
  - Run the system verification program (SVP) to verify system functionality.
  - Remove and install all system FRUs.
  - Install a dual-module system.

SiliconGraphics
Computer Systems

Module 17:Support Tools



Part Number: LBT139-2.0-MR-6.4-S-SD-W
October1997

17-1 FOIL: Module Topics

- Introduction to Support Tools
- Availmon
- System Verification Program (SVP)
- Remote Access Tool (RAT)
- FRU Analyzer
- Sysmon
- Supportfolio™
- MDK
- IRIX-based Field Diagnostics
- Field Stress Tool (FST)
- System Monitoring Tools

17-2 FOIL: Objectives

At the completion of this session, the students will be able to:

- Configure Availmon and interpret the reports
- Query the SGI database for Availmon information
- Run SVP and interpret the results
- Query the SVP database
- Configure RAT on the Origin system
- Interpret the reports generated by the FRU analyzer
- Use Sysmon to monitor system activity
- View and download patches using Supportfolio
- Run MDK tests and interpret the results
- Run the IRIX-based field diagnostics and interpret the results
- Run the field stress tool and interpret the results
- Configure **linkstat** and **xbstat** and interpret the results

17-3 FOIL: Introduction to Support Tools

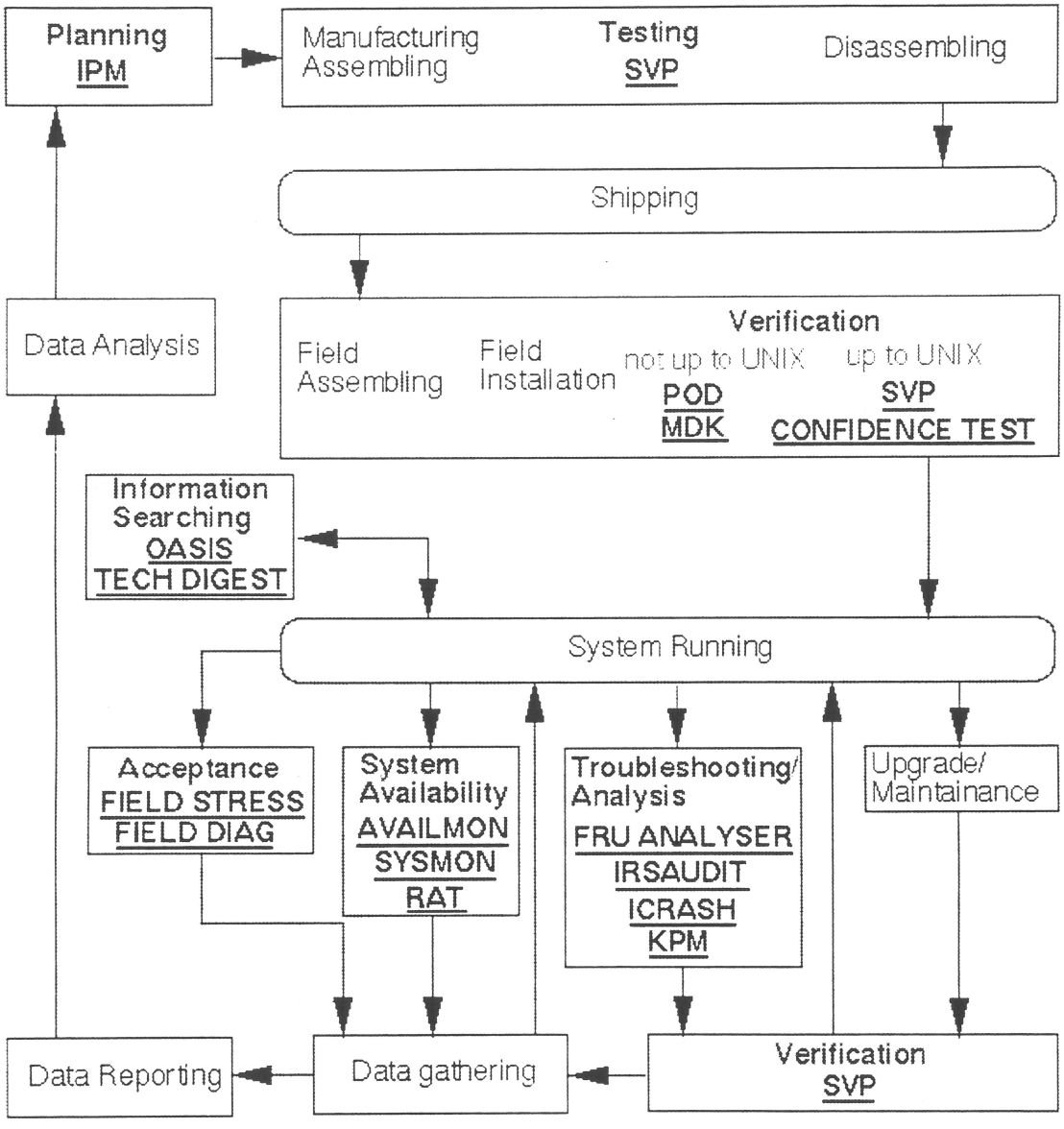
You should become familiar with the available support tools and how they are packaged and distributed. These tools are designed to help the service provider diagnose and report problems with the Origin and Onyx2 family of systems.

The following table lists the support tools and how they are packaged.

Support Tools Maintained by IST	O/S	IST CD	Web
SVP (System Verification Program)		X	X
ICRASH (Crash Utility)	X	Patch	
RAT (Remote Access Tool)		X	X
SysMon (System Monitor)	X		
IRIX-based Field Diagnostics		X	X
Field Stress Diagnostics		X	X
IPM (Installation Plan Manager)			X
Supportfolio			X
Other Internal Support Tools			
AvailMon (Availability Monitor)	X	Patch	
FRU analyzer	X	Patch	
POD	X		
IRSAUDIT (Graphics Diagnostics)	X	Patch	
KPM (Kona Post-Mortem)	X	Patch	
MDK (Micro-diagnostic Kernel		X	

17-4 FOIL: Support Tools Usage Block Diagram

The following block diagram shows where each support tool fits into the overall system maintenance strategy.



17-5 FOIL: Support Tools Release Strategy

A regular scheduled release is needed for any software product. This type of release serves as a vehicle for engineers to add new features to the software, improve performance, fix bugs, and so on. Because the Internal Support Tools Web site echoes the software tools on the Internal Support Tools CD, it does not need to have its own fixed release schedule. Instead, it follows the CD release schedule. This means whenever there is a CD release, the support tools images on the Web site are updated simultaneously.

A release occurs only if it addresses one or more of the following items:

- A bug that severely impacts the usability of the software
- Use of the software is overwriting or corrupting user data
- Security problems associated with the software
- Bugs/Features that impact SGI's contractual obligations or substantially enhance system performance

17-6 FOIL: Support Tools User Guides

There are user guides for several support tools. These guides are part of the support tools release and can also be downloaded from the Web at the following URL:

http://servinfo.cray.com

The following is a list of the available user guides.

- **SVP** - System Verification Program User's Guide (PostScript ™).
- **ICRASH** - IRIX Crash Analysis User's Guide (PostScript).
- **RAT** - Remote Access Tool User's Guide (PostScript).
- **Availmon** - Availmon User's Guide (PostScript).
- **FRU Analyzer** - FRU Analyzer User's Guide (PostScript).
- **MDK** - MDK User's Guide (PostScript).
- Diagnostic Roadmap for Origin2000 and Onyx2 Systems (PostScript).

17-7 FOIL: Support Tools URLs

IST Pages

- **IST Home Page**
  http://ist.csd.sgi.com/
- **IST Distribution Center**
  http://ist.csd.sgi.com/Tools/Home_pages/products_frame.html

Availmon

http://infohub.engr.sgi.com/availmon/

17-8 FOIL: Availmon

Availmon is a collection of programs that monitor and report on the availability of a system and diagnose system crashes. Availmon, along with the ICRASH and FRU Analyzer utilities, provides a platform for system availability and diagnostic data gathering and distribution.

Availmon is imbedded in the system boot and shutdown processes. It can capture and differentiate between a controlled shutdown, system panics, and system hangs. On high-end systems, it can collect the following data:

- Information on system failures and availability
- Config information
- Reboot data
- Controlled shutdown information
- System panics
- System hangs
- Power cycles
- Power failures

Availmon can also collect diagnostic information from:

- ICRASH
- SYSLOG
- Hinv
- Versions
- Gfxinfo

This information can be written to SGI databases.

Further Information

Refer to *Availability Monitor (Availmon) Reference Guide*, SGI part number 108-0169-002, for more information about Availmon.

This document can be downloaded from the following URL:

http://servinfo.cray.com/origin/index.htm

17-9 FOIL: Availmon Installation

Availmon can be installed from a patch for IRIX 5.3 systems and is included in the release of IRIX 6.1 and above.

- **IRIX 5.2**

  This is an old version of availmon with limited functionality. This version can only send availability reports, and the report format is different from the current version. To get the new version of availmon, please upgrade your operating system to IRIX 5.3 and install patch 1497 or newer version.

- **IRIX 5.3**

  Install patch 1497 from your local patch server. This patch replaces *eoe2.sw.availmon and eoe2.man.availmon* in IRIX 5.3. and updates ICRASH.

- **IRIX 6.0.1**

  This is an old version (Availmon601) of Availmon with limited functionality. This version can only send availability reports, and the report format is different from the current version. To get the new version of availmon, please upgrade your operating system to IRIX 6.1 or newer versions.

- **IRIX Releases Beyond IRIX 6.0.1**

  These releases will contain Availmon in the base IRIX. For IRIX 6.1, install patch 769 to fix known bugs and to upgrade ICRASH.

Note: Availmon was not released for IRIX releases 4.0.1 through 5.1, and 6.0.

17-10 FOIL: Availmon Registration (AMREGISTER)

You must register your system to activate Availmon. After installation and reboot, run **amregister** to configure Availmon, turn on autoemail, and register the system. For multiple systems, run amconfig on one system to set up a common e-mail configuration file (**/var/adm/avail/autoemail.list**) and copy this file to all the systems. Then run **amregister -r** on each system.

If you want a site log file for one or more systems, create a pseudo e-mail alias to pipe availability reports to **amreceive**, whose output is then appended to the site log file. You should follow this procedure before registering all the systems because initial availability reports will be sent out when registering. After setting up Availmon, run **amreport** on each system to view the availability statistics and reports for that system, or run **amreport**, using the site log file as input, to view overall availability statistics for all systems and availability reports for any system.

For examples of how to configure Availmon, refer to the user guide.

17-11 FOIL: Availmon Configuration (AMCONFIG)

Availmon has five configuration flags and an e-mail list that can be enabled or disabled with the **amconfig** command.

- **Autoemail** - enables automatic distribution of availmon reports using the list contained in the file: **/var/adm/avail/config/autoemail.list** (Default is off for all systems)
- **Shutdownreason**- enables querying for the cause of the system shut down (Default is on for high-end systems)

  Note: When you enable this option you are prompted for a shutdown reason
- **Tickerd** - enables running a light-weight daemon to monitor uptime. (Default is on for high-end systems)
- **Hinvupdate** - enables checking the output of hinv and gfxinfo commands (Default is on for high-end systems)
- **Statusinterval** - sets a status interval for sending availability reports to the Availmon database. If the interval is set to zero, no report is sent

17-12 FOIL: Availmon Reports (AMREPORT)

The Availmon program generates two types of reports: availability and diagnosis.

- **Availability** reports
  - System start time
  - Stop time
  - Stop reason
  - Uptime
  - Restart time
  - Summary of crash reasons (where relevant)
- **Diagnosis** reports
  - Crash analysis report
  - FRU Analyzer results
  - Syslog messages
  - System hardware/software configuration
  - Version information

Availability information is permanently stored in:

/var/adm/avail/availlog

Files in **/var/adm/avail** are maintained by Availmon and should not be deleted, modified, or moved. The most recent availability and diagnostic reports are stored in:

/var/adm/crash/availreport **and** */var/adm/crash/diagreport*

and are subject to removal like core dumps.

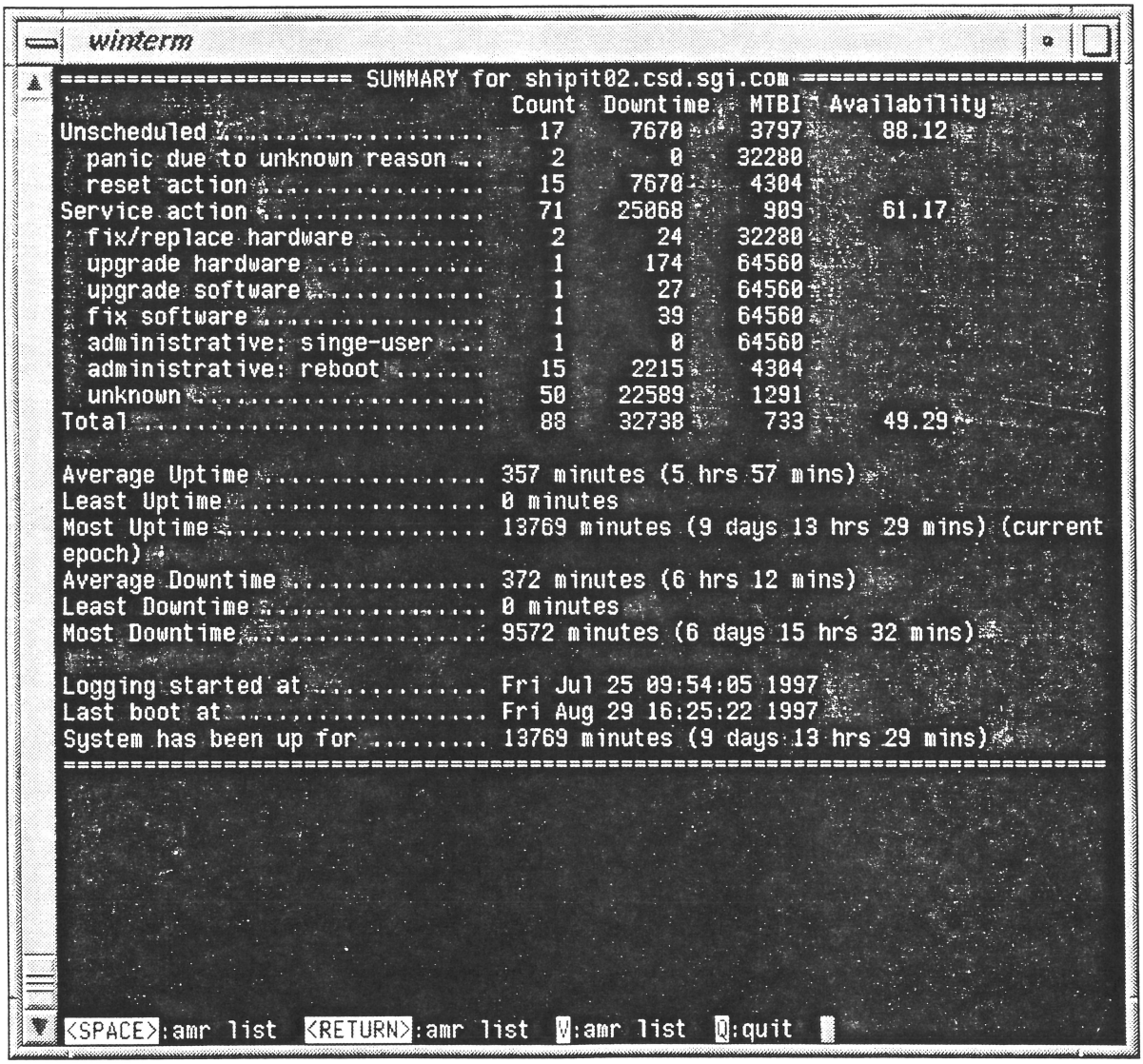
17-13 FOIL: Viewing Reports (AMREPORT)

There are two ways to view the report data:

- **Amreport** command
- Availmon Web site

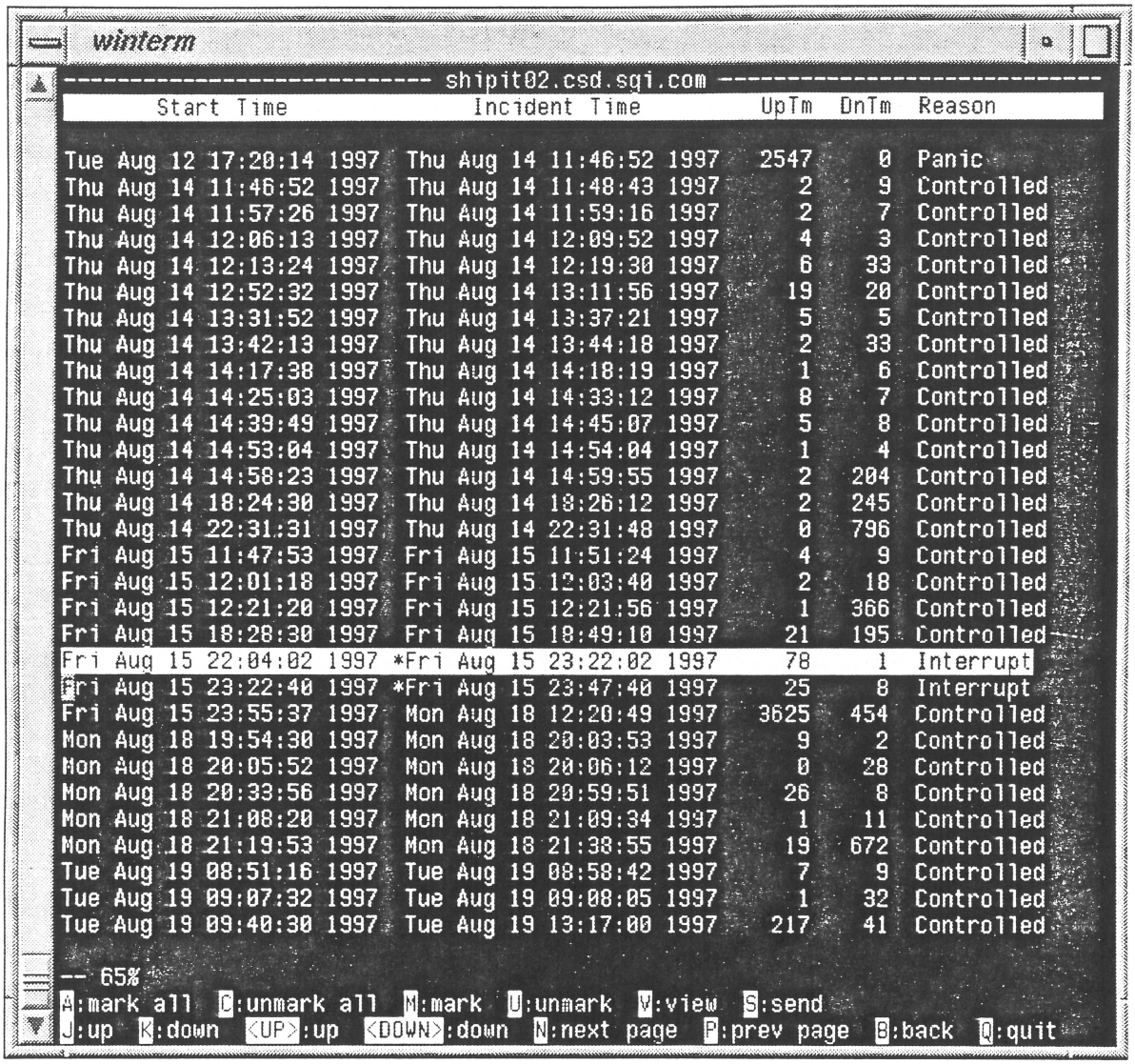
Amreport Command

The command **amreport** reviews Availmon reports and provides statistical availability information. This command processes local availability log files or aggregate availability reports (site log file) from different systems.



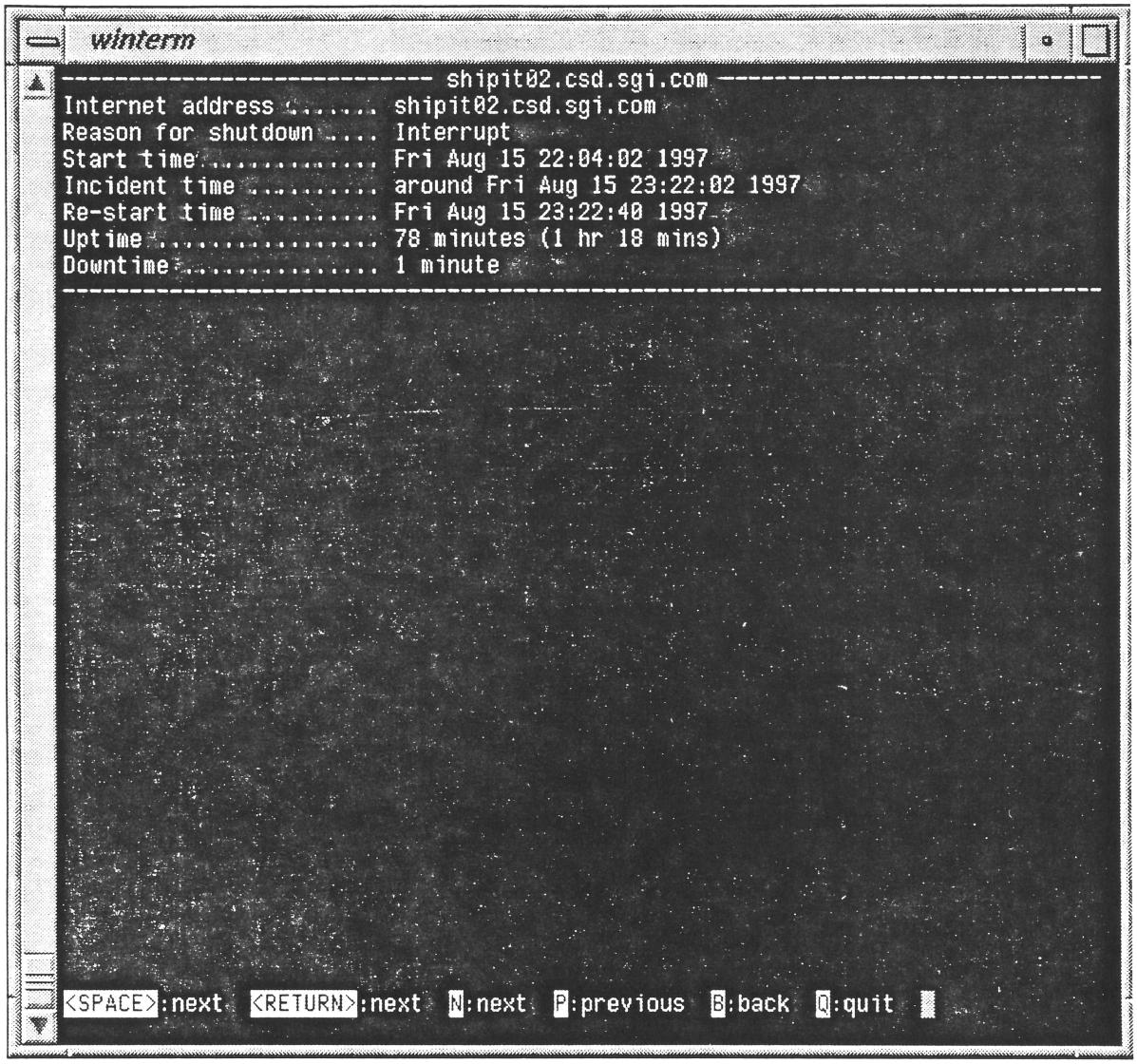
17-14 FOIL: Viewing Reports (AMREPORT) Cont.

The **amr** list displays individual incident entries. You can mark an entry in the log as shown in the example below and display more detailed information about the incident.



17-15 FOIL: Viewing Reports (AMREPORT) Cont.

Pressing the **v** key on a marked entry will display more detailed incident information similar to what is shown in the following example:



17-16 FOIL: Contents of the Availlog file

The **availlog** file resides in the **/var/adm/avail** directory. It contains a list of current patches, incidents, and output from the **hinv** command. You can use a common text editor to view the contents of this file.

Availlog contains an entry for each incident logged by the **tickerd** daemon called the **starttime** entry. An event number is associated with each starttime. This event number is the code given for the reboot reason. An example of the starttime entry and its associated event codes are listed below:

Starttime Entry

```
START|864137938|Tue May 20 07:18:58 1997
EVENT|7|864321299|Thu May 22 10:14:59 1997
UPTIME|3056
```

Event Codes

Uncontrolled shutdown

-3 Panic due to software faults
-4 Panic due to hardware faults
-5 NMI (hang)
-6 Power cycle
-7 Power failure
-8 Deregistration
-9 Registration
-10 No error (for live diagnosis)
-11 Hardware error (for live diagnosis)
-12 Software error (for live diagnosis) else Unknown reasons

Controlled shutdown:

-2 Shutdown reason unknown (not configured)
-1 Timed-out while waiting for operator to enter stop reason
0 User reset, cause unknown
1 Environmental reasons (move system, planned power outage,...)
2 Upgrade/add hardware or peripherals
3 Replace failed hardware component
4 Install software that requires reboot
5 Scheduled reboot/Routine maintenance
6 Software problem
7 Other unclassified reasons

Note: Event codes are automatically interpreted by the **amreport** command

17-17 FOIL: Availmon Query Pages

Another way to view Availmon data is on the web at:

http://bits.csd.sgi.com/availmon/

Note: This Web site is accessible only to employees of Silicon Graphics, Inc.

The Availmon query tool provides three query levels (basic, detailed, and custom); they are described on the Web page.

17-18 FOIL: Mailing Reports (AMSEND)

You can use the **amsend** command to manually transmit availability and diagnostic reports to a designated e-mail address or list of addresses. There are two ways to configure sending Availmon reports: automatic or manual.

Automatic

In the automatic process, the site can configure any number of recipients for each type of report. The recommended configuration is to send a diagnosis report to SGI for entry into the Availmon database. Other examples of e-mail configuration include availability reports to the site system administrator and diagnosis reports to local SGI field service personnel. Use the **amregister** and **amconfig** commands to enable automatic e-mail.

Manual

In the manual process, two types of reports are created in the directory **/var/adm/crash**; **availreport** and **diagreport**. System administrators can edit or filter the reports to remove sensitive or unwanted information and then use the **amsend** command to transmit the reports.

Availmon can be configured to compress and encode data. At the receiving end, a conversion program named **amreceive** decodes, uncompresses, and stores the data into an SGI database. Data encryption, if not prohibited, is recommended for security reasons.

17-19 FOIL: Searching Availmon Reports (AMPARSE)

The **amparse** command allows you to search for specific information within an availmon report. You can use command line options to specify different search criteria. This command is very useful for quickly obtaining system service information. Two examples are shown below:

amparse -HINV /var/adm/avail/availlog

This example would search the **availlog** file for all entries generated by the **hinv** command and redirect them to standard output.

amparse -VERSIONS /var/adm/avail/availlog

This example produces a list of all currently installed system patches.

17-20 FOIL: Availmon URLs

- **Availmon user Guide**
  http://ist.csd.sgi.com/Tools/Home_pages/tools_info.html
- **Availmon Introduction**
  http://infohub.engr.sgi.com/availmon/
- **Availmon query page**
  http://bits.csd.sgi.com/availmon/

17-21 FOIL: Service Verification Program (SVP)

The System Verification Program (SVP) is an IRIX- based software tool, that contains utilities and test programs. SVP inventories the system hardware and software components, loads and runs a suite of test programs, and generates a set of files that report the results of this activity to an SGI database.

SVP helps a field Systems Support Engineer (SSE) to verify the configuration and functionality of a newly installed system. SVP is a system acceptance test tool, and the test result is part of the system installation sign-off requirement for the SSE on high-end systems.

Rationale for Using SVP

- **The SVP process is a uniform approach that is used to verify the functionality of a newly delivered system**

  Running SVP is a mandatory requirement for all high-end systems acceptance trials.

- **The SVP process strengthens the link between Manufacturing and the field**

  SVP results in Manufacturing can be compared against results obtained in the field.

- **The SVP process ultimately reduces support costs**

  SVP gives the service provider and customer a predictable and consistent conclusion to the installation process.

- **SVP builds customer confidence**

  The SVP process assures customers that the complete shipment of hardware and software they have ordered has been delivered and that the assembled system is functional.

Further Information

Refer to *System Verification Program (SVP) Reference Guide*, SGI part number 108-0165-002, for more information about SVP.

This document can be downloaded from the following URL:

http://servinfo.cray.com/origin/index.htm

17-22 FOIL: SVP Testing Process

Manufacturing Testing Process

As the last step in the Origin system final integration-testing process, Manufacturing runs SVP to measure and document the functionality and performance of the system. This test immediately precedes shutting down, disassembling, and packaging the system for shipment to the customer.

Note: Manufacturing runs SVP on all Origin rack insert modules using a common set of overpack materials. In addition, Manufacturing tests all high-end systems in their final configurations using the customer's set of overpack items.

The SVP results and system configuration information are automatically logged into the Mosaic database (IB Database) after the Manufacturing run. You may view this information one day after a shipment is confirmed by Manufacturing through the Oracle® ordering system at:

http://illiad.corp.sgi.com/mosaic_queries/svp_config.html

http://illiad.corp.sgi.com/mosaic_queries/svp_testresult.html

You must provide your system serial number to view this information.

Site Testing Process

After delivery, installation, and initial power-on diagnostic testing at the customer site, the SSE retests the system with SVP, without overpack items installed. (Perform "Out of Box" tests first.) The results of this test session are compared to the results of the final Manufacturing test. Ideally, the results of both tests will match. Then the SSE should install the overpack items as the final setup of the customer configuration and run SVP again.

Subsequently, if the system installation plan requires that hardware or software modifications be made to the system, the SSE must ensure that the initial SVP session has completed successfully before the SSE proceeds with these modifications. Following system modifications, the SSE should again run SVP to establish a new configuration and return the new information to the SVP database. Running SVP is always the final step of any hardware or software modification activity.

17-23 FOIL: SVP Test Coverage

SVP runs the following tests to check the system.

- Peripheral confidence tests
- Board level diagnostics
    - Node
    - Memory
    - IO6
    - MSCSI
    - MIO
    - ATM
    - FDDI
    - HIPPI
- USRPCI diagnostics
- IRSAUDIT for graphics
- IRIX command level tests

17-24 FOIL: Viewing SVP Test Results

All SVP test results are stored in the file

/usr/SVP/RESULT/svp_*date*.LOG.

The results of **irsaudit** are stored in the file

/usr/SVP/RESULT/irsaudit_*date*.LOG.

All errors are also logged into the file

/usr/SVP/RESULT/svp_*date*.ERRLOG.

You may view these files with your preferred editor (for example, **vi**).

17-25 FOIL: SVP Configuration Data

All SVP configuration information is logged in the directory:

/usr/SVP/CONFIG

This directory contains the following files:

- **configdata**

  Contains an inventory list of all node, router, BaseIO, and XIO boards installed in the system.

- **installed_software**

  Contains an inventory list of all software and patches currently installed on the system.

- **mosaic_data.mps**

  Contains formatted information for the SGI SVP database.

- **mosaic_data.email**

  Contains formatted information for the SGI SVP database.

- **run_reason**

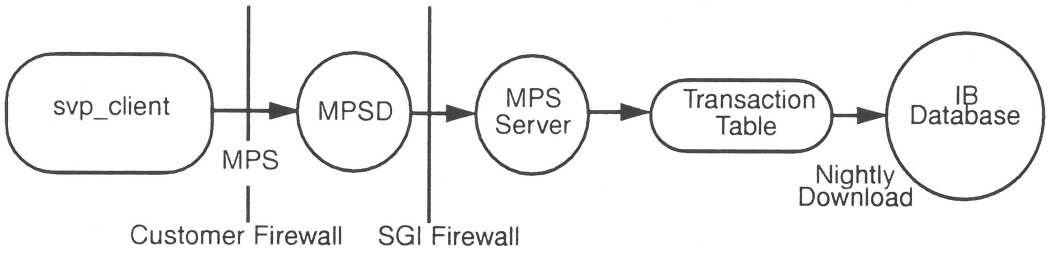
  All of comments that you entered at the beginning of the SVP run are inserted with a date entry into this file.

17-26 FOIL: SVP Client

The SVP software client is a program that returns information to the corporate Installed Base database, Some customers and secure sites do not allow information to be transmitted from their sites. Consequently, this client program is not automated; it must be run as an individual command line argument after running SVP to completion. There are two mechanisms to return information to the corporate database.

Message Passing System (MPS)

The first mechanism uses the MPS (Message Passing System) to transfer information to the database. This mechanism is preferred because it guarantees the delivery of information to the database. If transmission failures occur, the client will indicate how many records have been successfully written to the database and then automatically update the file that is being transmitted so that it contains only unsent information. To resume the transmission of unsent information, you must reinitiate the software client.



The svp_client program consists of a shell script and a binary file. You must initiate it from the command line after the completion of SVP by entering the following two instructions:

cd /usr/SVP

./svp_client

The svp_client program will print out a command completion message. If any errors occur, they are reported. If only partial data is sent, a message is issued to retry the program and the data file is automatically set to the remaining set of unsent data. After successful completion, both mosaic_data.nips and mosaic_data.email are removed.

17-27 FOIL: SVP Client (Cont.)

E-mail Client

An e-mail client is provided in the event that the customer is not on the Internet, and the SSE must copy the **mosaic_data.email** data file and e-mail it to SGI.

To manually send email, enter the following command:

mail svp@corp.sgi.com < /usr/SVP/CONFIG/mosaic_data.email

 # rm /usr/SVP/CONFIG/mosaic_data.email

The database is updated nightly. Data received from a site is available the following morning. A Web site has been created to view this data. It is accessible to SSEs who have access inside of the SGI firewall. The addresses for the configuration data and the test results are:

http://illiad.corp.sgi.com/mosaic_queries/svp_config.html

17-28 FOIL: SVP URLs

SVP General Information

http://bits.csd.sgi.com/digest/topics/Lego/status/RandD/SVP.html

SVP Installation Images

http://ist.csd.sgi.com/Tools/Home_pages/products_frame.html

SVP Query Page

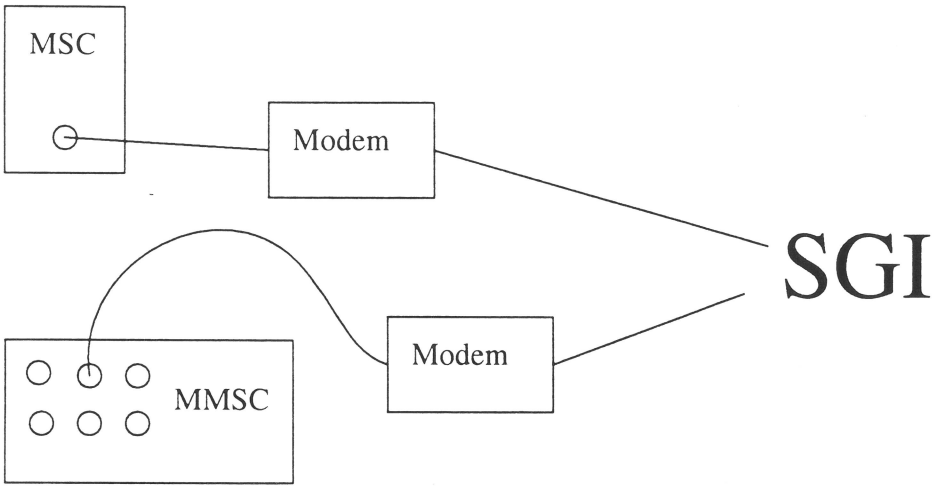
http://illiad.corp.sgi.com/mosaic_queries/svp_config.html

17-29 FOIL: Remote Access Tool (RAT)

The Remote Access Tool (RAT) is an application that you can use to interface to the system controllers on Origin2000, Origin200, and Onyx2 systems. It communicates with the system controller and allows a user to remotely control the entire machine through one interface. Most operations supported by the system controller command language for both the Module System Controller (MSC) and the Multi-Module System Controller (MMSC) are also supported via RAT.

RAT gives the user the ability to monitor and control a system remotely. As more modules are added to the system, the complexity of the Origin2000 system controller configuration increases. Therefore, RAT becomes vital to helping the user understand the configuration of the system remotely. Although you can enter system controller commands individually, it is normally faster to use RAT. In addition, the RAT commands are self-documenting, which makes the tool easy to use.

MSC to RAT connections



RAT was initially designed as an ASCII application that it would run on all SGI systems. There is now a version for laptop computers. It is important to remember that RAT resides in the workstation and not on the mainframe.

Further Information

Refer to *Remote Access Tool (RAT) Reference Guide*, SGI part number 108-0168-001, for more information about RAT.

This document can be downloaded from the following URL:

http://servinfo.cray.com/origin/index.htm

17-30 FOIL: RAT Functions

RAT can help the field engineer by performing the following functions:

- System reset
- NMI
- Fan control
- DIP switch read
- Power on/off/cycle
- Voltage margin
- Debug switch setting
- Interface to any CPU /module

17-31 FOIL: RAT Installation and Dependencies

Installation

RAT can be installed from the WCS Internal Support Tools CD out of the support image. Versions are available for IRIX 5.3, IRIX 6.2, IRIX 6.3, and IRIX 6.4. Please see the release notes on the Internal Support Tools CD for more details on how to install the version for your operating system version.

Dependencies

If the support image is being installed on an IRIX 5.3 system, the subsystem **eoe2.sw.uucp** must first be installed on the system in order to install the software for RAT. For IRIX 6.2 systems and above, the subsystem **eoe.sw.uucp** must first be installed before installing RAT.

RAT depends on **/usr/bin/cu** being on the system to perform dialer operations (either through a modem or to a serial device), which can be found in **eoe.sw.uucp**.

Note: The installation of the appropriate uucp image needs to be done only on the system that will initiate the dialing. It does not need to be installed on the destination system.

If you use a modem to connect to the MSC or MMSC, the appropriate settings must be configured. For U.S. Robotics®, Telebit®, or Microcom® modems, the S0 bit must be set to 1 so that auto answering will be configured correctly. After setting the S0 register, the values can be saved using the &W modem command.

The procedure to set up the modem will be discussed in the lab project.

17-32 FOIL: RAT Command Line Options

There are a number of command line options available with RAT:

- **-d**

  Generate debugging output to **stderr**. This is handy when tracing internal problems with RAT.

- **-e <*cmd*>**

  Run specified cmd instead of **/usr/bin/cu**. When this option is used, the **-d**, **-f**, **-l**, **-n** and **-s** options are invalid. RAT will take the command line arguments after the **-e** and use them as the command to execute. Also remember that you must specify the literal path to a command, such as **/usr/bsd/telnet** instead of **telnet**.

- **-f <*file*>**

  Save all data sent and received from the remote system controller into the specified file.

- **-l <*line*>**

  Specify the line to communicate with. This is the same device as the one specified with **/usr/bin/cu**.

- **-n <*telno*>**

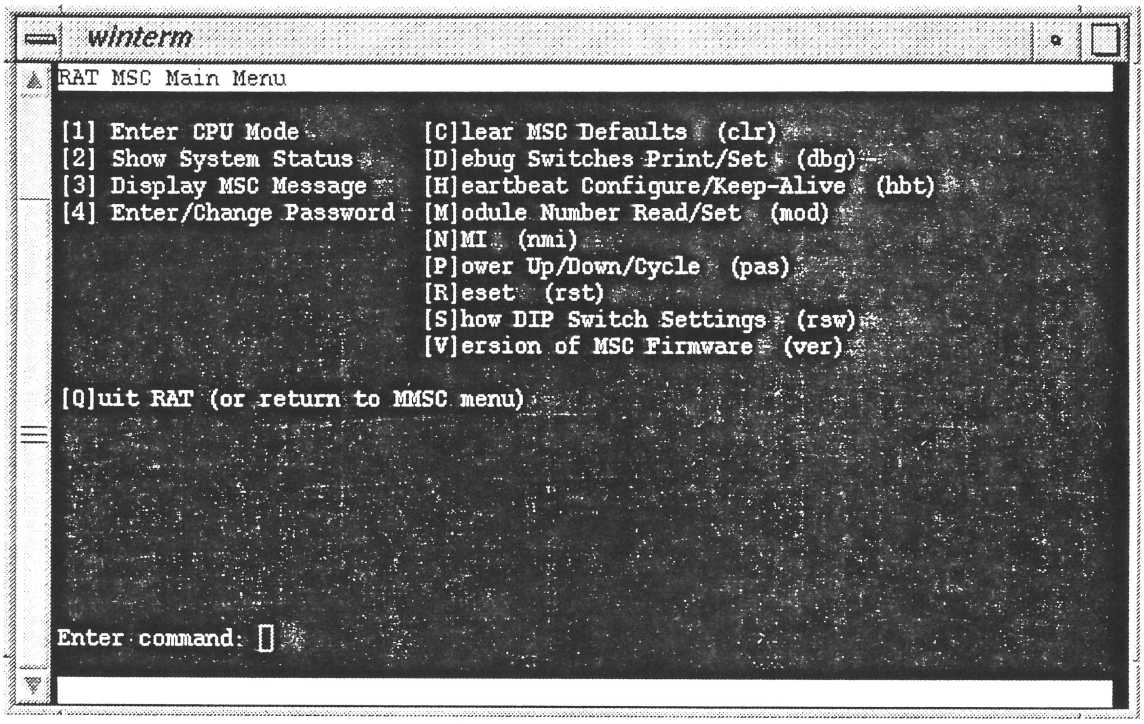
  Specify the telephone number to dial as **telno**. This is similar to the phone number specified with **/usr/bin/cu**, except that we require a **-n** option to the phone number.

- **-s <*speed*>**

  Specify the speed at which to try and connect. This passes a speed parameter to **/usr/bin/cu**, and assuming that the speed specified is valid according to **/etc/uucp/Devices**, the dialer will connect at the rate specified.

17-33 FOIL: RAT Display (MSC Mode)

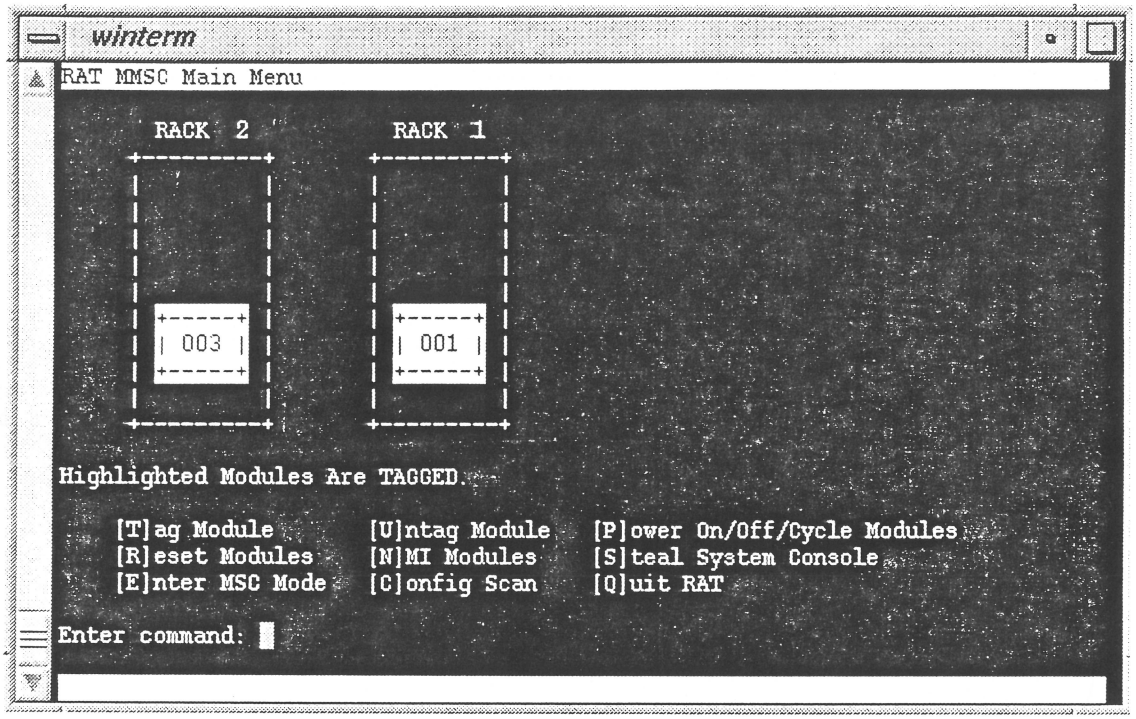
When you initialize RAT on a deskside system, it displays the following menu:



You can also display this screen by selecting the **[E]nter MSC Mode** option from the MMSC main menu.

17-34 FOIL: RAT Display (MMSC Mode)

When you initialize RAT on a rack system, it displays the following menu:



RAT displays all of the tagged (numbered) modules in the system. From this menu, you can perform several MMSC functions.

Note: RAT displays the tagged modules in the system as viewed from the rear of the system cabinets.

17-35 FOIL: RAT URLs

RAT Users Guide

http://ist.csd.sgi.com/Tools/Home_pages/tools_info.html

RAT General Information

http://bits.csd.sgi.com/digest/topics/Lego/status/RandD/RAT.html

17-36 FOIL: FRU Analyzer

The FRU analyzer is a PROM-based, IRIX-resident tool that provides the SSE with a starting point in the resolution of a system failure.

When the operating system determines that a fatal error has occurred, the hardware error state is collected and passed to the FRU analyzer for interpretation. The FRU analyzer applies a set of rules to the hardware state and from those rules it reaches a conclusion as to the cause of the fatal error. This conclusion can be that a specific piece of hardware has failed, that software caused the crash, or that it can reach no conclusions from the state it was given.

It generally will report that a FRU (Field Replaceable Unit) has failed, but in certain circumstances the error can be traced to a more specific part of a FRU and, in those cases, both the FRU and the detailed conclusion will be printed in the analysis summary.

The FRU Analyzer is best at tracking down straightforward errors, such as, failing memory DIMMs. It does not work very well at tracking failing CrayLink connections.

FRU requires no operator intervention; it is called automatically by the operating system.

Further Information

Refer to ***FRU Analyzer Reference Guide***, SGI part number 108-0166-002, and ***ICRASH Reference Guide***, SGI part number 108-0167-001 for more information about the FRU analyzer.

This document can be downloaded from the following URL:

http://servinfo.cray.com/origin/index.htm

17-37 FOIL: FRU Analyzer Output

The outputs from the FRU analyzer are located in the directory

/var/adm/crash

Two types of files are generated:

- **analysis.#**
- **fru.#**

analysis files contain a summary of information gathered from the **icrash** and **kpm** utilities after core dumps. They point to other files generated by the core dumps and record the panic strings generated by the crash.

fru files contain the analysis report generated by the FRU analyzer program.

Some examples of fru analysis files follow:

Example 1

In this example, the conclusion was a software error.

fru.1

```
++FRU ANALYSIS BEGIN
++
++
++                FRU Analysis Summary
++
++    Software                        : 90%
++
++FRU ANALYSIS END
```

17-38 FOIL: FRU Analyzer Output (Cont.)

Example 2

In this example, no conclusion could be reached.

fru.2

```
++FRU ANALYSIS BEGIN
++
++
++                FRU Analysis Summary
++
++          * Inconclusive hardware error state *
++
++FRU ANALYSIS END
```

Example 3

In this example the conclusion was a 40% chance that slot N3 was in error, specifically the hub chip. This also shows what the report looks like when a user initiates an NMI on the system.

fru.3

```
++FRU ANALYSIS BEGIN
++
++
++                FRU Analysis Summary
++
++    /hw/module/1/slot/n3/node/hub          : 40%
++
++FRU ANALYSIS END
```

The percentage of confidence is used to show how conclusive the error state was that this particular unit caused the failure. You can use the percent level to measure the confidence level against other FRU analyzer reports that have been generated by the same failure.

17-39 FOIL: FRU Analyzer URLs

FRU General Information

http://babylon.engr.sgi.com:80/systemsw/projects/lego/fru.html

FRU Users Guide

http://ist.csd.sgi.com/Tools/Home_pages/tools_info.html

17-40 FOIL: System Monitor (Sysmon)

You can use the System Monitor program to monitor system activity and to inform you of critical problems that may need prompt attention. It also provides detailed help messages for these errors to enable users to understand the problem and take appropriate actions.

SYSMON uses the **/var/adm/SYSLOG** file for most of its operations. Various system processes log messages in **SYSLOG** for error reporting and logging other important activities. SYSMON notifies the users of some of these errors via a pop-up notifier. Users can then click on the "help" button in the notice window for more details on the problem that caused the error. This feature is provided using the Notifier component of the SYSMON tool.

SYSMON also gives you the ability to sort and filter messages according to priority. For example, if you want to view only panic and error messages, you can configure SYSMON to filter out all messages that do not match these criteria.

The SYSMON tool will be covered in greater detail during the support tools lab project.

Further Information

Refer to the following URL or view the **sysmon(1M)** man page:

http://ist.csd.sgi.com/Tools/Tool_pages/Smon_proj.html

17-41 FOIL: Supportfolio™

Supportfolio™ Online is a collection of web-based electronic support tools designed to help Silicon Graphics system administrators more effectively manage and troubleshoot their SGI systems.

Supportfolio Online Functions

- Electronic case reporting and management
- Online patch catalog
- Online support information

Key Supportfolio Features

- Easy access from any web-enabled system
- Electronic access to software patches
- Electronic case logging and case management
- Technical Support information
- Full-featured search and information retrieval
- Personalized customer subscription service

Supportfolio Customer Benefits

- Customers do not need to wait for answers or updates. Supportfolio Online users can access patches and information 24 hrs/day, 7 days/week, and log their support calls electronically
- Electronic call logging and management allows customers to document and track their support calls

Note: Initially, Supportfolio Connect will be available for customers in the U.S. and Canada only.

- The web interface enables the customer to get the tools up and running quickly

Three pieces comprise Supportfolio: Supportfolio Search, Supportfolio Patch, and Supportfolio Connect.

17-42 FOIL: Supportfolio Search

The search feature of Supportfolio allows quick access to technical and support information. You can search by:

- **Keyword**

  Finds technical support information by searching on any word or combination of words through Keyword Search.

- **Topic Tree**

  Use the Topic Tree to find documents by concept such as Hardware, Multimedia, Networks, and so on.

- **Products**

  Find information about a particular product such as C++, Indy Presenter, MIDI, and others.

- **Sources-Support**

  Browse or limit your search by choosing specific Support information sources:

  - **Pipeline**

    A Silicon Graphics technical periodical that is published every other month

  - **QNAs**

    Common questions and answers generated by Silicon Graphics Customer Support

  - **Patches**

    Independent software modules that fix problems with products after they have released

- **Sources-6.2 Desktop**

  Browse or limit your search to the following 6.2 Desktop information sources:

  - **6.2 Technical Publications Books**

    The SGI End User, SGI Admin, and SGI Developer bookshelves for IRIX 6.2

  - **6.2 Reference (man) pages**

    UNIX pages that describe shell commands and system calls for IRIX 6.2

17-43 FOIL: Supportfolio Search (Cont.)

- **6.2 Release Notes**

  IRIX 6.2 product specific information that supplements the standard product release documentation

- **Sources-Web Sites**

  Browse or limit your search to the available Web sites:

  - **Silicon Surf**

    Silicon Graphics Web server

  - **MIPs Inc.**

    MIPs Inc. Web server

Documents that can be searched or browsed are drawn from the following sources:

- Technical Publication "IRIS InSight™" Books (IRIX 6.2)
- Manual (man) pages (IRIX 6.2)
- IRIX release notes (IRIX 6.2)
- Patches1 - All OS's
- Pipeline2 - The bimonthly support newsletter
- QNAs2 - Common Questions and Answers
- Web - Silicon Surf and the SGI MIPs web server

17-44 FOIL: Supportfolio Patch

Supportfolio Patch allows you to locate, download, and install the latest patchset and patches.

Several features are available with Supportfolio Patch:

Patch Browse

Find a specific patch by sorting patches any of the following ways:

- **Release Date or Patch Number**

  If you know the release date or the number of the patch you are looking for, quickly browse this list to find your patch.

- **Category**

  Search for the patch you need by category such as Graphics, Multimedia, SCSI, and more.

- **All Patches**

  Browse the list of all patches available; this list contains all patches for all IRIX operating systems.

- **Replaced Patches**

  Patches are often replaced by more up-to-date patches; this list shows every patch ever replaced and links you to the patch that currently replaces it

- **Patch Search**

  Customize your search for patches. Patch Search allows you to select several parameters before searching; for example, select an IRIX version, category, release date range and enter a keyword. There are many combinations to choose from.

- **Patch News**

  Read the latest news about patches here.

- **Security**

  Visit the Silicon Graphics Security Headquarters site. You will find information and resources for computer security including free security patches.

17-45 FOIL: Supportfolio Connect

Supportfolio Connect allows the customer to report, document, and track technical support cases.

Supportfolio Online is used to view and track:

- **Active Cases**

  Displays a list of support cases. The customer can read or update any case in this list. Newly submitted cases are automatically added to the Active Cases list.

- **New Case**

  Opens a new support case with the Silicon Graphics Technical Assistance Center (TAC).

- **Case Search**

  Searches and views both the open and closed cases that have been logged since January 1994. The customer can also add any of these cases back to the Active Cases list from here.

- **Update Users**

  Grants co-workers who have access to Supportfolio Connect permission to view and update the customer's cases by adding or deleting their names from this list.

Supportfolio URL

The URL for the Supportfolio package is:

http://support.sgi.com/

17-46 FOIL: Micro-Diagnostic Kernel (MDK)

The micro-diagnostic kernel (MDK) is a standalone diagnostic environment that runs in kernel mode on Origin200, Origin2000, CRAY Origin2000, and Onyx2 systems. The MDK lets you load and run diagnostic tests in an environment that provides access to hardware components that cannot be accessed from a user program that is running under the IRIX operating system.

MDK-based diagnostics require full use of the system: you must bring down (reboot) the system to boot the MDK. Therefore, no other kernel (that is, the IRIX operating system) can be running when you use the MDK-based diagnostics. You should use the MDK-based diagnostics to isolate failures for which the IRIX based diagnostics do not provide enough error information or to test hardware that the IRIX based diagnostics cannot test.

The MDK provides features that enable MDK-based diagnostic tests to operate like normal user programs; these features include memory allocation, console output, exception handling, and multiprocessing. The MDK also provides system configuration information (memory size and number of CPUs, Nodes, and Routers) for the diagnostic tests.

MDK Test Packages

The MDK comprises 3 test packages:

- The **t5r.mdk** test package contains the micro-diagnostic kernel, the R10000 processor tests, and the Router SSO test.
- The **memory.mdk** test package contains the micro-diagnostic kernel, the secondary cache test, and several memory tests.
- The **stress.mdk** package contains the micro-diagnostic kernel and the cache thrasher with I/O test.

Further Information

Refer to *MDK-Based Field Diagnostics*, SGI part number 108-0162-001, for more information about MDK.

This document can be downloaded from the following URL:

http://servinfo.cray.com

17-47 FOIL: MDK Tests

Two types of diagnostic tests are available in the MDK:

- **Directed tests**
- **Stress tests.**

Directed Tests

Directed tests focus on one area of the system.

- **R10000 processor tests**

  The R10000 processor tests are directed tests that focus on the components within each R10000 processor. These tests check the following areas:

  - the functionality of the external interface to each R10000 processor
  - the arithmetic logic unit (ALU) in each R10000 processor
  - the floating-point unit (FPU) in each R10000 processor
  - the primary cache in each R10000 processor
  - the instruction pipeline in each R10000 processor
  - the register-renaming functionality of each R10000 processor

  If an R10000 processor test fails, the failing hardware is the processor in which the test detected the errors. The failing field replaceable unit (FRU) is the Node board that contains the failing R10000 processor.

  To run the R10000 processor tests, you must load the **t5r.mdk** test package.

- **Secondary cache test**

  The secondary cache test verifies the functionality of the secondary cache located on each Node board.

  If this test fails, the failing hardware component is the secondary cache or memory. The failing FRU is the Node board that contains the failing secondary cache or the failing memory dual inline memory module (DIMM).

  To run the secondary cache test, you must load the **memory.mdk** test.

17-48 FOIL: MDK Tests (Cont.)

- **Memory tests**

  The memory tests verify that memory actually contains data that was written to it. The memory tests range from testing memory with 1 or 2 CPUs to testing memory with all Nodes and CPUs in the system.

  If a memory test fails, the failing hardware is one or more memory DIMMs, the secondary cache (HIMM), a Node board, a midplane, or a Router board. The failing FRU is either a DIMM, Node board, midplane, or Router board.

  To run the memory tests, you must load the **memory.mdk** test package.

- **Router SSO test**

  The Router SSO test checks the Router links.

  If this test detects a Router failure, the failing hardware is a compression (CPOP) connector, a terminator, or a Router chip. The failing FRU is a Node board, midplane, Router board, or connecting cable. If this test detects a HUB failure, the failing FRU is a Node board, Router board, or midplane.

Note: Before you replace any FRUs, reseat the suspected boards and reconnect the suspected cables, and then run the test again. If the test still fails, you will need to replace one or more of the FRUs.

To run the Router SSO, you must load the **t5r.mdk** test package.

Stress Tests

The stress tests use many hardware components to test a larger portion of the system. There is one stress test available for use in the field: the cache thrasher with the I/O.

- **Cache Thrasher With I/O Test**

  The cache thrasher with I/O test stress tests the CrayLink network and the HUB-to-XBOW links. It also stress tests cache coherency from the processor and from I/O.

  If this test fails, check the Router board LEDs and Crossbow LEDs on the midplane for lost links.

  The cache thrasher with I/O test is part of the stress.mdk test package; you must load the **stress.mdk** package to run the cache thrasher with I/O test.

17-49 FOIL: Running MDK tests

You must boot the MDK from the PROM monitor prompt. Therefore, the system must be shut down during testing. After the test has completed, you must reset the system to exit the MDK kernel and return to the PROM monitor. The example below demonstrates how to load and run the **t5-1** diagnostic:

1. Use the boot command from the PROM monitor (at the BaseIO command prompt, **>>**) to start the t5r.mdk test package:

   >>boot dksc(0,1,0)/stand/t5r.mdk

   MDK displays boot status information and the **MDK>** prompt.

2. Load the **t5-1** test from the `MDK>` prompt:

   ```
   MDK> t5-1
   ```

   The **t5-1** test loads into the system and begins testing the R10000 processors.

3. Perform a soft reset of the system to exit the MDK and return to the PROM monitor prompt:

   ```
   MDK> reset
   ```

17-50 FOIL: MDK Test Output

All MDK-based diagnostics return output to the BaseIO console. The diagnostics print pass and fail messages as they run; all MDK-based diagnostics print a *testname* `test has completed` message when they complete testing.

Output From Successful Completion

When an MDK-based diagnostic completes successfully, the diagnostic displays one or more messages that indicate that it completed without errors.

For example, the Router SSO test returns the following messages when it completes testing without detecting any hardware failures:

```
TEST RESULT: **** TEST PASSED  ****
MDK Router_sso test run is complete
```

Output When a Test Detects a Failure

When an MDK-based diagnostic detects a hardware failure, it prints a failure message and any information related to the failure.

For example, the `ERROR: Test Failed` message in the following output is from a test that detected a hardware failure:

```
NODE0 CPU1 Pid 2 FPU3 Test Started
NODE0 CPU0 Pid 1 FPU3 Test Started
NODE0 CPU1 Pid 2 FPU3 Test Passed
NODE0 CPU0 Pid 1 ERROR: Test Failed
```

17-51 FOIL: MDK Test Output (Cont.)

Output When an Unexpected Exception Occurs

When an unexpected exception occurs, MDK prints out the contents of the Coprocessor 0 registers, the general-purpose registers, and the following information about the exception: the type of exception, the address where the exception occurred, and any other information that is relevant to the exception.

After MDK prints out this information, the `MDK>` prompt appears. At this point, you can load another diagnostic test, reset the system, or issue a non-maskable interrupt (NMI).

The following example shows MDK output from an unexpected exception:

```
ERROR: Unexpected Exception Occured.
 Nasid 1: Local CPU A: Global CPU 2: PID 3: Multiple exceptions at
0xa800000000a0c4e8: sr = 0xb4007fe6: cause = 0x881c
 Nanos: Frozen 22A 001:
 Status: 0xffffffffb4007fe6        XCtxt: 0xffffffe000002a30
 Hi:     0x                0       Lo:    0x                0
 epc:    0xa800000000a0c4e8        cause: 0x             881c
 count:  0x         18afb669       comp:  0xffffffffffffffff
 Enhi:   0x                0       CacErr: 0xffffffffd442aa80
 R01/AT: 0x            1ff00       R02/V0: 0xffffffffb4007fe6
 R03/V1: 0x           a257b0       R04/A0: 0xa800000000a27000
 R05/A1: 0x           a1dd38       R06/A2: 0x         8bfdfdf0
 R07/A3: 0x                8       R08/A4: 0x                4
 R09/A5: 0xa800000183fdf978        R10/A6: 0x         3e17aad8
 R11/A7: 0xa800000300caaad8        R12/T0: 0x               38
 R13/T1: 0x                0       R14/T2: 0x                0
 R15/T3: 0xa800000300caaa98        R16/S0: 0xa800000183fdf978
 R17/S1: 0xa8000000005fbdc8        R18/S2: 0x                0
 R19/S3: 0x                0       R20/S4: 0x                2
 R21/S5: 0xa800000008000000        R22/S6: 0x                3
 R23/S7: 0x               e0       R24/T8: 0x          8000000
 R25/T9: 0xa800000000a07ed8        R28/GP: 0x                0
 R29/SP: 0xa800000183fdd928        R30/S8: 0xa800000000a27000
 R31/RA: 0xa800000000a08eac
 Nanos: Frozen 3
```

17-52 FOIL: MDK Debugging Tips

If a hang or unexpected exception occurs while you are running MDK-based diagnostics, use the following POD mode commands to gather more information that you can use to isolate the problem:

- **nmi**, which issues a nonmaskable interrupt (NMI)
- **why**, which displays the current exception and NMI status information
- **error**, which displays error information from the MD section of the HUB
- **error_dump**, which displays detailed information about all error bits in the HUB
- **dumpspool**, which dumps a CPU's PI error spool
- **pr**, which prints the contents of an R10000 or HUB register
- **crb**, which dumps the I/O interface (II) CRBs
- **crbx**, which dumps the II CRBs in a 133-column-wide format

Refer to ***Origin2000 Power-On Diagnostics***, SGI part number 108-0161-001, for more information about POD mode and these commands.

17-53 FOIL: MDK Quick Reference List

The table below provides quick-reference information about the MDK-based tests. This information includes the following items:

- the functional unit that each test checks
- the name of each test
- the commands that you use to invoke each test from the PROM monitor (at the BaseIO command prompt, >>)
- the failing FRU(s) associated with each test

Functional Unit	Test	Command Line Syntax	Failing FRU
R10000 processors	*t5-1*	>>**boot dksc(0,1,0)/stand/t5r.mdk** MDK>**t5-1**	Node board
	t5-2	>>**boot dksc(0,1,0)/stand/t5r.mdk** MDK>**t5-2**	Node board
Secondary cache	*scache*	>>**boot dksc(0,1,0)/stand/memory.mdk** MDK>**scache**	Node board
Memory	*mem.qs*	>>**boot dksc(0,1,0)/stand/memory.mdk** MDK>**mem.qs**	DIMM, Node board, midplane, or Router board
	memum.qs	>>**boot dksc(0,1,0)/stand/memory.mdk** MDK>**memum.qs**	DIMM, Node board, midplane, or Router board
	memum.nb	>>**boot dksc(0,1,0)/stand/memory.mdk** MDK>**memum.nb**	DIMM, midplane, or Node board
	memum.in	>>**boot dksc(0,1,0)/stand/memory.mdk** MDK>**memum.in**	DIMM, Node board, midplane, or Router board
Memory (continued)	*mem.lo*	>>**boot dksc(0,1,0)/stand/memory.mdk** MDK>**mem.lo**	DIMM, Node board, midplane, or Router board
	memum.lo	>>**boot dksc(0,1,0)/stand/memory.mdk** MDK>**memum.lo**	DIMM, Node board, midplane, or Router board
Router links	*router*	>>**boot dksc(0,1,0)/stand/t5r.mdk** MDK>**router**	Router board, Node board, midplane, or cable
CrayLink network and HUB-to-XBOW links	*ct_io_turbo*	>>**boot dksc(0,1,0)/stand/stress.mdk** MDK>**ct_io_turbo**	Router board, Node board, midplane, or cable

17-54 FOIL: IRIX-Based Field Diagnostics

The IRIX based field diagnostics are diagnostic tests that you can run from an IRIX prompt while user operations continue on Origin200, Origin2000, CRAY Origin2000, and Onyx2 systems. There are two components to the IRIX based field diagnostics:

- **field_diag program**

  The **field_diag** program is an ASCII user interface that you can use to run any of the IRIX based field diagnostics that are available for the hardware in your system.

Note: This interface is identical to the user interface for the field stress tool (FST). The similarity of the two interfaces provides a common "look and feel" for diagnosing hardware failures.

- **Various diagnostic tests**

  Diagnostic tests are available to test the following hardware components while IRIX is running on a system:

  - Ethernet hardware on the BaseIO boards
  - floating-point unit on the Node boards
  - LINC DMA hardware on the Node boards, ATM boards, and HIPPI-S boards
  - MediaIO boards
  - memory
  - RAID array hardware (restricted to inquiry to the customer-replaceable unit level)
  - networking hardware (ATM board, BaseIO board, HIPPI-S board, MENET board)

Note: The System Verification Program (SVP) must be installed to run the field diagnostics because the field diagnostics package uses several of the diagnostics contained in SVP.

Further Information

Refer to *IRIX-Based Field Diagnostics*, SGI part number 108-0163-001, for more information.

This document can be downloaded from the following URL:

http://servinfo.cray.com/origin/index.htm

17-55 FOIL: 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. This interface enables you to perform the following actions:

- select any available diagnostic test
- modify any user-adjustable parameters for the selected diagnostic test
- run the selected diagnostic the to test the system
- view the output from the selected diagnostic test

Starting the field_diag User Interface

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:

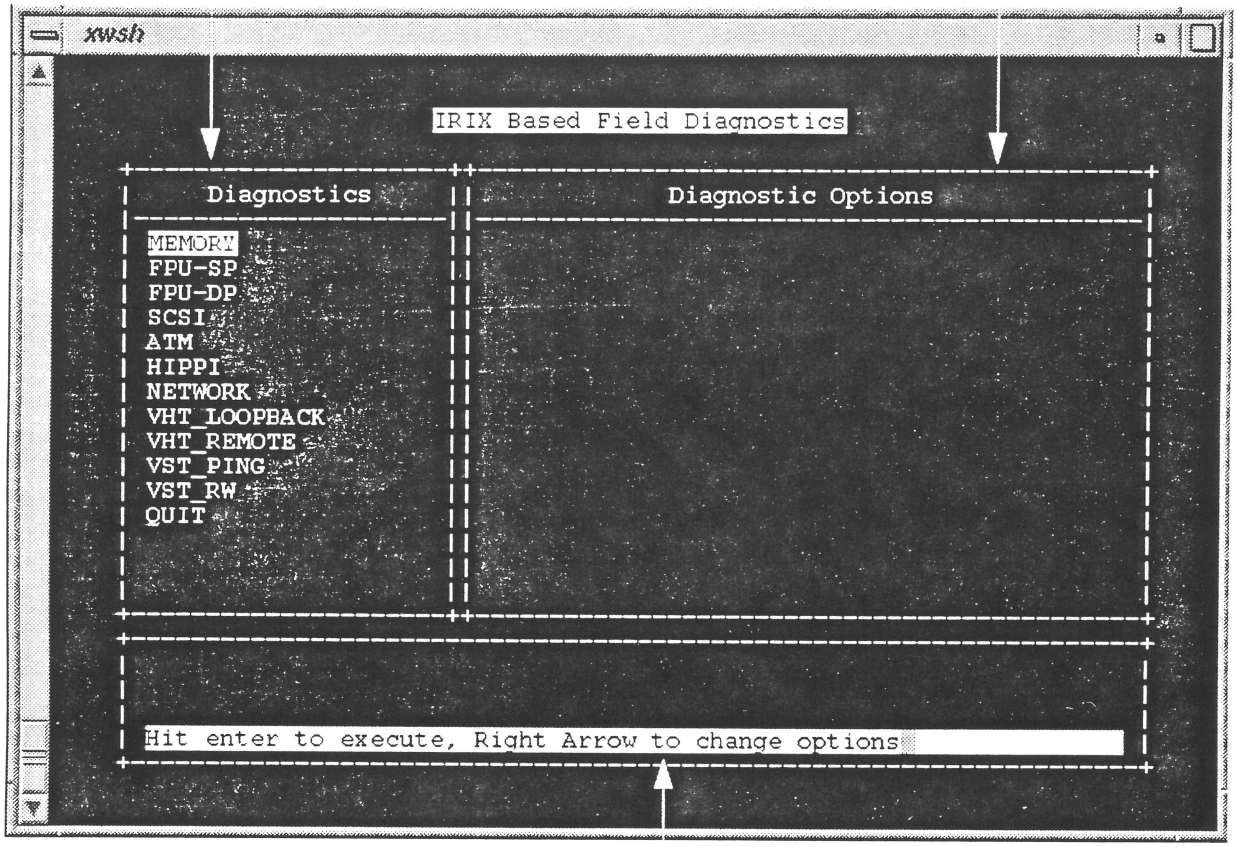
   ```
   cd /usr/diags/diags_interface
   ```

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

   ```
   ./field_diag
   ```

17-56 FOIL: field_diag Main Menu

The illustration below shows the components of the main screen of the **field_diag** user interface:



The main screen includes the following components:

- Diagnostic list
- Diagnostic Options list
- diagnostic status area

When you type **Ctrl+T** on the main screen, the Test Log Output window appears. This window displays the output messages from the test that is currently running.

17-57 FOIL: field_diag Diagnostic List

The Diagnostic list contains the names of all diagnostics that are available for the system configuration that you are using. The **field_diag** program automatically determines the hardware available in the system being tested and adjusts this list to display only the diagnostics that can test the hardware in the system.

Option	Description
ATM	Selects the ATM LINC DMA verification test. Refer to Section 3.1, "ATM LINC DMA Verification Test," for more information.
FPU-SP	Selects the floating-point unit (single-precision) test. Refer to Section 3.2, "Floating-Point Unit (Single-Precision) Test," for more information.
FPU-DP	Selects the floating-point unit (double-precision) test. Refer to Section 3.3, "Floating-Point Unit (Double-Precision) Test," for more information.
HIPPI	Selects the HIPPI-S LINC DMA verification test. Refer to Section 3.4, "HIPPI-S LINC DMA Verification Test," for more information.
MIO	Selects the MediaIO board test. Refer to Section 3.5, "MediaIO Board Test," for more information.
MEMORY	Selects the memory test. Refer to Section 3.6, "Memory Test," for more information.
NETWORK	Selects the network thrasher test. Refer to Section 3.7, "Network Thrasher Test," for more information.
RAID	Selects the RAID verification test. Refer to Section 3.8, "RAID Verification Test," for more information.
SCSI	Selects the SCSI thrasher test. Refer to Section 3.9, "SCSI Thrasher Test," for more information.
VHT_LOOPBACK	Selects the vicious HIPPI test (loopback test). Refer to Section 3.10, "Vicious HIPPI Tests," for more information.
VHT_REMOTE	Selects the vicious HIPPI test (remote test). Refer to Section 3.10, "Vicious HIPPI Tests," for more information.
VST_PING	Selects the vicious socket test (ping test). Refer to Section 3.11, "Vicious Socket Tests," for more information.
VST_RW	Selects the vicious socket test (read and write test). Refer to Section 3.11, "Vicious Socket Tests," for more information.
QUIT	Quits the field_diag program.

17-58 FOIL: Field Stress Tool (FST)

FST is an application that simulates varying customer application loads on Origin200™, Origin2000™, CRAY® Origin2000, Onyx2™, CHALLENGE®, and Onyx® computer systems to determine the health of these systems. You can use it for preventative maintenance to verify that a system will continue to perform without failures.

FST has two main components:

- **ASCII user interface**

  This common ASCII interface enables you to start the FST stress tests and monitor their progress.

  Note: This interface is identical to the user interface for the IRIX based field diagnostics. The similarity of the two interfaces provides a common "look and feel" for diagnosing hardware failures.

- **Stress test packages**

  FST supports 2 test packages. You can run these packages from the ASCII user interface or from the IRIX command line:

  - **Application stress test**

    The Application stress test includes application programs that stress the system with varying levels of activity.

  - **Pandora stress test**

    The Pandora stress test includes a programmable set of processes that test specific hardware. Pandora configures itself before starting any test processes and tests the following hardware components:

    - 1 or more CPUs
    - 32 MB to 4GB of memory
    - I/O hardware (SCSI and MSCSI devices)
    - networking hardware (Ethernet, FDDI, and ATM interfaces)
    - multiple-pipe graphics subsystem

      This test creates five types of test processes: floating-point processes, memory processes, I/O processes, network processes, and graphics processes. These processes generate activity throughout the whole system by using shared resources.

17-59 FOIL: Field Stress Tool (Cont.)

- **Stress test options**

  FST supports 5 different test options. Each option is described below in greater detail:

  - **PANDORA option**

    Runs the Pandora test suite for 30 minutes.

  - **DEFAULT_STRESS_TEST option**

    Runs a predetermined suite of stress tests for 15 minutes. This option is part of the Application test package.

  - **GFX_STRESS_TEST option**

    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 option**

    Runs a nongraphics stress test. This option is part of the Application test package.

  - **CLEAN option**

    Deletes any processes that were left running by a previous stress test. Use the CLEAN option after a test completes testing to clean up any processes that the field stress tool created for the test. This option is part of the Application test package.

Further Information

Refer to ***Field Stress Tool Reference Guide,*** SGI part number 108-0152-001, for more information.

This document can be downloaded from the following URL:

http://servinfo.cray.com/origin/index.htm

17-60 FOIL: FST User Interface

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. This interface is identical to the user interface for the IRIX based field diagnostics. The similarity of the two interfaces provides a common "look and feel" for diagnosing hardware failures.

Starting the FST User Interface

You must have PERL installed on the system to use FST, and have root privileges to run the FST user interface.

Perform the following procedure from the IRIX prompt to start the FST user interface:

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

17-61 FOIL: FST User Interface Main Screen

The FST main screen includes the following components:

- **Stress Test list**

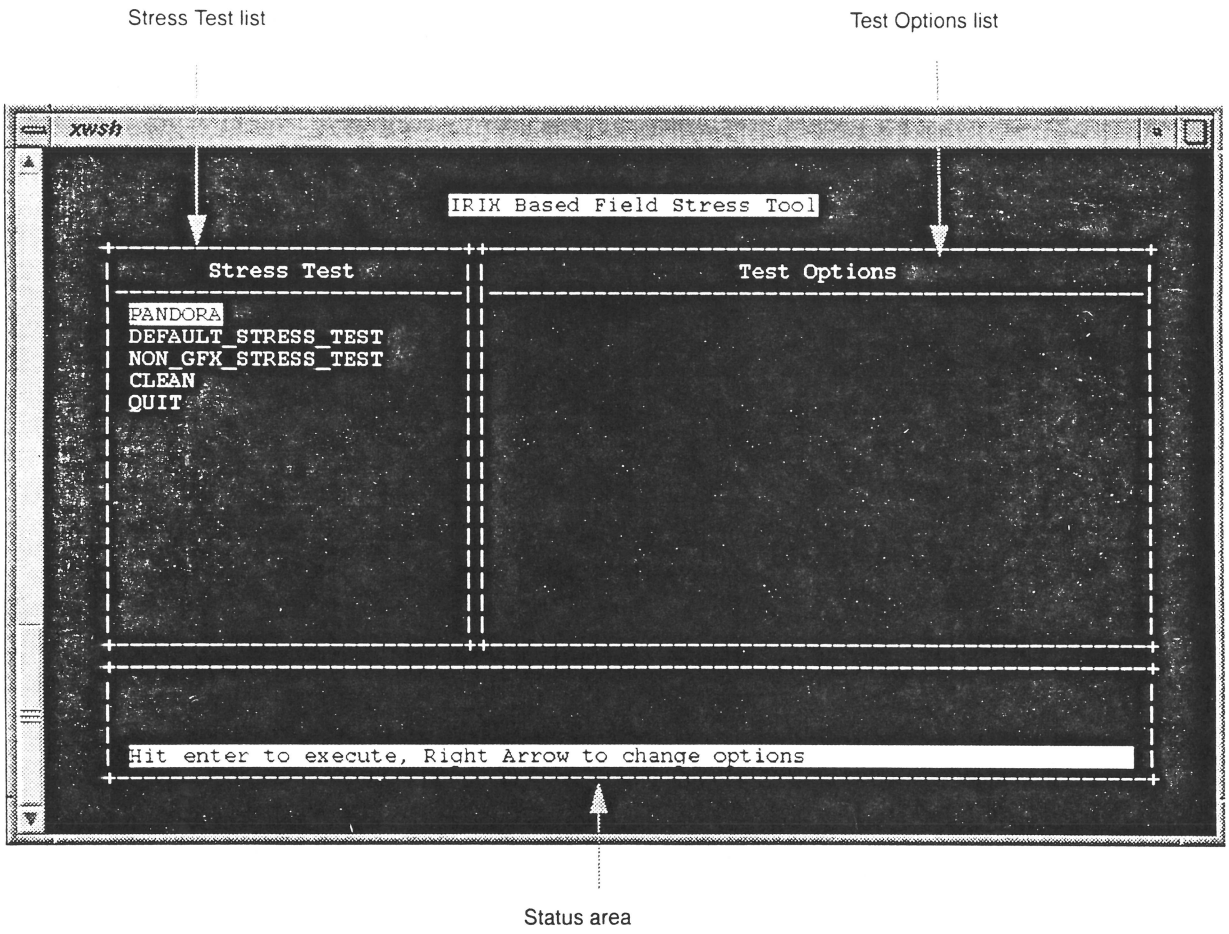
  The Stress Test list contains the names of the tests that you can run from the FST user interface. FST automatically detects the hardware in your system and adjusts this list to display only the stress tests that are available for the available hardware.

- **Test Options list**

  The Test Options list shows the parameters that you can adjust for each test. If nothing appears in this list, the test does not have user-adjustable parameters.

- **Status area**

  The status area provides information about the current status of the interface and the test that is running.



17-62 FOIL: FST Test Output

You can access output from FST tests by using a common text editor to view the test output files, or you can type Ctrl+T on the main screen of the ASCII user interface to view the output messages from the test that just completed.

Test Output Files

- **Application Stress Tests**

  All results from the Application stress test are logged in the **/usr/diags/stress/RESULTS/** directory structure. The Application stress test generates the following output files:

  - **/usr/diags/stress/RESULTS/SYSLOG_ALL**

    This file contains all relevant messages from **SYSLOG**; these messages are copied into the **SYSLOG_ALL** file after the test completes running.

  - **/usr/diags/stress/RESULTS/TEST_MSG_LOG**

    This file 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.

  - **/usr/diags/stress/RESULTS/*testcase*.LOG.*timestamp***

    This file 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. The ***timestamp*** portion contains the date and time at which the test was run.

    (For example, **/usr/diags/stress/RESULTS/APPLICATIONS/ONE_NG_2H.LOG.Mon-Jul-28-06:41:40-1997**)

  - **/usr/diags/stress/RESULTS/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.

17-63 FOIL: FST Test Output (Cont.)

- PANDORA Stress Tests

  All results from the PANDORA stress test are logged in the **/usr/diags/stress/RESULTS/** directory structure. The PANDORA stress test generates the following output file:

  - **/usr/diags/stress/RESULTS/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.

Two important commands exist that you can use to monitor Hub, Router, and XBOW errors on your system:

- **linkstat**

  The **linkstat** command reports on router link performance and error rates. It also reports error rates on the Hub chip Craylink network interface (NI) and I/O Interface (II) links.

- **xbstat**

  The **xbstat** command monitors the traffic from and to each of the active ports of the Crossbow (XBOW) i/o interconnect. It reports the total number of micropackets sent from and received at each link. It also reports the LLP retries at each link.

  xbstat gets its information from the two hardware performance counters and the Link Auxiliary Status Registers in the XBOW ASIC. It uses a multiplexing technique to sample and report the SRC and DST link traffic at each XBOW port. Cumulative counts since the beginning of the monitoring period are reported in units of micropackets.

  xbstat is capable of monitoring any selected XBOW in a multi-XBOW system. The file **/dev/xbmon** should point to the selected XBOW in the hwgraph. Use **find /hw -name mon -print** to find all possible XBOW hwgraph files to which **/dev/xbmon** can point. Then create a symbolic link between the selected XBOW hwgraph file and the **/dev/xbmon** file before invoking this command.

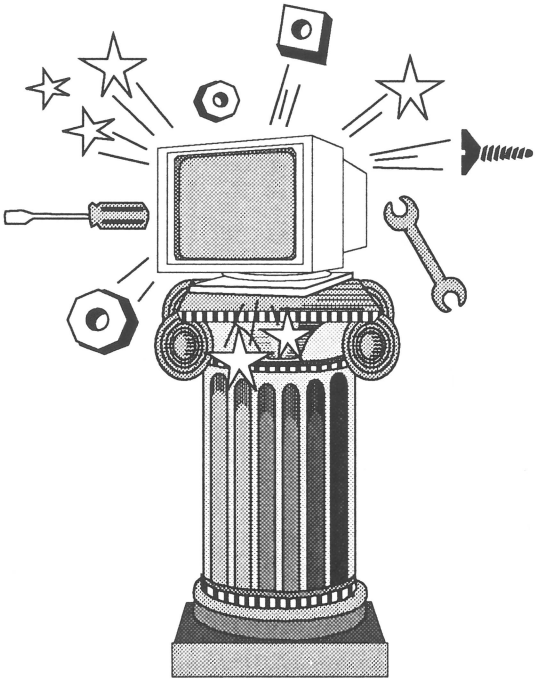
  Only one XBOW can be monitored at a time.

Further Information

Refer to the **xbstat(1)** and **linkstat(1)** man pages

SiliconGraphics
Computer Systems

Module 18: Fault Isolation: Entire System



Part Number: LBT139-1.0-MR-6.4-S-SD-W
August 1997

18-2 FOIL: Module Objectives

The objective of this module allows each student time to apply the knowledge learned during the course, and reinforce that learning by troubleshooting instructor-induced faults within the complete system.

- Each student must demonstrate that a majority of the instructor-induced faults can be identified using any available documentation or tool.
- Each student must demonstrate that diagnostics and special troubleshooting procedures can be used to locate induced failures.

Student Notes

18-3 FOIL: Fault Isolation: Entire System

- Student teams troubleshoot each fault together, and it is recommended that only **one team member make the keyboard entries.**
- Each lab system has a different instructor-induced fault, and as the team begins to evaluate a new fault, a **different team member** should be selected to **make the keyboard entries.**
- The team must leave a system in the **same** condition that it was in when they took control of the system.
- Time allotted to complete this lab: **4 hours.**
- Logic boards can be backed out, but **do not remove** or **replace** any components from the system. If a team wants to determine what would happen if a board or component is replaced, ask the lab instructor how the replacement would effect the system: since lab time is important, this allows the troubleshooting evaluation to proceed at a faster pace, and it also reduces the risk of another fault being accidentally introduced into the system.

Index: Student Workbook

Numbers

10Base-T Ethernet 9-33
2-RM pipe 13-21
24 inch monitor 2-29
2D 12-9
2D Drawings 12-27
2D Exercise 12-29
3-Dimensional Space 12-31
3D 12-9
4-RM pipe 13-21

A

AC OK 9-17
ADAT 13-37
air temperature 9-7
aliased 12-19
anti-aliasing 13-69
ASCII Terminal Attributes 9-43
ASICs 3-23
attribute descriptions 13-121
authority levels 9-33
authority passwords 9-39
AutoLoad 8-15
Availmon 17-101

B

Base IO block diagram 7-9
Base IO board 10-25
Basic Block Diagram 13-49
Bit Plane 12-7
Block Diagra 13-49
Block Diagram 3-47
Board Illustration 13-35
Board Numbering 4-63
Boot Process 10-9, 13-85
Boot process 10-29
Bridge 3-23

C

cable length 7-11
Cac 10-49
Cache Coherence 3-59
Cache coherence 3-13
cache line 3-59
cached memory 10-49
Cached Mode 10-49
CHALLENGE vs.Origin2000 3-7
channel attribute descriptions 13-121
channel parameter descriptions 13-121
chkconfig 13-93
Clipping 12-9
clipping planes 12-35
Clogin 13-99
Clogin window 13-95
Color Code 7-11
color hue 12-13
Color Map Mode 12-23
Colors Exercise 12-17
command interpreter 10-5
Command Monitor 8-13
Command Monitor Commands 8-13
Composite Video 13-77
configuration state variables 13-93
Configuration Table 5-23
console 8-15
Console Base IO 10-25
Console Base IO algorithm 10-25
ConsoleIn 8-15
ConsoleOut 8-15
ConsolePath 8-15, 10-25
Course Outline 1-1, 2-1, 3-1, 5-1, 8-1, 9-1, 10-1, 12-1, 13-1
CPU Daughter Card 16-21
cpufreq 8-15

CRAY Origin2000 System 2-23
CRAY Router 3-43
Cray Router 4-59
CrayLink 13-19
CrayLink Considerations 7-11
CrayLink Interconnect 5-7
crossbar 3-25
Crosstown Board 13-19
Crosstown Cable 13-31
current format 13-107
Cyan 12-15

D

DAC 12-7, 12-11
Data Integrity 3-53
daughter card 16-13
dbaud 8-15
dbg 9-29
dbg command 9-27
dbgtty 8-15
Debug Switch Meaning 9-23, 9-25
Debug Switches 9-27
default format 13-107
Depth Cuing 12-9, 12-39
Deskside Slot Positions 13-15
device files 8-19
Device Naming 8-19, 8-21
Dex 10-49
DG5 13-13, 13-33
DG5 Block Diagram 13-75
DG5 Functions 13-71
DG5-2 12-23, 13-33
DG5-8 12-23, 13-33
DG5-8 Connections 13-43
DG5-DG0 13-33
DG5-GV0 13-33
DG5-GVO Connections 13-45
Diagnostics 16-47
Dirty Exclusive Mode 10-49
diskless 8-15
DISPLAY
0.0 13-99
distributed shared I/ 3-13
Distributed shared memory 3-13
Dot 12-5
Double Buffer 12-23
Download combination 13-119
downloader mode 9-39
Drive Bays 16-27
Dual Origin200 16-9
Dual Rack Configuration 13-25, 13-27, 13-29
Dual-module Installation 16-37
Dynamic Framebuffer Resizing 13-73, 13-79

E

Early Exception Codes 10-23
early warning interrupt 9-7
Edge Interpolation 13-63
Edit attributes 13-119
Edit attributes window 13-121
Edit globals window 13-113
EEPROM 13-107, 13-109
Eight-node System 5-17
ELSC 9-5
Entering POD 10-51
entry-level server 2-5
env 13-99
error checking 3-53
Error Codes 10-23
error files 13-97
/etc/inittab 13-87, 13-89
/etc/rc2 13-91
/etc/rc2.d/S98xdm 13-91
Ethernet controller 16-15
Ethernet hub 9-37
exception 10-49
Express Links 7-17

F

failsafe 13-101
fan functionality 9-7
FFSC 9-5
Fill the Triangle 13-65
Firmware Flow Overview 10-7
flash prom 10-5
Flashcpu 8-13
flashing code 10-21
Flashio 8-13

Floating Point Numbers 12-5
Force clear passwords 9-39
Four-node System 5-15
Frame 12-7
Frame Buffer 13-59
frame buffer 13-7, 13-9, 13-11
FrameBuffer 12-7

G

gamma 13-99
gConsoleIn 8-15
gConsoleOut 8-15
GE14 13-7, 13-13, 13-33, 13-53
GE14 Functional Block Diagram 13-55
GE14 Functions 13-53
Genlock 13-107
Geometric Figures 12-5
Geometry Engine 12-5, 12-31
Geometry Engines 13-7, 13-9, 13-11, 13-33
GFX Components 13-13
GFX Monitor Connections 13-41
GFX Terminology 12-5
gfxinfo 13-83
gfxinit 13-83, 13-99
Global Master 7-21, 8-7, 10-5, 10-7, 10-21, 10-25, 10-27
Global Master algorithm 10-27
Global POD 9-23, 10-51

H

Hardware Debug Switch 9-9
Hardware Debug Switches 9-21, 9-27, 9-29, 10-51
Hardware Graph 8-21
hardware graph 7-27
Hardware Inventory 7-23
Hidden Surface Removal 12-9, 12-39
hidden surface removal 13-69
HIMMs 3-31
hinv 7-23, 13-83
hinv -mvvv 7-23
Host Processor 12-5
HUB 3-23
HUB ASIC 4-17

I

image buffer 12-39
Image Buffers 12-23
Image Memory 12-7
Image Memory Processor 13-69
IMP 13-69
InfiniteReality 2-25, 13-33
init 13-89
init process 13-87
initdefault 13-89
inittab 13-87
Integers 12-5
Interconnect Fabric 5-9
interconnection fabric 2-7
IO6 Microcode 8-9
IO6 PROM 8-5, 10-5
IO6G 13-13, 13-37, 13-47
IO6G Connections 13-39
IOC3 3-23
IP27 Microcode 8-7
IP27 PROM 8-5
IP27 Prom 10-5
Ircombine 13-77
ircombine 13-83, 13-109
ircombine window 13-111
IRIX boot path 10-5
Irsaudit 13-105
irsaudit 13-83, 13-103

J

JTAG 13-103
Jtag Bus 13-49

K

KCAR 13-5, 13-11, 13-21
KEGO 13-5
Key ASICs 3-23
KONA 13-5
KonaTown 13-21
Ktown 13-13, 13-21, 13-31

L

LED 7-13
LED Error Codes 10-23
LEDs 9-19, 16-43, 16-49
Lighting 12-9, 12-43
Lighting Exercise 12-45
Lighting Model 12-43
linkstat command 7-27
Local POD 9-23, 10-51
Logic Carrier 16-13
.login 13-101
login window 13-99

M

Magenta 12-15
Main System Board 16-15
Main system board 16-13
Matrix Math 12-5
Memory 4-29
Memory DIMMs 4-35
Memory Layout 4-31, 16-19
Memory Organization 3-57
Memoryless POD 9-23, 10-51
meshed triangles 13-9
Midplane 3-35, 4-61
midplane laser number 10-25
MMSC 9-5
MMSC Cable Configuration 9-35
MMSC Display Panel 9-39
MMSC Features 9-33
MMSC secret functions 9-39
MMSC special functions 9-39
MMSC-to MMSC 9-37
mod command 7-7
modem cable 9-43
Module Block Diagram 7-9
Module Numbers 7-7
Monitor Connection 13-41
motif 13-101
Motion Exercise 12-25
MSC 7-7, 9-5, 16-25, 16-45
MSC DIP Switch Settings 16-39
MSC Features 9-7
MSC Front Panel 9-9
MSC Functions 9-7
MSC Interconnect Diagram 9-13
MSC Messages 9-11
Multi-seat Onyx2 13-47
Multiple Output Channels 13-81
mvmodule command 7-7

N

NASID 7-19, 10-53
netaddr 8-15
Network Interface (NI) 4-21
NIC 3-27
NMI 10-49, 10-51
Node Board 3-31, 4-11
Node Board LEDs 10-21
noiconlogin 13-93, 13-95, 13-99
Non-Maskable Interrupt 10-51
nonstop 8-15
Null Router 3-43, 4-47
null-modem cable 9-43
NUMA 7-19, 10-53
Number In a Can 3-27

O

Object Space 12-5
Onyx2 2-5
Onyx2 Architecture 13-51
Onyx2 Board Illustrations 13-35
Onyx2 Board Set 13-33
Onyx2 Deskside 2-25
Onyx2 Graphics Systems 2-11
Onyx2 InfiniteReality 13-5, 13-11
Onyx2 Products 13-5
Onyx2 rack 13-11
Onyx2 Rack Systems 2-29
Onyx2 Reality 13-5, 13-7
Origin Servers 2-11
Origin Vault 2-33
Origin200 2-13, 2-15, 2-17, 16-7
Origin200 Profiles 16-9
Origin2000 2-5, 2-9
Origin2000 Deskside 2-19
Origin2000 Rack 2-21

Orthographic 12-31
Orthographic Space 12-33
OSLoadPartition 8-15

P

page 3-59
Page migration 3-13
pcfg 7-19, 7-25
pcfg command 7-25
PCI Backplane 16-23
PENB 9-17
Perspective 12-31
Perspective Space 12-35
PG 13-65
Physical Debug Switches 9-27
physical debug switches 9-29
pipe 13-5
Pixel 12-7, 12-13
Pixel Bus 13-49, 13-69
Pixel Generator 13-59
pixel transfer rate 13-71, 13-81
Pixel vs. FrameBuffer 12-21
POD 8-7, 8-13, 10-5
POD Command Interpreter 10-49
POD Prompt 10-53
POKB 9-17
Polygon 12-5
Power and Cooling 4-65
Power Supply 16-31
Power-On Boot Analysis 10-29
Power-On process 10-9
Power-on Sequence 9-15
Primary Colors 12-11
primary instruction 3-31
printenv 8-17
ProbeAllScsi 8-15
Processor Interface (PI) 4-19
.profile 13-101
PROM environmentals 8-15
PROM monitor menu 8-11
PROM variable descriptions 8-15
PS_EN 9-17

R

R Bus 13-49
R10000 2-7
Rack Numbers 7-5
Rack Slot Positions 13-17
rackid 7-5
Rackmount Installation 16-41
Rackmount Origin200 16-11
rbaud 8-15
Reality 13-33
RGB Mode 12-23
RM Functions 13-59
RM7 13-9, 13-13, 13-33
RM7 Block Diagram 13-61
RM8 13-7, 13-13, 13-33
RM/TM 13-33
root 8-15
route 7-19
Route Table 7-19
route tables 8-7
Router 3-23, 3-41
Router Board Types 4-45
Router LEDs 7-13
router status 7-27
rsw 9-29
run level 2 13-89

S

S2MP 3-29
S98xdm 13-91
SASH 8-5
Save As 13-119
Save to EEPROM 13-119
Screen Coordinates 12-19
Screen Pixel 12-13
secondary cache 3-31
Secondary Colors 12-15
Select a video format 13-117
Serial 8-13
setmon 13-109
sgilogo 8-15
.sgisession 13-101
Single Buffer 12-23
Single Rack Configuration 13-23

Slave CPUs 10-21
Slot Position 13-15
Slot Positions 13-17
software password 9-7
Software Tools 13-83
Standard Router 3-43, 4-51
Star Router 3-43, 4-55
startgfx 13-83
starting ircombine 13-111
stop graphics 13-103
stopgfx 13-83
switch block 9-21
sync 13-107
System Boot Process 13-85
System Components 3-15
System Features 3-11
SystemPartition 8-15

T

T Bus 13-49
Terminal Attributes 9-43
text-port windo 13-101
Texture Mapping 13-67
texture memory 13-7, 13-9, 13-11
TimeZone 8-15
TM7 13-13
toolchest 13-101
topology command 7-27
tri-pixel 12-13
Triangles 13-63
Troubleshooting Tips 13-105
Two-node System 5-13

U

Ultra SCSI 13-37
Unc 10-49
Uncached Mode 10-49
/usr/bin/X11/4Dwm 13-101
/usr/bin/X11/userenv 13-101
/usr/bin/X11/wait4wm 13-101
/usr/bin/X11/xdm 13-91, 13-97
/usr/bin/X11/Xsgi 13-91
/usr/gfx/stopgfx 13-103
/usr/lib/X11/xdm/xdm-errors 13-97
/usr/lib/X11/xdm/Xlogin 13-99
/usr/lib/X11/xdm/Xresources 13-99
/usr/lib/X11/Xsession 13-101

V

VC Bus 13-49
Vector 12-5
vector route 7-21
Vertex Data 13-57
Vertex Point 12-7
vertical spans 13-63
video output format 13-107
Virtual Debug Switches 9-27
visuallogin 13-93, 13-99, 13-101
VOC 13-73
VOF 13-107
VOF RAM 13-107
voltage margining 9-7
Voltages 9-17
volume 8-15
VTR 13-73

W

Why POD 10-55
Window 12-31
windowsystem 13-91, 13-93
World Space 12-5
World Space Exercise 12-37
Wyse Terminal 9-43
Wyse terminal 9-43

X

X display manager 13-91
X window server 13-91
X-server 13-47
XBOW 3-23
XBow 13-19
xbow status 7-27
xbstat command 7-27
xdm 13-91, 13-93, 13-97
xdm-errors 13-97

XIO Boards 3-45
XIO Interface (II) 4-23
XIO Slots 3-39
XMAP 12-23, 13-73
XpressLinks 5-19, 5-21
Xsession 13-101
.xsession 13-101
Xsgi 13-91, 13-101
Xtown 13-13, 13-19, 13-31

Y

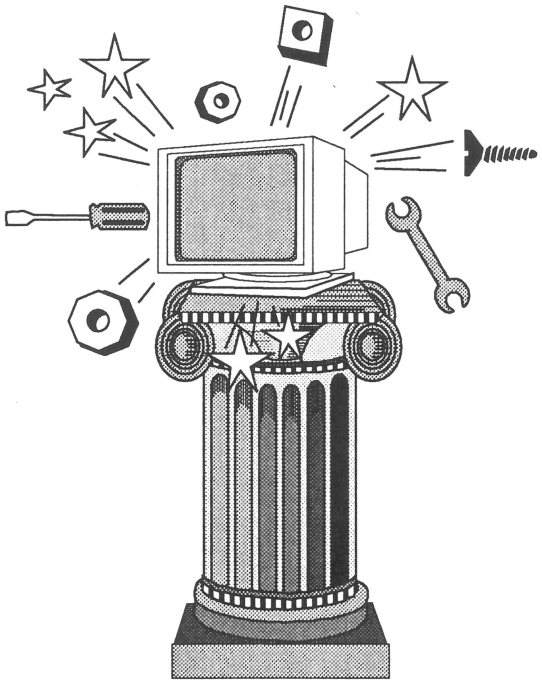
Yellow 12-15

Z

Z-Buffer 12-39
Z-buffer 13-69
Z-Buffer Exercise 12-41

SiliconGraphics
Computer Systems

Module 19: SCSI Peripherals



Part Number: LBT139-2.0-6.4-S-SD-W
October 1997

19-1 FOIL: Module Topics

- SCSI Performance Characteristics
- Origin SCSI Hardware Options
- Origin Vault
- MSCSI XIO Board
- Cabling Options
- Peripheral Devices
- Hardware Graph Files
- SCSI Controller Numbering

19-2 FOIL: Objectives

Upon Completion of this module, using available documentation, you will be able to:

- List the SCSI disk and tape storage systems supported for the Origin servers
- List the versions of Origin vault storage systems available
- List the drive options and channels for the Origin vault
- Identify Origin vault system
- Differentiate between a differential (DF) and single-ended (SF) Origin vault
- Diagram Origin server to Origin vault cabling
- Identify the 5.25-inch drive carrier
- Install and remove a 5.25-inch drive
- Locate the SCSI device ID range jumper
- Successfully install a new Origin vault into an Origin2000 rack or connect an Origin deskside vault to an Origin2000 system
- Correctly identify an Ultra SCSI (MSCSI) XIO card
- Select a correct Origin server XIO slot for MSCSI insertion
- Explain XIO controller numbering
- Correctly install an MSCSI XIO card in an Origin2000 server
- Correctly test for a functioning MSCSI XIO card

References:

Origin200 and Origin Vault Rackmount Kit Installation Guide (Document Number 007-3575-001)

Origin200 and Origin Vault Installation Instructions (Document Number 108-0153-001)

Origin Vault Owner's Guide (Document Number 007-3455-001)

19-3 FOIL: SCSI Performance and Cabling Characteristics

The table below shows the peak performance characteristics of the various SCSI protocols as currently defined in the SCSI standards.

The Origin family's SCSI implementation conforms to the Ultra SCSI Fast-20 specification. All Origin SCSI peripheral **devices** are single-ended:

- 5.25" devices use the Ultra SCSI Fast-Narrow protocol
- 3.5" devices use the Ultra SCSI Fast-Wide protocol

20

Type	Narrow	Wide
SCSI-1	5 MB/sec	10 MB/sec
SCSI-2 (Fast-10)	10	20
Ultra SCSI Fast-20	20	40 (Ultra Wide SCSI)
Ultra 2 SCSI	40	80
Ultra 3 SCSI	80	160

Note: This table reflects *maximum* transfer rates for SCSI protocols. Actual sustained transfer rates can vary substantially.

The Origin family of products supports both single-ended and differential Ultra SCSI Fast-10 and Fast-20 channels. The table below shows the maximum cable length that each channel type supports:

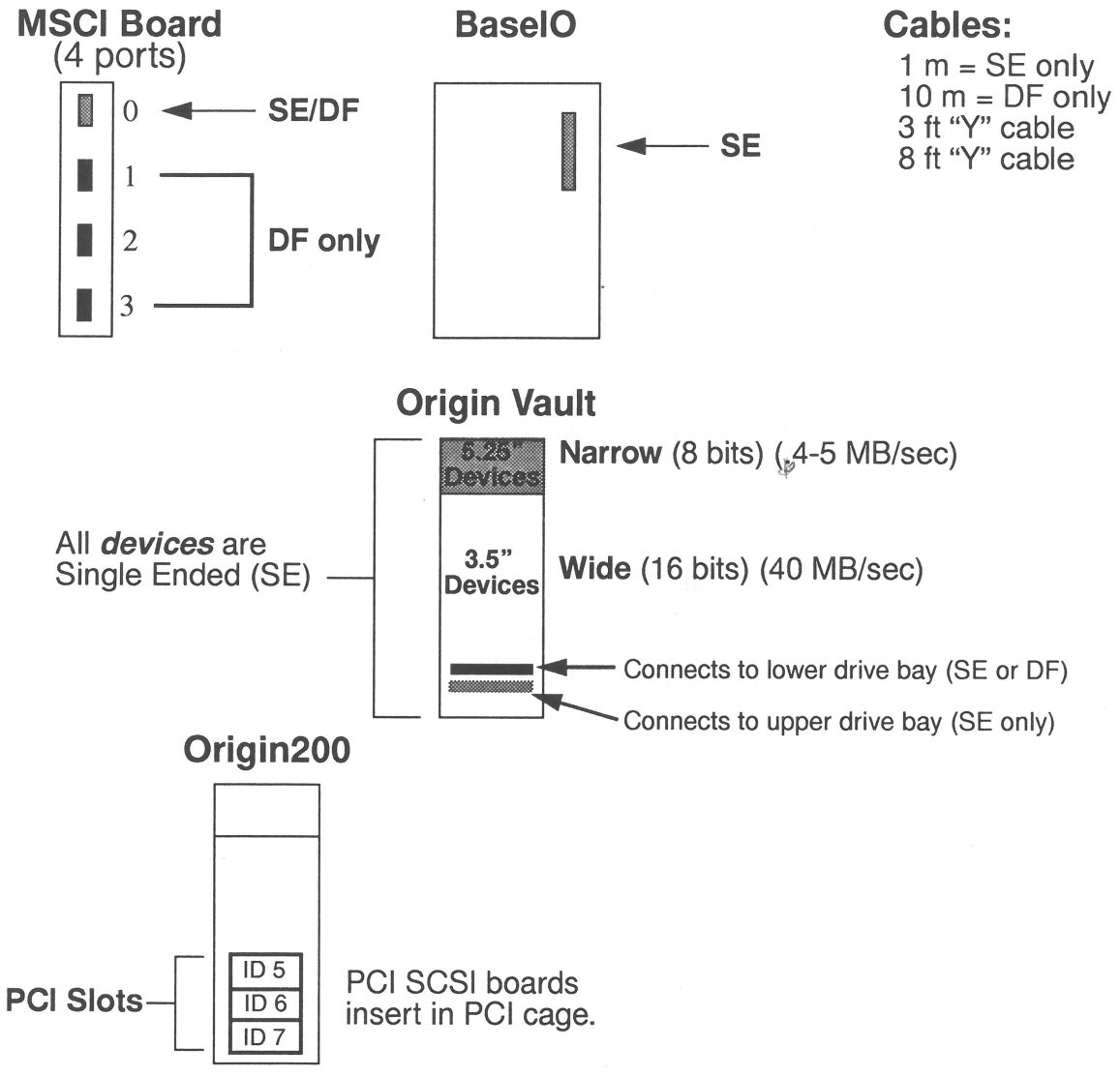
Type	Maximum Cable Length
Single-ended Fast-10	3 m
Single-ended Fast-20	1.5 m
Differential Fast-20	25 m

19-4 FOIL: Origin SCSI Hardware Options

The illustration below shows the various SCSI hardware options that are available for the Origin family of systems.

The Origin Vault is available in single-ended (SE) and differential (DF) versions.

The DF version allows greater cable distances between the controller and the vault; it does not increase the maximum transfer rate of the SCSI bus.



19-5 FOIL: Origin Vault Features and Capabilities

The Origin vault supports the following features:

- Each Origin vault enclosure holds up to:
  - six 3.5-inch disk drives
  - one full-height 5.25-inch tape/CD-ROM or two half-height 5.25-inch tape/CD-ROM
- Provides expansion capacity for Silicon Graphics Origin2000, Origin200, and Onyx2 deskside and rackmount servers and workstations.
- The Origin vault can be configured from 2.2 GB to multiple terabytes of disk space
- Currently supported Ultra SCSI disk drives:
  - 2.2 GB 5400 rpm disk drive (P-S-2)
  - 4.5 GB 7200 rpm disk drive (P-S-4)
  - 9.1 GB 7200 rpm disk drive (P-S-9)

    The 4.5 and 9.1 GB disks have a sustained transfer rate of 9.2MB/sec.
- Currently supported removable media devices:
  - 4mm DDS-2 DAT (Digital Audio Tape) (P-S-DDS2-SCA)
  - 8mm Exabyte (P-S-8MMXL-L-IDC)
  - DLT (Digital Linear Tape) (P-S-DLT7-S)
  - CD-ROM (Compact Disk Read Only Memory):

    P8-S-CDROM4X-1

    MS-D-CDROM-12X

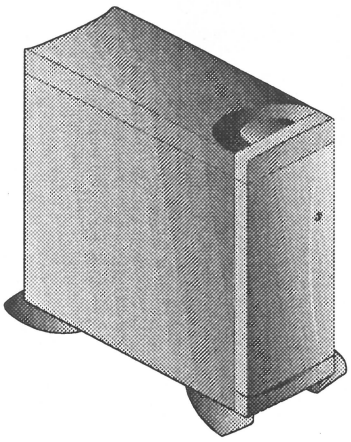
19-6 FOIL: Origin Vault Versions

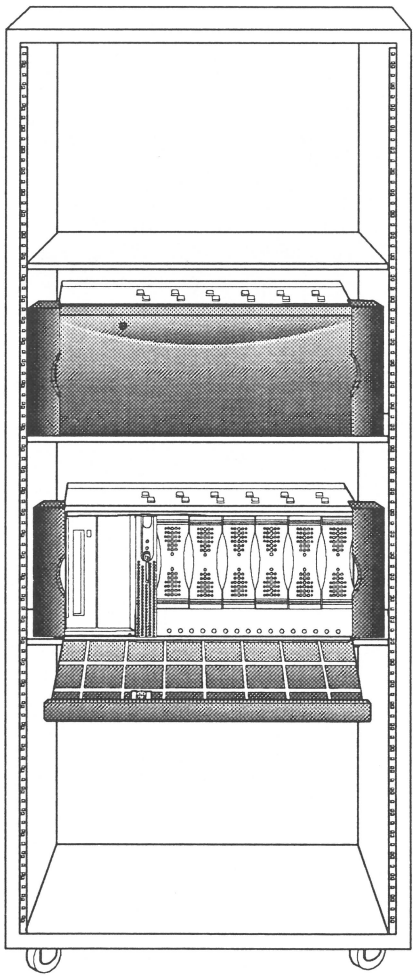
The Origin vault is available in two configurations:

- Deskside
  - The deskside unit resides in the same style chassis as the Origin200 and uses the same skins
- Rackmount
  - The rackmount version comes with mounting hardware and a different front door design

Both the tower and the rackmount enclosure support SE or DF controllers for 3.5" devices. Since all drives are single-ended, the differential option only provides greater cable length for configuration flexibility. Both versions support only SE 5.25" device controllers.

Will mount in 19" or 21" rack.

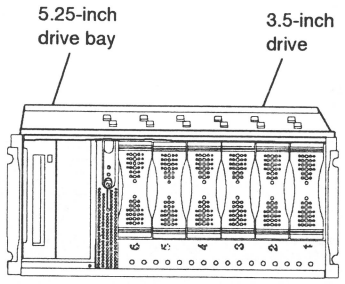


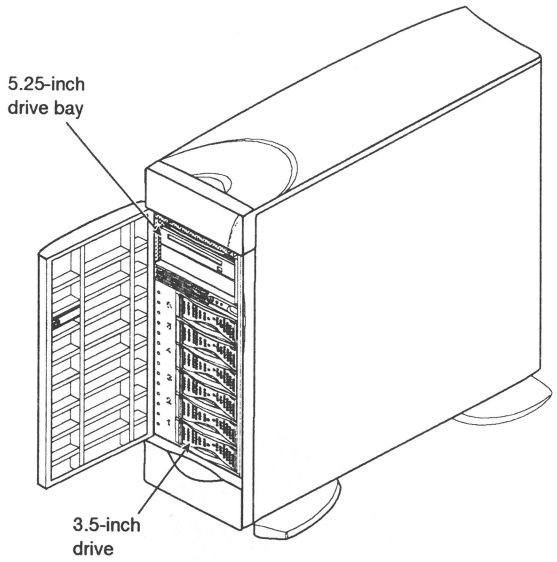


19-7 FOIL: Origin200/Vault Drive Options and SCSI Busses.

The Origin200 and Origin vault support the following SCSI bus and drive options:

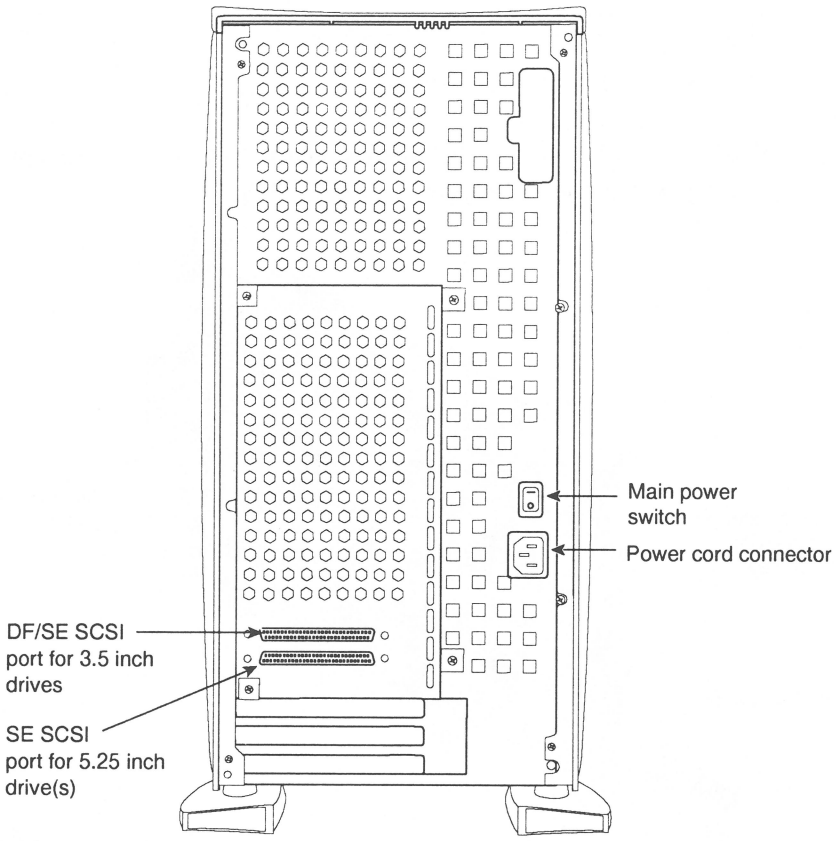
- Two independent SCSI controllers
  - Ultra SCSI Fast-Narrow (for tape devices)
  - Ultra SCSI Fast-Wide (for disk drives)
- The 3.5-inch devices operate at 20 or 40 megabytes/second (Fast-Wide)
- 5.25-inch devices operate at 5 or 10 megabytes/second (Fast-Narrow)
- The SCSI connectors for both the 3.5-inch and 5.25-inch drives terminate within the Origin chassis
- The Origin vault chassis does not currently provide a port for daisy chaining

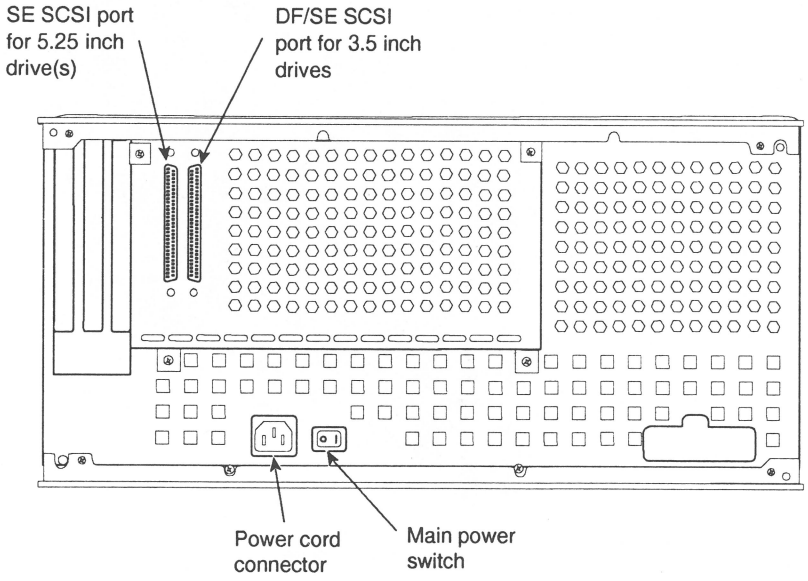




19-8 FOIL: Origin Vault SCSI Cabling Ports

The Origin vault provides separate connectors for the 3.5″ and 5.25″ drive bays:



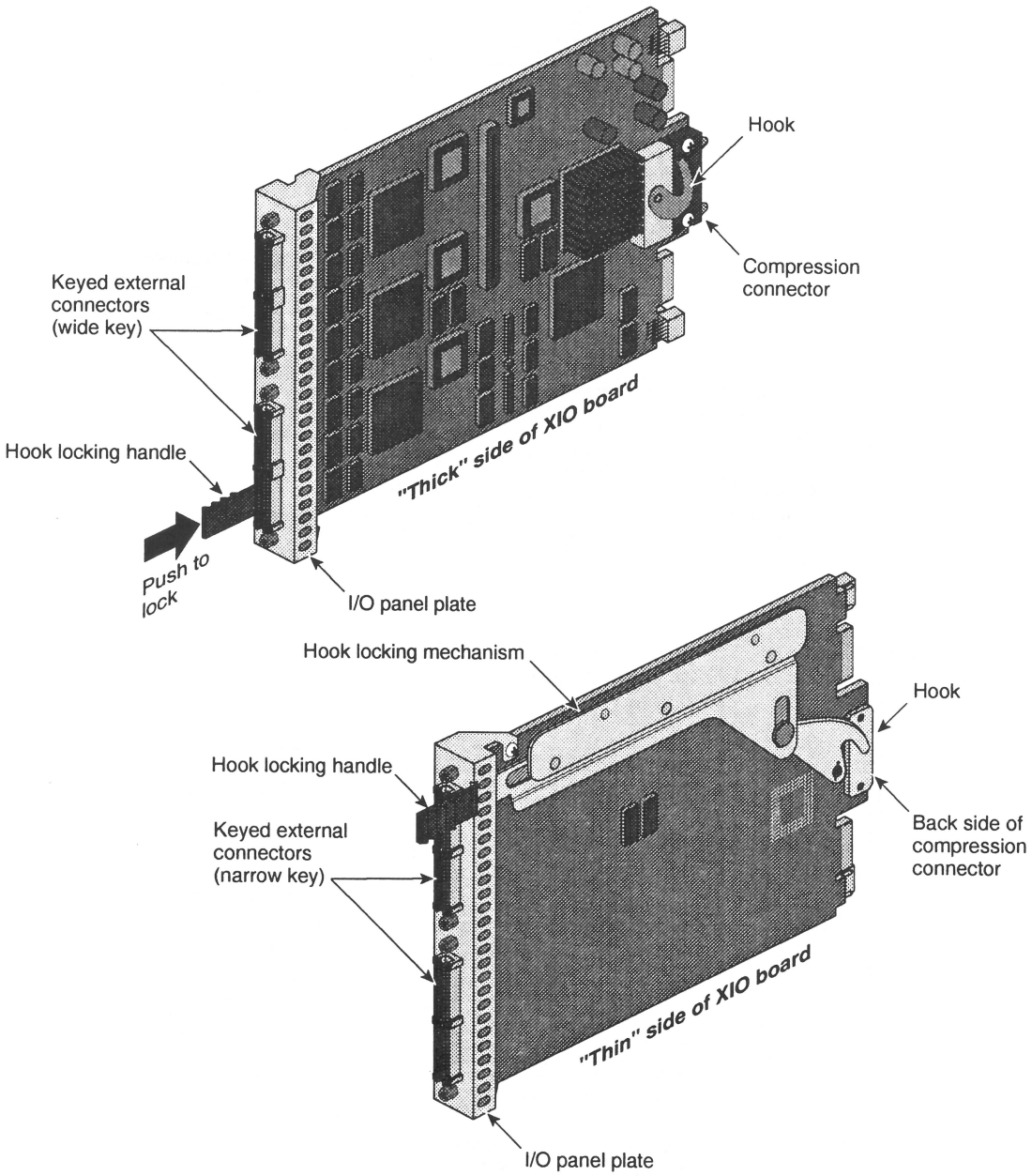


19-9 MSCSI XIO Board

The Ultra SCSI (MSCSI) controller kit consists of a board (030-0872-002), two Y cables (three feet (018-0561-001) or eight feet (018-0561-102) long) and sheets of labels. The board has four Fast-20 channels:

- channel 0: DF or SE (autosensing)
- channels 1, 2 and 3, DF only

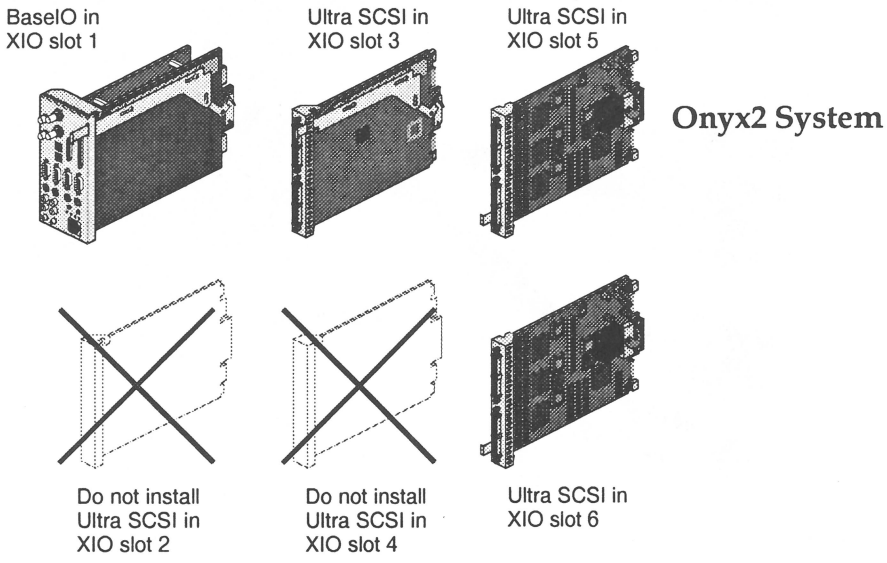
Label sheets included in the option contain peel-off stickers for modules and for XIO slot/SCSI channel designation.



FOIL: MSCSI Board Placement Restrictions

The following restrictions apply to MSCSI board placement:

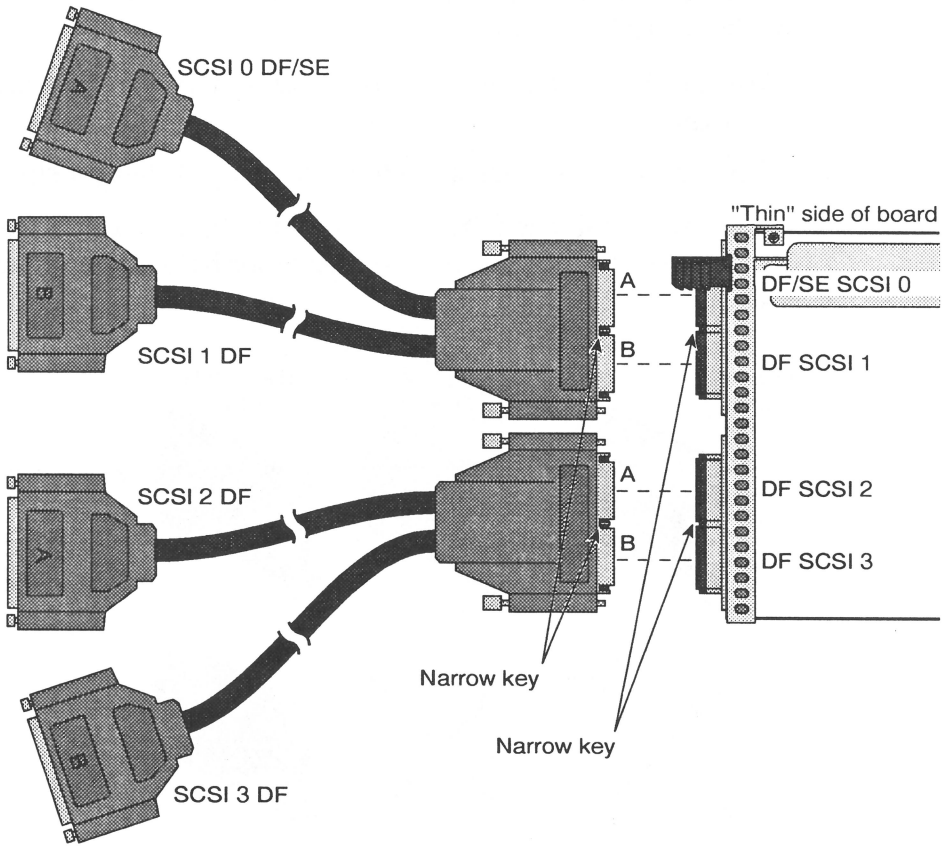
- Because of its compression connector type, the MSCSI board cannot be inserted into slot 2 in any Origin2000 or Onyx2 system.
- In modules with no IO6 board, the MSCSI board can go into slot 1, where it activates the internal SCSI channels for the module and enables the system's internal disk drives and CD-ROM drive to be used
- The Onyx2 deskside system has a different board orientation scheme from the other three systems
  - You can install either three full-height boards or five half-height boards
  - Slot 4 combines with slot 3 to provide a full-height slot
  - You can install the Ultra SCSI board (a half-height board) into slot 3, but you cannot install it or any other half-height board into slot 4



FOIL: Cabling the MSCSI Board

As you cable the devices to the MSCSI board, attach a slot/channel label to each end of each leg of each Y cable, and to each end of any additional cables needed to connect to the target SCSI device.

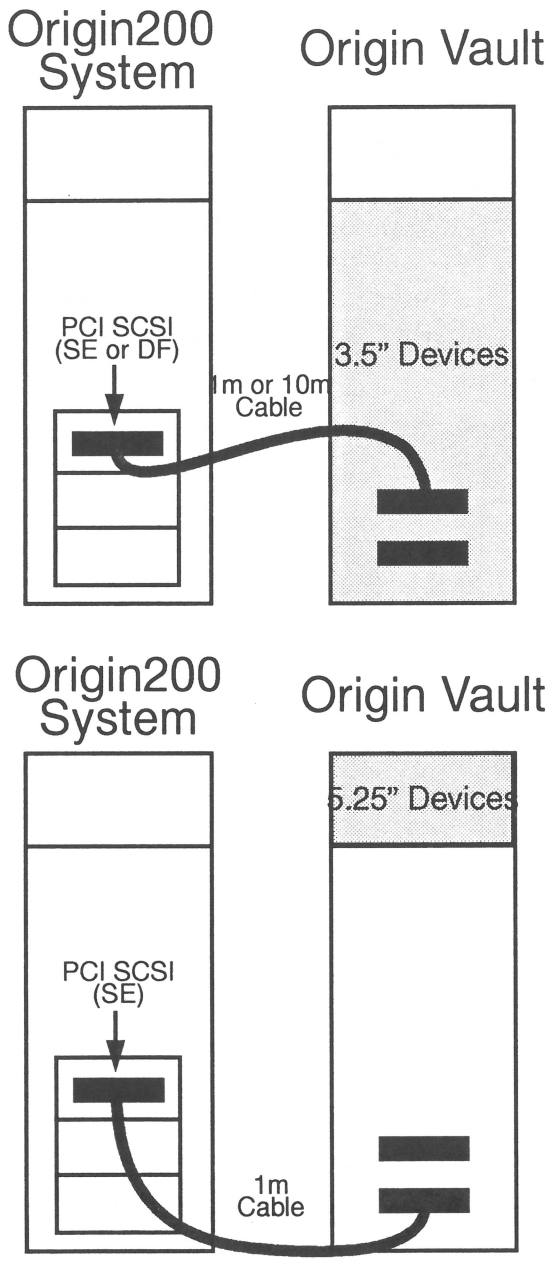
Warning: The high-density cable connectors that plug into the board ports are keyed; if they are inserted incorrectly, the cable connector is destroyed, and the system does not see the bus and the devices on it.



19-10 FOIL: Origin200 SCSI Cabling Options

You can cable an Origin200 system to external SCSI devices with a PCI SCSI controller board. SE and DF PCI controllers are available.

See Origin 200 Owner's Guide (007-3415-002) for PCI installation instructions.



19-11 FOIL: Origin2000 SCSI Cabling Options

You can use the BaseIO (IO6) external SCSI channel or an XIO MSCSI (X9-SCSI-4P) controller to connect an Origin2000 system to deskside or rackmount Origin vaults.

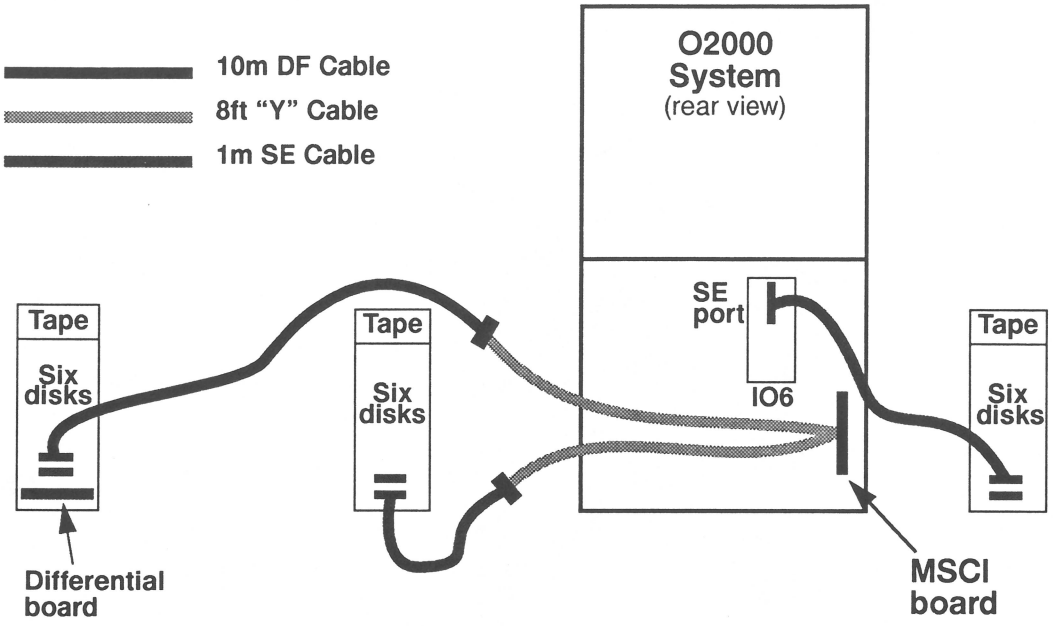
The various cable lengths and cable combinations that are required to connect the controller to SE and DF vaults are shown in the illustration on the next page.

Controller kits do not include 1m and 10m cables; you must order them separately.

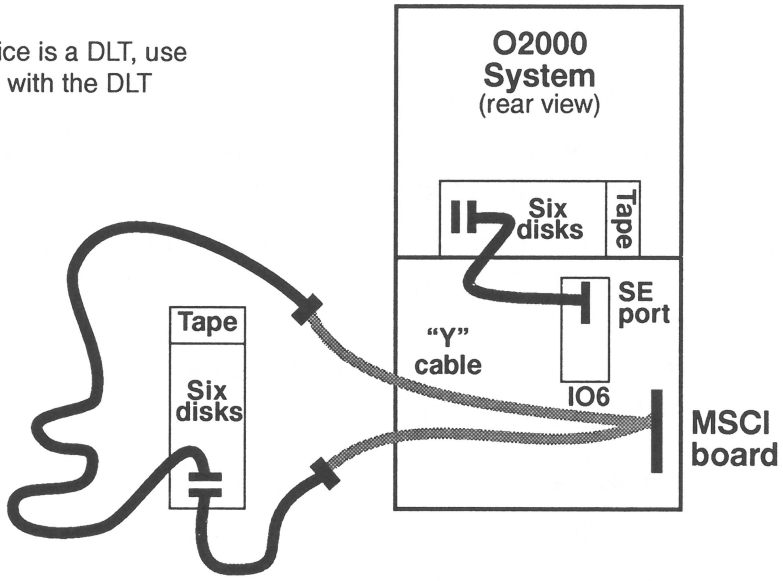
You can use an 8 ft. "Y" and a 1 m cable to attach MSCSI channel 0 to a 5.25" device installed in an Origin vault. PROM `hinv` will not find the device, but it will be recognized when you boot IRIX.

Note: Because their connector genders are reversed, you cannot connect "Y" cables directly to the Origin vault.

19-12 FOIL: Origin2000 SCSI Cabling Options (diagram)



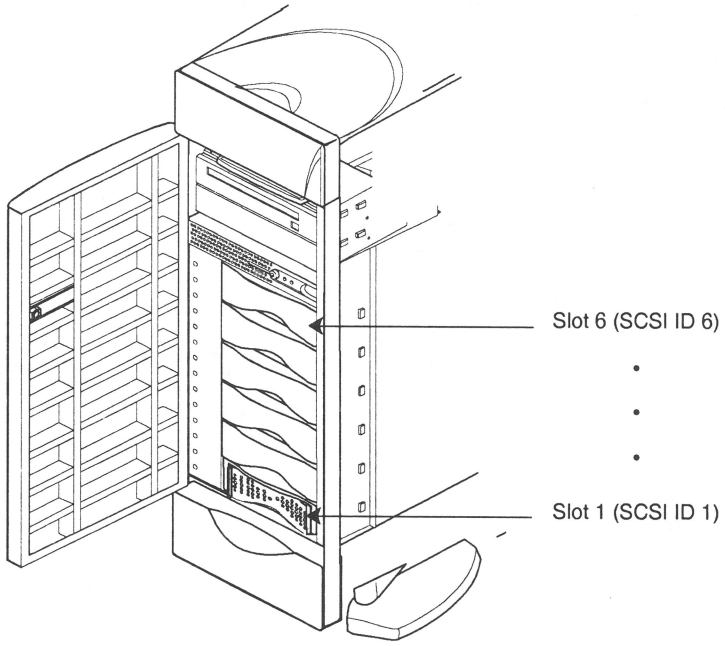
Note: If the tape device is a DLT, use the cable that comes with the DLT option.



19-13 FOIL:3.5-Inch Disk Drives

The 3.5-inch disk drives provide the following features:

- Sled-mounted interface to the SCSI backplane with a Fast-20 80-pin SCA (Single-Connector Assembly)



- Slot-assigned SCSI IDs
  - By default, drives are numbered 1 through 6, starting with the slot farthest from the power button.
  - Drive numbering from 9 through 14 is available by jumpering two pins on the backplane.
- The Origin Vault expansion option is shipped either with six 3.5-inch disk modules or with no disk modules. You can order individual disk modules as shown below:

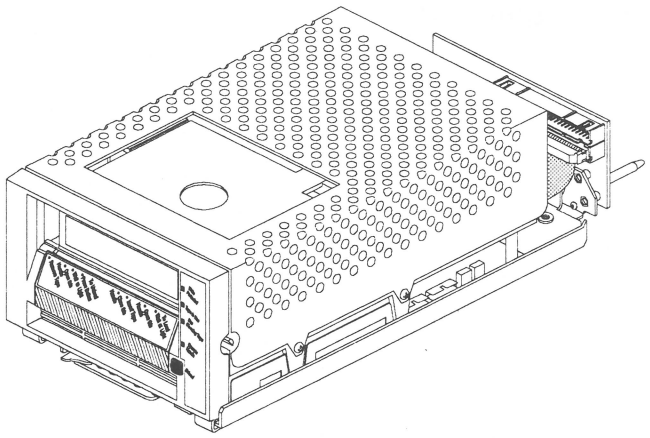
Disk Module	Marketing Code	Systems
2.2-GB SE Fast-20 SCSI	P-S-2	Origin2000, Origin200, and Octane
4.5-GB SE Fast-20 SCSI	P-S-4	Origin2000, Origin200, and Octane
9.1-GB SE Fast-20 SCSI	P-S-9	Origin2000, Origin200, and Octane

- Currently, only the Origin200 platform supports the hot-swapping of disk devices

19-14 FOIL: 5.25-Inch Tape/CD-ROM Drives

The 5.25-inch bay provides a Fast-10 rate of transfer and supports the device types below. 5.25-inch drives are single-ended (narrow) only (the internal cable is a 50-pin cable).

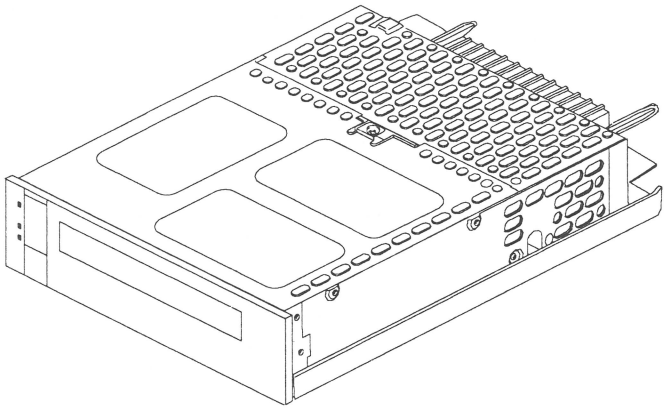
- One full-height removable media drive, such as a a DLT™ (Digital Linear Tape) tape drive (P-S-DLT7-S):



- One or two half-height removable media drives:
  - CD-ROM drive

    P8-S-CDROM4X-1

    MS-D-CDROM-12X
  - 4mm DDS-2 DAT (Digital Audio Tape) (P-S-DDS2-SCA)
  - 8mm Exabyte (P-S-8MMXL-L-IDC)



19-15 FOIL: Drive LEDs

- Each drive has a corresponding LED, mounted on the SCSI backplane, which is visible from the front of the tower or expansion chassis by the use of light pipes
- The drive LEDs indicate disk activity
- Drive fault is indicated by messages on the system console
- A LED next to the power button lights to indicate that power is being supplied to the Origin Vault

19-16 FOIL: Proper Operating Conditions

Supported operating conditions:

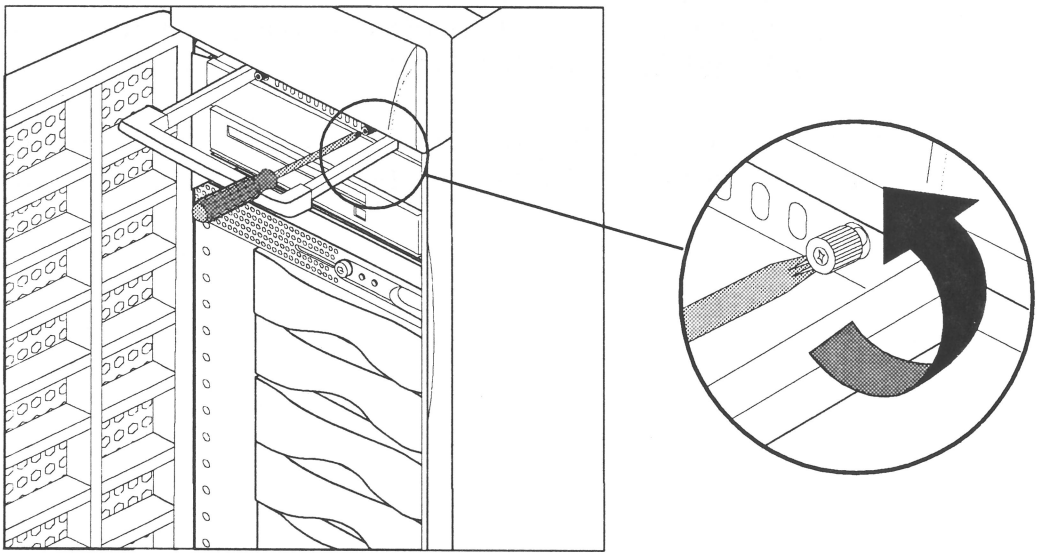
- If the standalone tower is connected to an Origin200 system, both systems must be plugged into the same grounded power circuit to prevent damage to components on the Origin200 system main system board
- The standalone tower should have air clearances of four inches (2.5 cm) on all sides. For more information, see Site Preparation for Origin Family and Onyx2 (007-3452-001)

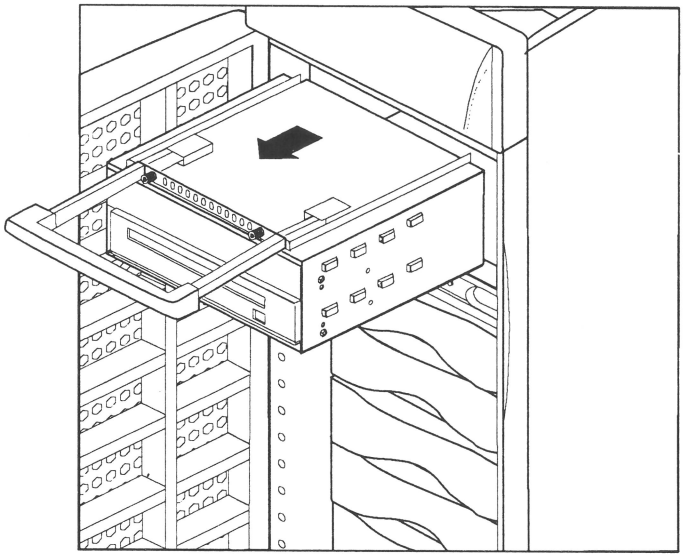
Air flow through the Origin Vault is from the front of the system to the rear, and is drawn by three DC fans located just behind the drive bays.

19-17 FOIL: Installing 5.25-Inch Tape/CD-ROM Drives

5.25-inch drives are installed in a removable carrier.

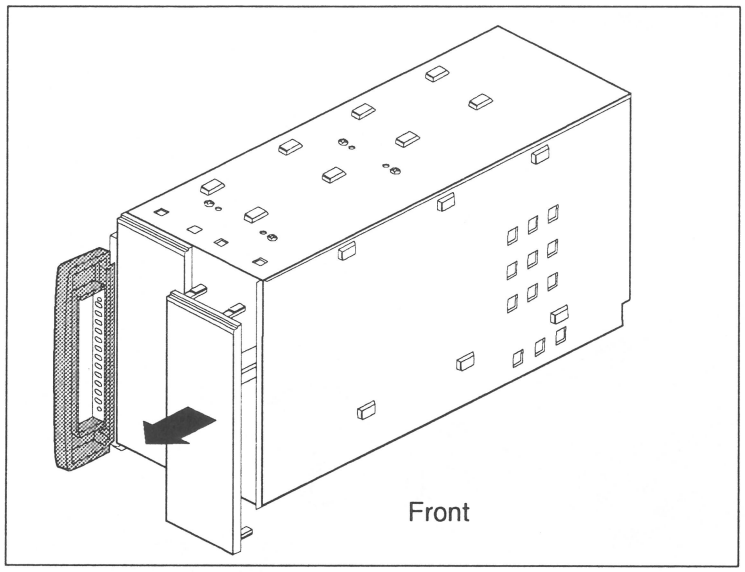
- Pull out the carrier handle.
- Release the captive screws that attach the carrier to the system.
- Pull the carrier out of the system by the handle.

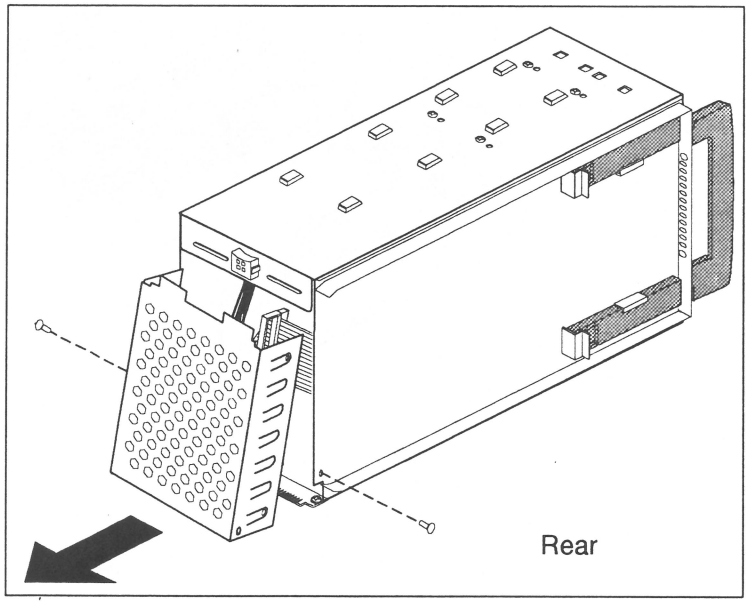




19-18 FOIL: Installing 5.25-Inch Tape/CD-ROM Drives (Cont.)

- Remove plates from the front and rear of the carrier to install or remove devices.

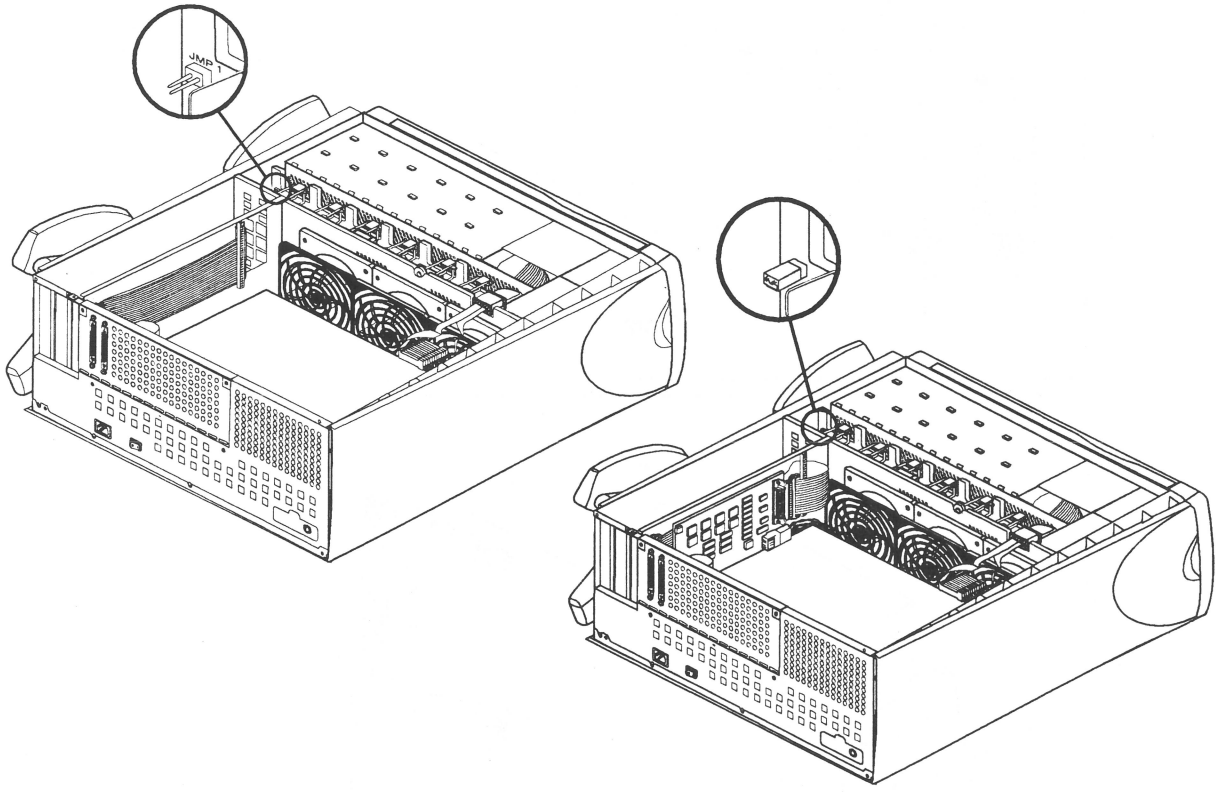




19-19 FOIL: Jumpering the 3.5-Inch Disk Drive SCSI ID Range

The default SCSI IDs for the drives are 1 to 6, starting at the drive farthest from the on/off button.

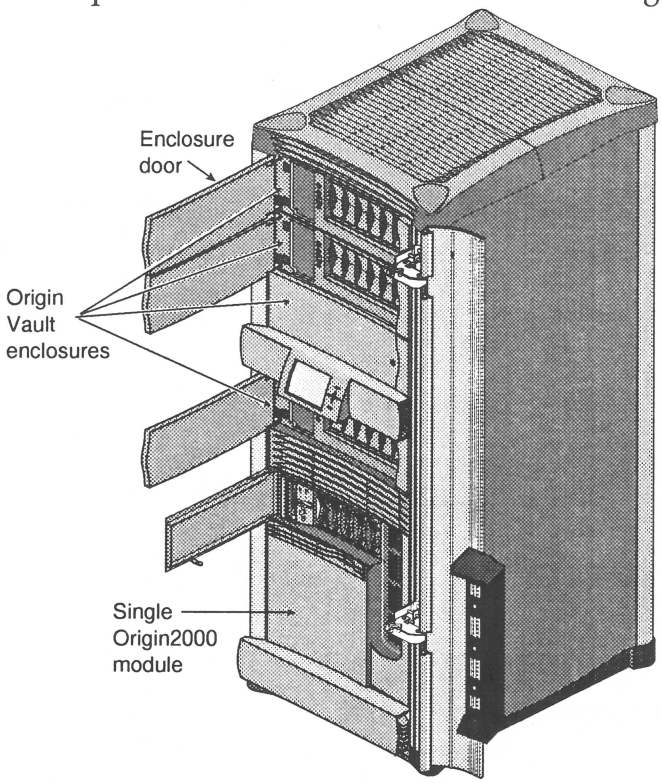
To reset the drives to a higher range of SCSI IDs (9 to 14), install a jumper at the location shown below:



19-20 FOIL: Installing Origin Vault Enclosures in the Rack

Warning: Lifting the Origin Vault rackmount enclosure into a rack requires two people.

- Up to four Vaults can be installed in a single-module rack:



- A 10-meter (X-DS-10M) cable ships with the Vault
- Shorter cables are available:

Marketing Code	Description
X-S-1M	CBL 1-METER SE SCSI
X-S-2M	CBL 2-METER SE SCSI

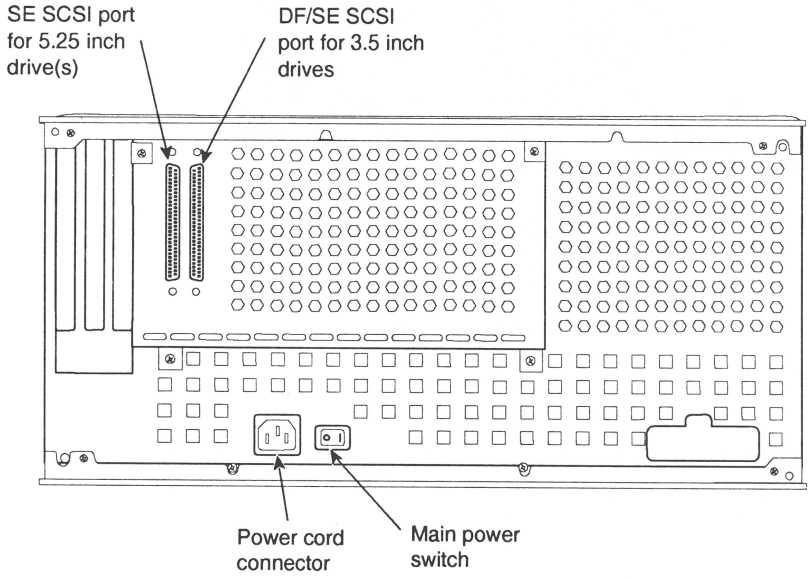
- The rackmount vaults ship with mounting ears and rails

19-21 FOIL: Cabling the Enclosure

- The external port on the Origin2000 IO6/IO6G is a single-ended port
- MSCSI XIO board output channels 1, 2, and 3 are differential; channel 0 is autosensing to work with either differential or single-ended SCSI devices

Note: If you connect a differential SCSI port to a single-ended expansion option, the entire differential bus shuts down

Note: If you connect a single-ended SCSI port to a differential expansion option, the drives in the Origin Vault are not visible to the operating system.



19-22 FOIL: Cabling the Enclosure (cont)

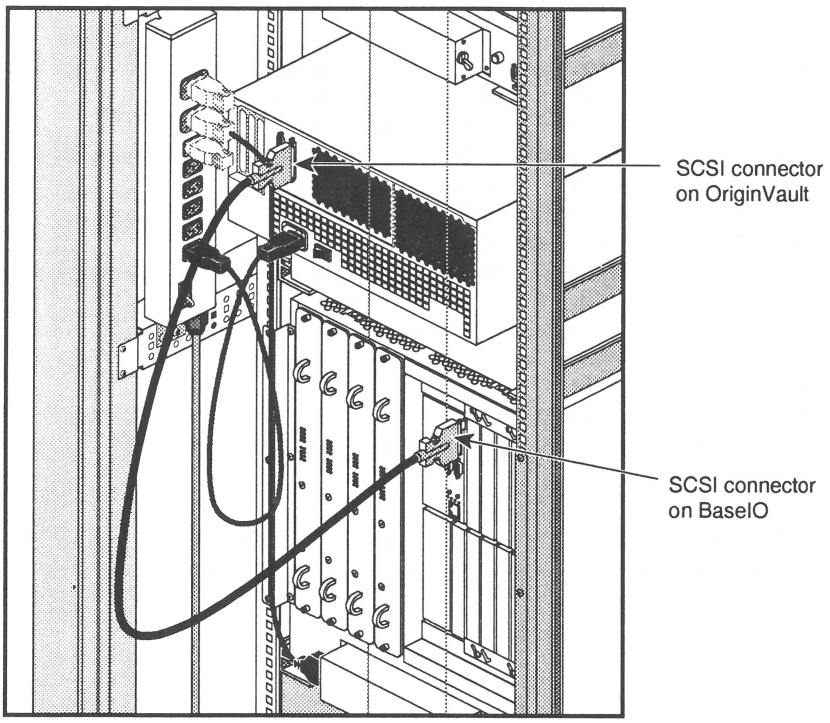
- **Single-ended Vault**: Use the optional 1m SCSI cable to connect the 3.5-inch disk drive port to the IO6 external SCSI port. You can also cable it to a single-ended SCSI port on an option board, such as the PCI option: PCI-SCSI-Q-SE-1P

  If peripherals are present in the 5.25-inch tape/CD-ROM drive bay, connect this controller port to a to a single-ended SCSI port on an option board, such as channel 0 on the MSCSI XIO board or PCI option

- **Differential Vault**: Use the 10m SCSI cable (or 1m or 2m cable option, if you have ordered it) to connect the 3.5-inch disk drive controller to the host.

  You can also cable the vault to a differential SCSI port, such as any channel on the MSCSI XIO board, or the port on the PCI option

If the Origin Vault enclosure connects to the MSCSI XIO board, you can cable the Origin Vault to a SCSI connector on one of the Y cables supplied with the MSCSI controller kit.



19-23 FOIL: Verifying Drive Installation

Use the following checklist to determine if a drive is installed correctly in the Origin Vault tower or rackmount enclosure:

1. When you power on the system, watch the drive LEDs. The LEDs should blink one after the other as they are accessed: first the 3.5-inch drives (sequentially), and then the 5.25-inch drives.
2. Check the system console; drives that are malfunctioning generate an error message on the system console as the system boots.
3. When the system boots in either single-user mode, or multi-user mode, use `hinv` to list the available drives:

```
FPU: MIPS R10010 Floating Point Chip Revision: 0.0
CPU: MIPS R10000 Processor Chip Revision: 2.6
2 190 MHZ IP27 Processors
Main memory size: 64 Mbytes
Instruction cache size: 32 Kbytes
Data cache size: 32 Kbytes
Secondary unified instruction/data cache size: 1 Mbyte
Integral SCSI controller 0: Version QL1040B
  Disk drive: unit 1 on SCSI controller 0
  Disk drive: unit 2 on SCSI controller 0
  Disk drive: unit 3 on SCSI controller 0
  Disk drive: unit 4 on SCSI controller 0
  Disk drive: unit 5 on SCSI controller 0
Integral SCSI controller 1: Version QL1040B
Integral SCSI controller 7: Version QL1040B
  Disk drive: unit 1 on SCSI controller 7
  Disk drive: unit 2 on SCSI controller 7
  Disk drive: unit 3 on SCSI controller 7
  Disk drive: unit 4 on SCSI controller 7
  Disk drive: unit 5 on SCSI controller 7
  Disk drive: unit 6 on SCSI controller 7
Integral SCSI controller 8: Version QL1040B
Integral SCSI controller 9: Version QL1040B
Integral SCSI controller 10: Version QL1040B
IOC3 serial port: tty1
IOC3 serial port: tty2
Integral Fast Ethernet: ef0, version 1
IOC3 external interrupts: 1
```

Note: The PROM monitor version of **`hinv`** will not return SCSI controller numbers unless the **`ProbeAllScsi`** variable is set to **`y`**.

19-24 FOIL: Verifying Drive Installation (Cont.)

4. Make sure that the system sees the board and attached peripherals; look for the hardware graph listing of the XBOW ASIC that controls the MSCSI board you installed. The example below shows the active XIO widget IDs for XBOW 0:

```
# ls /hw/module/1/slot/n1/node/xtalk
```

The following output shows an active XIO link with an XIO widget ID of 14 octal (0wE). Widget ID 14 corresponds to XIO slot 5.

```
14
```

To obtain a more specific location, use the -l option of the ls command to link the widget ID to an XIO slot number; for example:

```
# ls -l /hw/module/1/slot/n1/node/xtalk
```

The following output shows XIO widget ID 14 linked to an MSCSI option board installed in XIO slot 5:

```
drwxr-xr-x    0 root     sys             0 Mar 14 17:36 0
lrw-------    0 root     sys            28 Mar 14 17:36 14 ->
/hw/module/3/slot/io5/mscsi
```

5. Use the hardware graph listing to determine if the system sees the SCSI devices to which the MSCSI board is cabled.

You can also use `hinv` to check the number of system controllers and devices.

19-25 FOIL: Configuring Drives

Silicon Graphics-provided peripherals are all recognized by the IRIX operating system and are ready to be configured for use with the system. For general reference on disks and filesystems, consult the latest edition of IRIX Admin: Disks and Filesystems.

Information for Drive Configuration

Operation	Reference (Man) Pages or Commands
Creating device files	MAKEDEV(1M) mknod(1M)
Disks	
Formatting and initializing	fx(1M)
Volume header files, adding and removing	dvhtool(1M)
Displaying partitions	prtvtoc(1M) fx(1M)
Repartitioning	fx(1M) *xdkm* graphical tool
System (boot) disk creation	PROM monitor (from the System Maintenance Menu) fx(1M), mkfs(1M), MAKEDEV(1M) dump(1M), xfsdump(1M)
Mounted filesystem, adding	Add_disk(1)
Use quotas	quota(1)
Filesystems	
Creating an EFS or XFS	mkfs(1M)
Spanning across multiple disks	xlv(7M)
Mounting and unmounting	mount(1M), umount(1M) fstab(4)
Automatically mounting (automount)	automount(1M) exportfs(4) chkconfig(1M)

19-26 FOIL: The Hardware Graph (hwgraph/hwgfs)

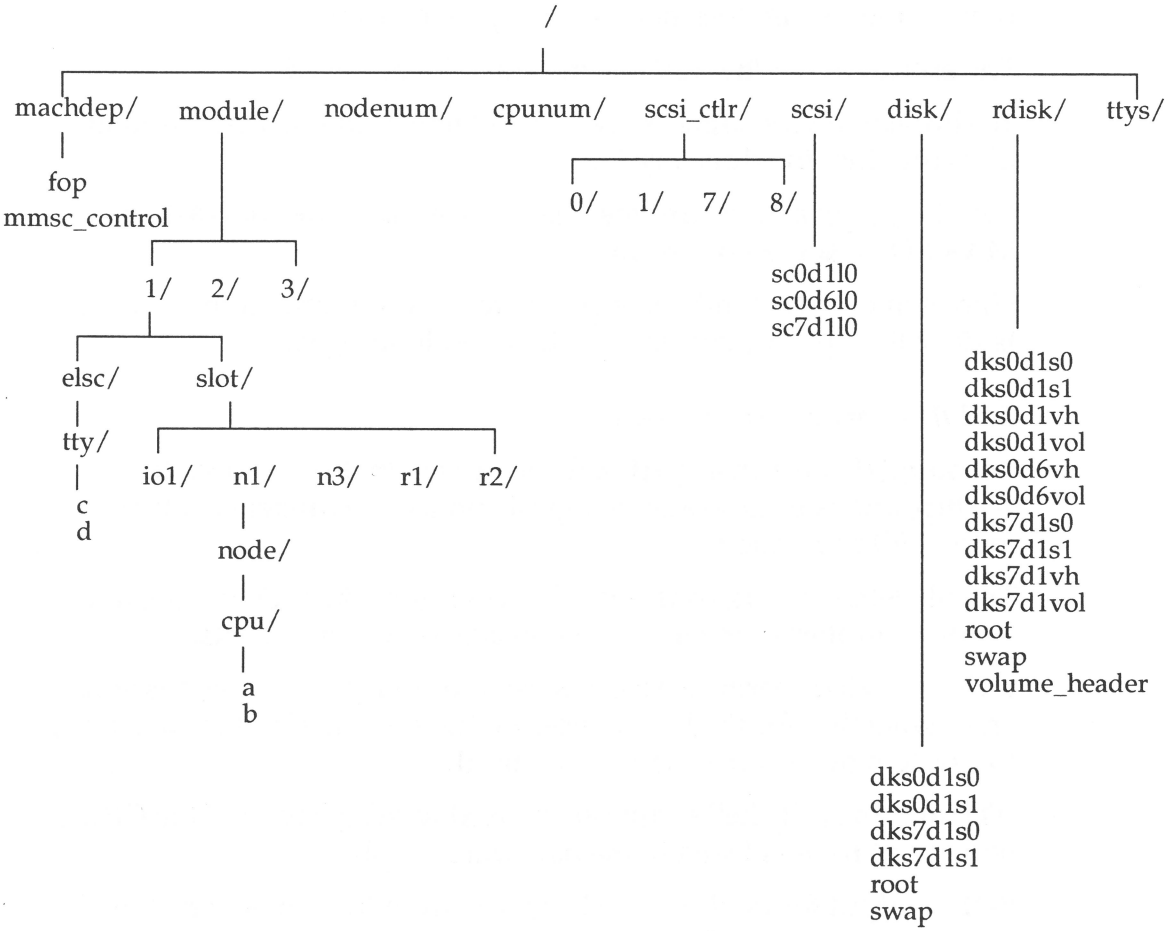
At every boot, the kernel builds the **hwgraph** database and creates entries in the **/hw** pseudo filesystem.

- Since the **hwgraph** is maintained by the kernel in memory, the **/hw** pseudo filesystem does not take up space on disk
- The entries in the **/hw** filesystem cannot be removed
- Traditional device names have been retained but most are symbolic links into the **/hw** directory tree
- For "hwgraph aware" drivers, the older utilities **mknod(1M)** and **MAKEDEV** serve no function
- Since some of the symbolic links are recursive, search commands like **ls -R** will loop. One current way to avoid looping is:

  find /hw -print | xargs ls -ld
- **Ioconfig(1M)** is run as part of the script **/etc/bcheckrc** at system startup, and is used to assign logical controller numbers to all devices in the I/O subsystem
- The file **/etc/ioconfig.conf** is used to configure the I/O devices, and keeps controller numbers consistent across system reboots.
- If the file **/etc/ioperms** exists, it is used to configure the permissions and ownership for the I/O devices on the system. If it does not exist, the system default permissions are used.
- The **topology(1)** shell script can be used to get a listing of the CPUs, nodes and routers listed in the hardware graph.
- Better output for field use can be gotten from **hinv -m -v**, which will include memory display.

19-27 FOIL: The Hardware Graph (hwgraph/hwgfs) (Cont.)

The following is part of a typical **/hw** directory tree (**hwgraph**):



19-28 FOIL: The /etc/ioconfig.conf File

- The **/etc/ioconfig.conf** file is created and appended to by the **ioconfig** command in **/etc/bcheckrc**.
- Changes to the I/O configuration result in additions to this file.
- Unless the file is removed prior to a reboot, it is possible for multiple entries to exist for a given slot (see sample file below for module 1, slot 2).

Sample output:

```
shipit01> cat /etc/ioconfig.conf
2 /hw/module/1/slot/io2/pci_xio/pci/0/scsi_ctlr/0
3 /hw/module/1/slot/io2/pci_xio/pci/1/scsi_ctlr/0
4 /hw/module/1/slot/io2/pci_xio/pci/2/scsi_ctlr/0
5 /hw/module/1/slot/io3/fibre_channel/pci/0/scsi_ctlr/0
6 /hw/module/1/slot/io3/fibre_channel/pci/1/scsi_ctlr/0
7 /hw/module/1/slot/io4/fibre_channel/pci/0/scsi_ctlr/0
8 /hw/module/1/slot/io4/fibre_channel/pci/1/scsi_ctlr/0
9 /hw/module/1/slot/io5/fibre_channel/pci/0/scsi_ctlr/0
10 /hw/module/1/slot/io5/fibre_channel/pci/1/scsi_ctlr/0
11 /hw/module/1/slot/io6/fibre_channel/pci/0/scsi_ctlr/0
12 /hw/module/1/slot/io6/fibre_channel/pci/1/scsi_ctlr/0
13 /hw/module/1/slot/io7/fibre_channel/pci/0/scsi_ctlr/0
14 /hw/module/1/slot/io7/fibre_channel/pci/1/scsi_ctlr/0
15 /hw/module/1/slot/io8/fibre_channel/pci/0/scsi_ctlr/0
16 /hw/module/1/slot/io8/fibre_channel/pci/1/scsi_ctlr/0
17 /hw/module/1/slot/io9/fibre_channel/pci/0/scsi_ctlr/0
18 /hw/module/1/slot/io9/fibre_channel/pci/1/scsi_ctlr/0
19 /hw/module/1/slot/io10/fibre_channel/pci/0/scsi_ctlr/0
20 /hw/module/1/slot/io10/fibre_channel/pci/1/scsi_ctlr/0
21 /hw/module/1/slot/io11/fibre_channel/pci/0/scsi_ctlr/0
23 /hw/module/1/slot/io11/fibre_channel/pci/1/scsi_ctlr/0
22 /hw/module/1/slot/io12/fibre_channel/pci/0/scsi_ctlr/0
24 /hw/module/1/slot/io12/fibre_channel/pci/1/scsi_ctlr/0
25 /hw/module/1/slot/io2/controller/pci/1/scsi_ctlr/0
26 /hw/module/1/slot/io2/controller/pci/2/scsi_ctlr/0
```

Warning: You should remove duplicate entries by editing the /etc/ioconfig.conf file.

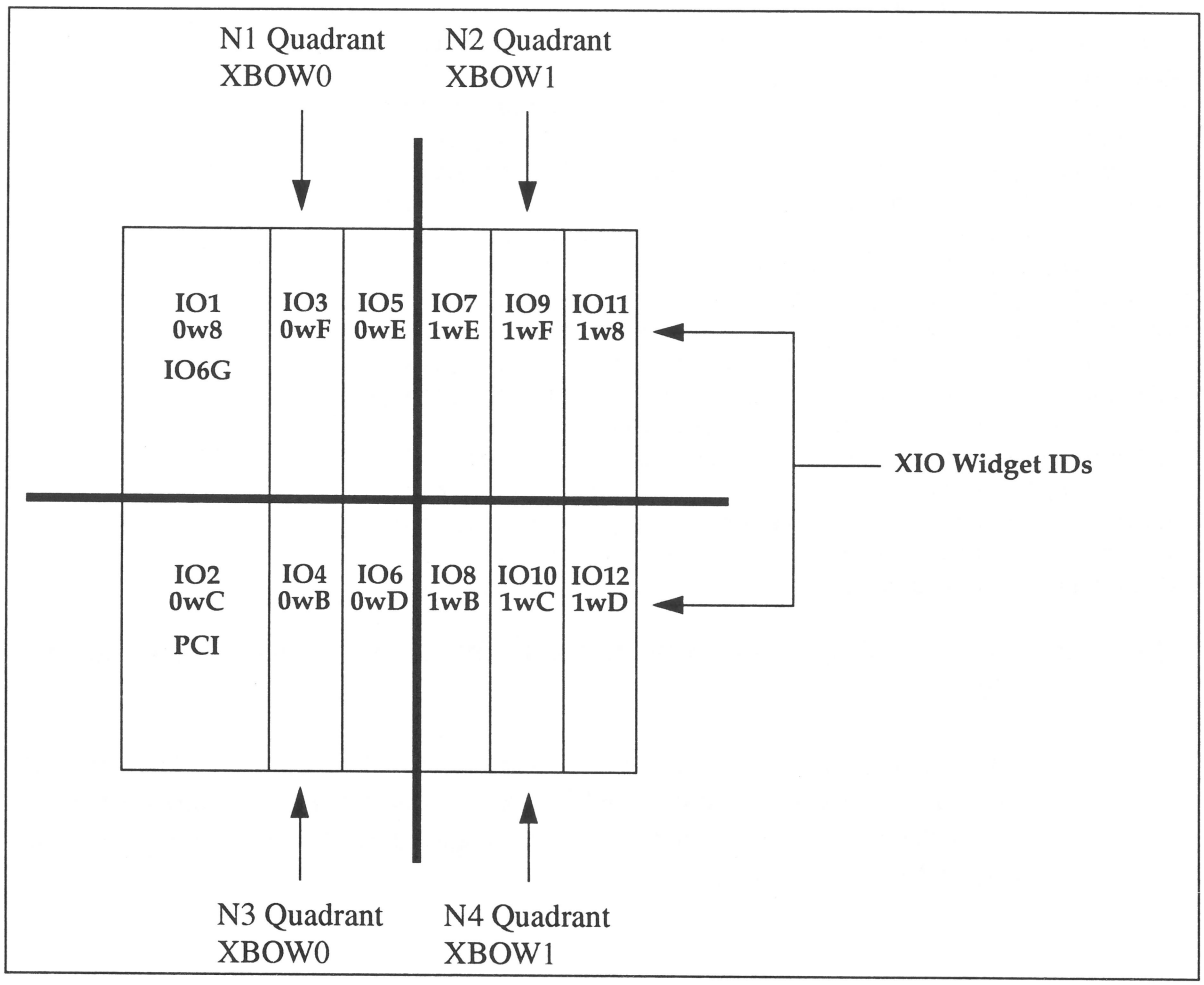
Although you can eliminate duplicate entries by removing the file and resetting the system, you must be extremely careful to ensure that filesystem mount points have not changed after the discovery process renumbers the controllers.

19-29 FOIL: SCSI Controller Numbering

The BaseIO PROM microcode uses a specific algorithm to dynamically assign logical controller numbers during every system reset. This algorithm assigns numbers based on the following priority levels:

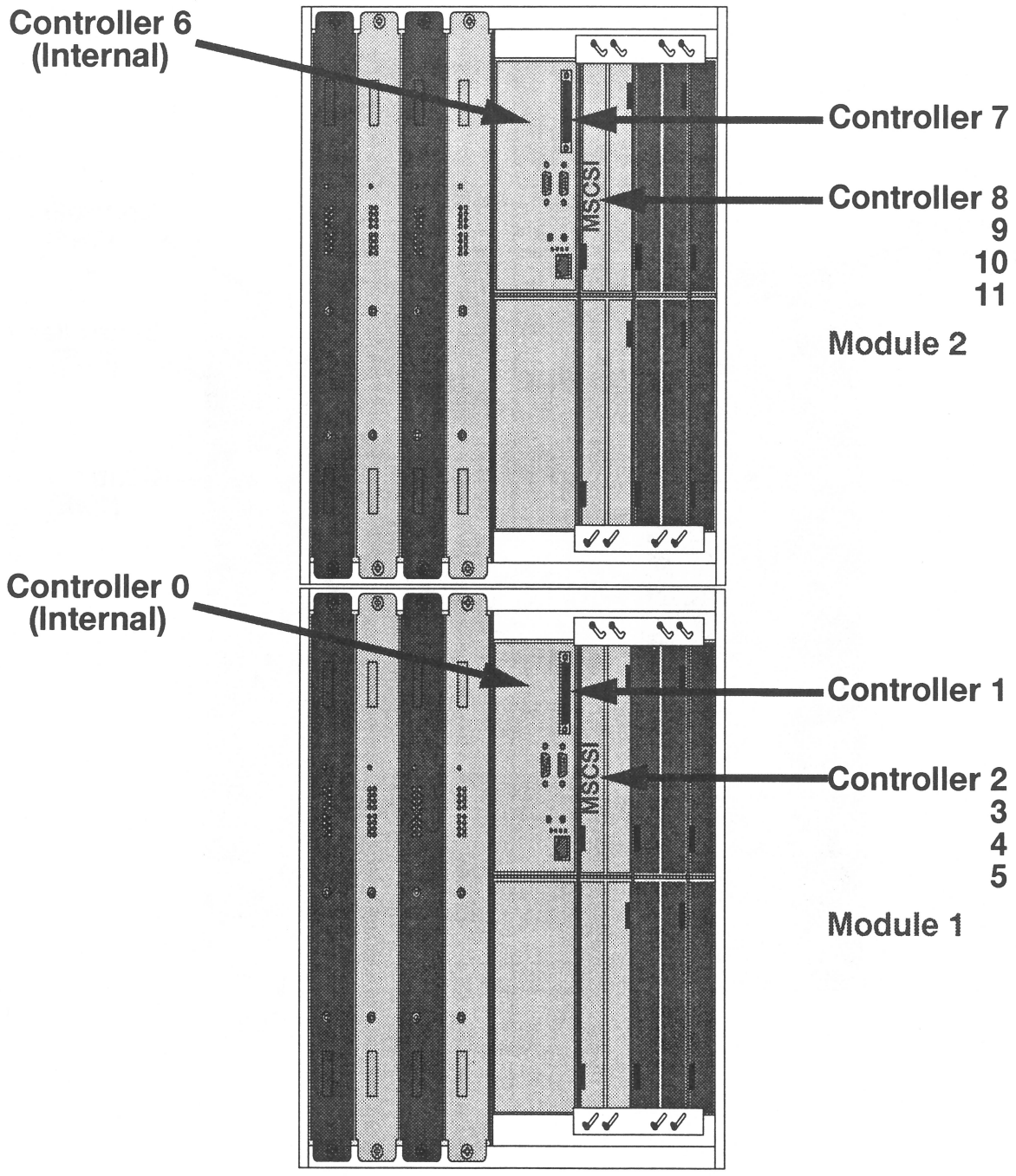
- Module ID
- BaseIO internal SCSI bus
- XBOW number
- XBOW widget ID
- MSCSI controller port

As shown in the illustration below, each XBOW ASIC assigns a widget ID to each XIO slot:



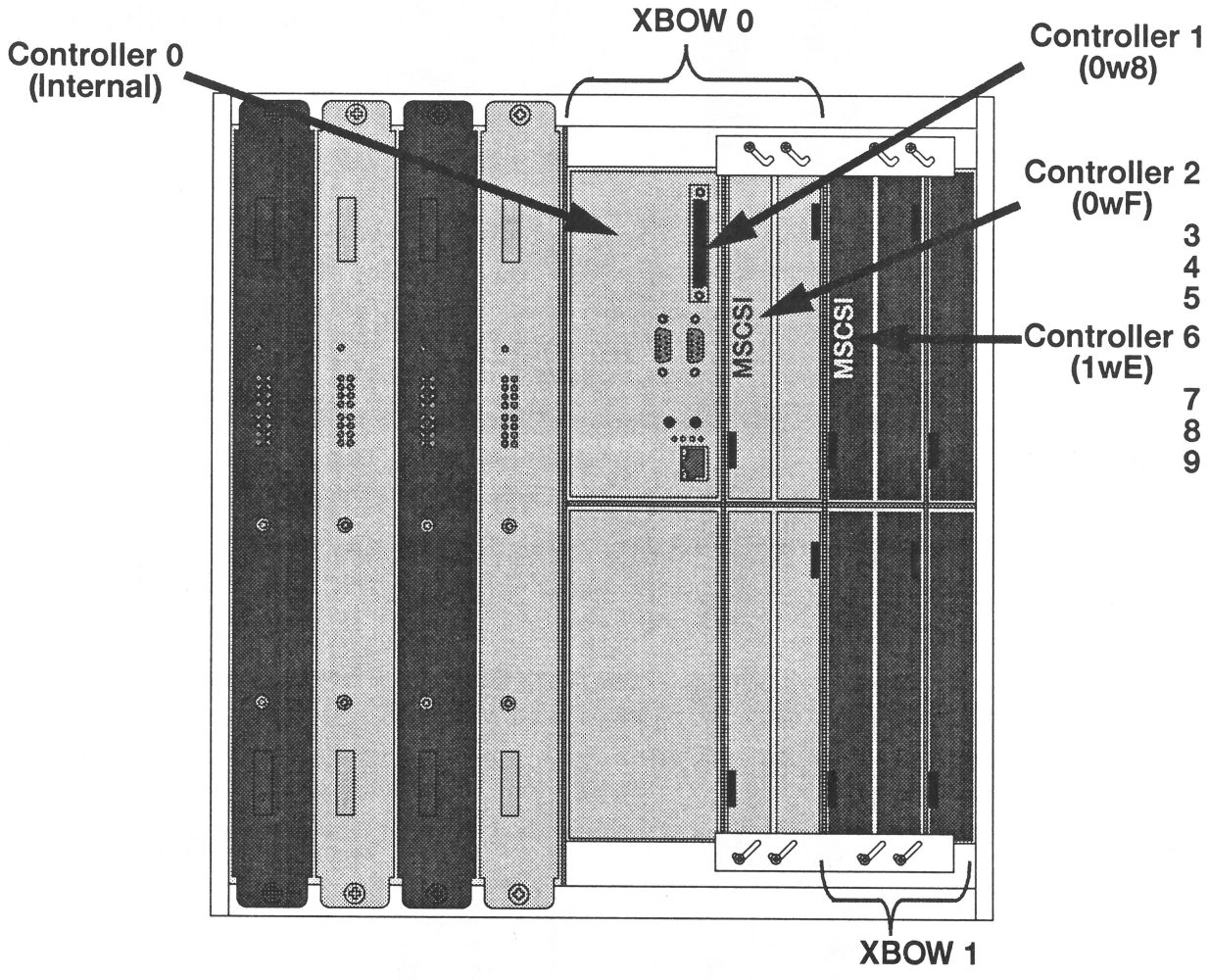
19-30 FOIL: Controller Numbering Examples

The following examples demonstrate the controller numbering algorithm. One MSCSI board is located in XIO slot 3 of each Origin module. In this case, IRIX assigns the lower controller numbers to the MSCSI board in the module with the lowest ID number. Note that the algorithm numbers every port on the MSCSI board whether or not disk drives are attached to that port.



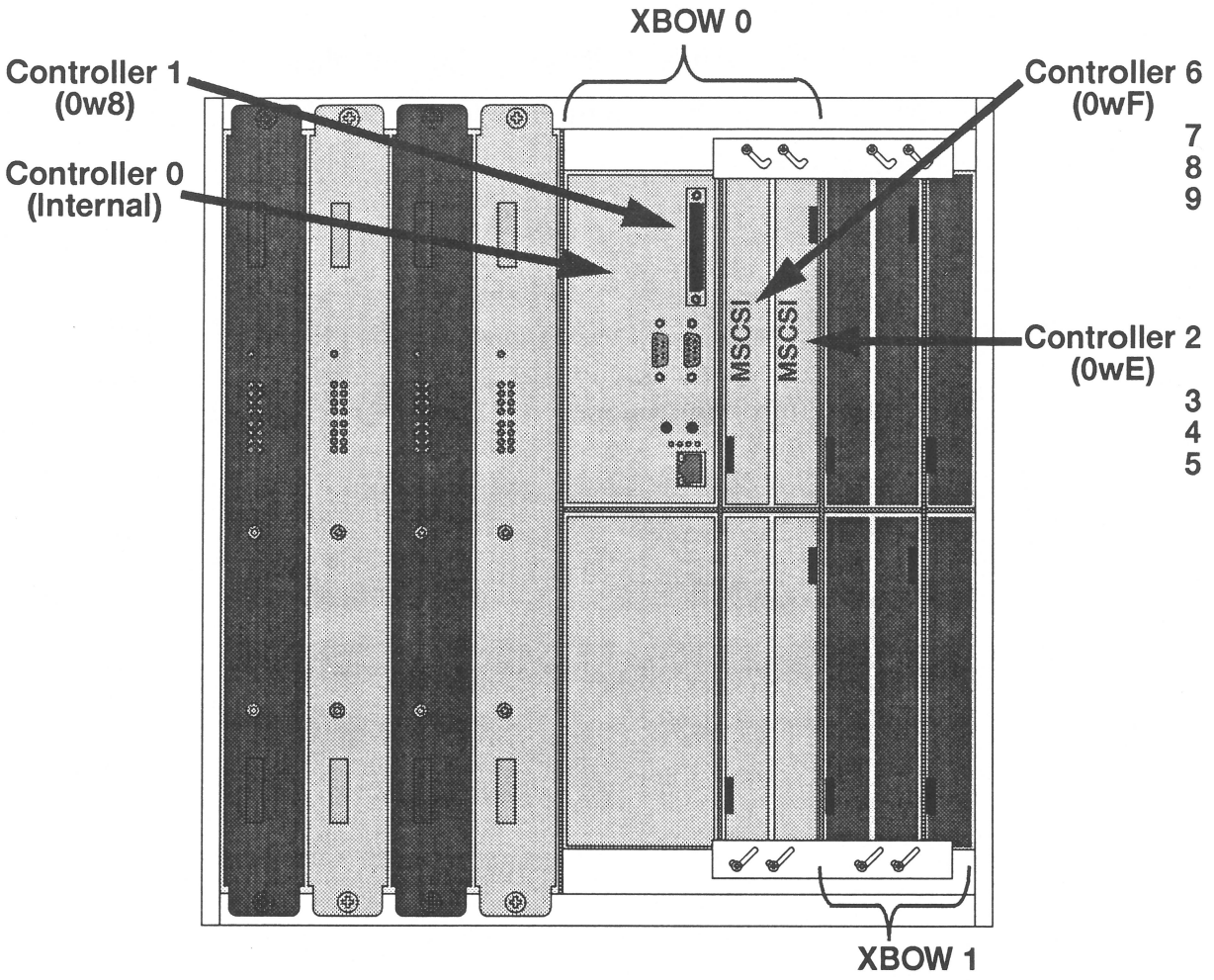
19-31 FOIL: Controller Numbering Examples (Cont.)

If a module contains more than one MSCSI board, the controller numbers are assigned by lowest XBOW and XIO widget number. In the example below, XBOW 0 controls one MSCSI board and XBOW 1 controls a second MSCSI. Therefore, IRIX assigns controllers 2-5 to XBOW 0's board and controllers 6-9 to XBOW 1's. As in the previous example, the BaseIO SCSI controllers are numbered first. XIO widget IDs are shown in parentheses:



19-32 FOIL: Controller Numbering Examples (Cont.)

If a single XBOW controls more than one MSCSI board, IRIX assigns controller numbers by widget IDs. In the example below, one MSCSI board resides in io3 and the other in io5. XBOW 0 controls both boards. The board in io5 receives the lower controller numbers because its widget ID (0wE) is lower than io3's (0wF):



19-33 The SCSI Probe Process

The BaseIO PROM microcode probes the system's available SCSI controllers during every boot discovery process and reports its results to the system console. You can view this information to determine if a system recognizes all installed SCSI devices.

You can modify the behavior of the probe process by changing the value of the `ProbeAllScsi` variable:

- `ProbeAllScsi=n`
  - probes only Module 1's BaseIO SCSI controllers
  - **hinv** reports controller numbers for Module 1's BaseIO only
- `ProbeAllScsi=y`
  - probes all installed SCSI controllers
  - **hinv** reports all logical controllers numbers

Note: By default, the `ProbeAllScsi` variable is set to `n` because the extended probe process can add many minutes to the boot time of a large system configuration.

19-34 FOIL: SCSI Probe Output

The following examples show the SCSI probe output of a 2-module Origin2000 rack system with the following configurations:

- Module 1
  - Device ID 1 on BaseIO internal SCSI controller 0
  - Device ID 6 (CDROM) on BaseIO internal SCSI controller 0
  - Device ID 2 on BaseIO external SCSI controller 1
  - MSCSI board in XIO slot 5 with no disks installed
- Module 2
  - Device ID 3 on BaseIO internal SCSI controller 0
  - Device ID 4 on MSCSI in XIO slot 5, port 0

The first example shows the console and **hinv** output with **`ProbeAllScsi=n`**:

Console output:

```
Installing Devices...

Running enet_all diag (Bridge base = 0x9200000008000000  PCI dev = 2  Mode= 0)
Running scsi_ram diag (Bridge base = 0x9200000008000000  PCI dev = 0)
Running scsi_dma diag (Bridge base = 0x9200000008000000  PCI dev = 0)
Running scsi_controller diag (Bridge base = 0x9200000008000000  PCI dev = 0)

Walking SCSI Adapter 0 (/hw/module/1/slot/io1), (pci id 0)
1+ 2- 3- 4- 5- 6+ 7- 8- 9- 10- 11- 12- 13- 14- 15- = 2 device(s)
Running scsi_ram diag (Bridge base = 0x9200000008000000  PCI dev = 1)
Running scsi_dma diag (Bridge base = 0x9200000008000000  PCI dev = 1)
Running scsi_controller diag (Bridge base = 0x9200000008000000  PCI dev = 1)

Walking SCSI Adapter 1 (/hw/module/1/slot/io1), (pci id 1)
1- 2+ 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 1 device(s)
Initializing devices...
Checking hardware inventory...
```

19-35 FOIL: SCSI Probe Output (Cont.)

PROM hinv Output:

```
>> hinv
System  SGI-IP27
12 180 MHz IP27 Processors
Main memory size: 768 Mbytes
Integral SCSI controller 0
Integral SCSI controller 1
Integral Fast Ethernet
IOC3 serial port
Integral SCSI controller
Integral SCSI controller
Integral SCSI controller
Integral SCSI controller
Integral SCSI controller
Integral SCSI controller
Integral Fast Ethernet
IOC3 serial port
Integral SCSI controller
Integral SCSI controller
Integral SCSI controller
Integral SCSI controller
Graphics Controller
    Disk drive: unit 1 on SCSI Controller 0, (dksc(0,1,0))
    CDROM: unit 6 on SCSI Controller 0, (cdrom(0,6,7))
    Disk drive: unit 2 on SCSI Controller 1, (dksc(1,2,0))
```

19-36 FOIL: SCSI Probe Output (Cont.)

The second example shows the probe output with `ProbeAllScsi=y`:

Console Output:

```
Installing Devices...

Walking SCSI Adapter 0 (/hw/module/1/slot/io1), (pci id 0)
1+ 2- 3- 4- 5- 6+ 7- 8- 9- 10- 11- 12- 13- 14- 15- = 2 device(s)

Walking SCSI Adapter 1 (/hw/module/1/slot/io1), (pci id 1)
1- 2+ 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 1 device(s)

Walking SCSI Adapter /hw/module/1/slot/io5 (pci id 0)
1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 0 device(s)

Walking SCSI Adapter /hw/module/1/slot/io5 (pci id 1)
1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 0 device(s)

Walking SCSI Adapter /hw/module/1/slot/io5 (pci id 2)
1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 0 device(s)

Walking SCSI Adapter /hw/module/1/slot/io5 (pci id 3)
1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 0 device(s)

Walking SCSI Adapter /hw/module/2/slot/io1 (pci id 0)
1- 2- 3+ 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 1 device(s)

Walking SCSI Adapter /hw/module/2/slot/io1 (pci id 1)
1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 0 device(s)

Walking SCSI Adapter /hw/module/2/slot/io5 (pci id 0)
1- 2- 3- 4+ 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 1 device(s)

Walking SCSI Adapter /hw/module/2/slot/io5 (pci id 1)
1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 0 device(s)

Walking SCSI Adapter /hw/module/2/slot/io5 (pci id 2)
1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 0 device(s)

Walking SCSI Adapter /hw/module/2/slot/io5 (pci id 3)
1- 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- = 0 device(s)

Initializing devices...
Checking hardware inventory...
```

19-37 FOIL: SCSI Probe Output (Cont.)

PROM hinv Output:

```
>> hinv
System  SGI-IP27
12 180 MHz IP27 Processors
Main memory size: 768 Mbytes
Integral SCSI controller 0
Integral SCSI controller 1
Integral Fast Ethernet
IOC3 serial port
Integral SCSI controller 2
Integral SCSI controller 3
Integral SCSI controller 4
Integral SCSI controller 5
Integral SCSI controller 6
Integral SCSI controller 7
Integral Fast Ethernet
IOC3 serial port
Integral SCSI controller 8
Integral SCSI controller 9
Integral SCSI controller 10
Integral SCSI controller 11
Graphics Controller
    Disk drive: unit 1 on SCSI Controller 0, (dksc(0,1,0))
    CDROM: unit 6 on SCSI Controller 0, (cdrom(0,6,7))
    Disk drive: unit 2 on SCSI Controller 1, (dksc(1,2,0))
    Disk drive: unit 3 on SCSI Controller 6
    Disk drive: unit 4 on SCSI Controller 8
```