Origin™ and Onyx2™ Theory of Operations Manual

Document Number 007-3439-002

CONTRIBUTORS

Written by Joseph Heinrich
Illustrated by Dan Young and Cheri Brown
Production by Linda Rae Sande
Engineering contributions are listed in the *References and Source Material*.
St Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower image courtesy of Xavier Berenguer, Animatica.

Origin™ and Onyx2™ Theory of Operations Manual
Document Number 007-3439-002

Contents

List of Figures vii

List of Tables ix

About This Guide xi
References and Source Material xiii
Typographical Conventions xiv
Italic xiv
Bold Text xiv
For More Information xiv
Comments and Corrections xiv

1. Overview of the Origin Family Architecture 1
Origin200 5
Origin2000 9
Processor 16
Memory 16
I/O Controllers 17
Hub 17
Directory Memory 17
CrayLink Interconnect 17
XIO and Crossbow (XBOW) 17
What Makes the Origin2000 System Different 18
Scalability and Modularity 20
System Interconnections 21
Interconnection Fabric 21
Crossbar 23

Distributed Shared Address Space (Memory and I/O) 25
Origin2000 Memory Hierarchy 25
System Bandwidth 27

2. Origin Family Boards 31
Node Board 32
Hub ASIC and Interfaces 33
Processors and Cache 33
Distributed Shared-Memory 34
Load/Store Architecture 35
Memory Specifications 36
Memory Blocks, Lines, and Pages 37
Virtual Memory 38
Translation Lookaside Buffer 40
Hits, Misses, and Page Faults 40
Cache-Coherence Protocol 41
Snoopy-Based Coherence 43
Directory-Based Coherence 43
Maintaining Coherence Through Invalidation 44
Why Coherence is Implemented in Hardware 44
Memory Consistency 45
Directory States 45
Sample Read Traversal of Memory 46
Sample Write Traversal of the Memory Hierarchy 46
System Latency 48
Locality: Spatial and Temporal 48
Page Migration and Replication 49
Directory Poisoning 50
I/O 51
Global Real-time Clock 51
Node Board Physical Connections 51

XIO Protocol and Devices 53
Distributed I/O 53
Crossbow Expansion 53
XIO Devices — Widgets 53
Crossbow Configuration 54
Router Board 55
Types of Router Boards 57
SGI Transistor Logic (STL) 57
Connectors 58
Xpress Links 59
Midplane Board 63
BaseIO Board 67
MediaIO (MIO) Board 71
Crosstown Board 73
Origin200 Mother and Daughter Boards 73

3. **Origin Family ASICs** 75
Hub ASIC 77
Hub Interfaces 77
Cache Coherence 79
Static Partitioning of I/O 79
Router ASIC 80
SSD/SSR 82
Link-Level Protocol (LLP) 82
Router Receiver and Sender 82
Routing Table 83
Router Crossbar 83
Crossbow (XBOW) ASIC 84
Bridge ASIC 86
IOC3 ASIC 87
LINC ASIC 87

Glossary 91

Index 103

List of Figures

Figure i Organization of this Manual xii
Figure 1-1 Developmental Path of SGI Multiprocessing Architectures 2
Figure 1-2 Nodes in an Origin2000 System 3
Figure 1-3 Single Datapath Over a Bus 4
Figure 1-4 Multi-dimensional Datapaths Through an Interconnection Fabric 4
Figure 1-5 Origin200 Mother and Daughterboards 6
Figure 1-6 Location of Origin200 PCI slots 7
Figure 1-7 Layout of Origin200 Memory 8
Figure 1-8 Origin2000 Block Diagram 9
Figure 1-9 Block Diagram of a System with 4 Nodes 10
Figure 1-10 Exploded View of the Origin2000 Deskside Chassis 12
Figure 1-11 Front View of Origin2000 Chassis, with Components 13
Figure 1-12 Front View of Origin2000 Chassis, Front Facade Removed 14
Figure 1-13 Rear View of Origin2000 Chassis 15
Figure 1-14 Block Diagram of an Origin2000 System 16
Figure 1-15 Datapaths in an Interconnection Fabric 22
Figure 1-16 Logical Illustration of a Four-by-Four (4 x 4) Crossbar 23
Figure 1-17 Crossbar Operation 24
Figure 1-18 Memory Hierarchy, Based on Relative Latencies and Data Capacities 26
Figure 2-1 Block Diagram of the Node Board 32
Figure 2-2 Horizontal In-Line Memory Module 33
Figure 2-3 Origin2000 Address Space 34
Figure 2-4 Memory Banks and Interleaving 36
Figure 2-5 Cache Lines and Memory Pages 37
Figure 2-6 Allocating Physical Memory to Virtual Processes 38
Figure 2-7 Converting Virtual to Physical Addresses 39

Figure 2-8 Virtual-to-Physical Address Mapping in a TLB Entry 40
Figure 2-9 Exclusive Data 41
Figure 2-10 Shared Data 42
Figure 2-11 Directory-Based Coherence 43
Figure 2-12 Origin2000 Local and Remote Page Access Counters 50
Figure 2-13 Physical View of the Node Board 52
Figure 2-14 Crossbow Connections 54
Figure 2-15 Location of a Router Board in an Origin2000 System 55
Figure 2-16 Physical View of the Router Board 56
Figure 2-17 Routing Board Connectors 58
Figure 2-18 16P System Using Xpress Links 60
Figure 2-19 24P System 61
Figure 2-20 32P System Using Xpress Links 62
Figure 2-21 Physical Location of the Midplane in Deskside Enclosure 63
Figure 2-22 Front View of the Midplane 65
Figure 2-23 Rear View of the Origin2000 Midplane Board 66
Figure 2-24 Logical Location of an BaseIO Board in an Origin2000 System 68
Figure 2-25 BaseIO Board Block Diagram 69
Figure 2-26 Physical Layout of the BaseIO Board 70
Figure 2-27 MIO Board Block Diagram 72
Figure 2-28 Crosstown Link 73
Figure 3-1 ASIC Protocols 75
Figure 3-2 Block Diagram of a Hub ASIC 78
Figure 3-3 Block Diagram of the Router ASIC 81
Figure 3-4 Functional Location of Crossbow ASIC 84
Figure 3-5 Block Diagram of a Crossbow ASIC, Showing Eight Ports Connected to Widgets 85
Figure 3-6 Bridge ASIC 86
Figure 3-7 Block Diagram of LINC ASICs With Bridge ASIC 88

List of Tables

Table 1-1 Comparison of Peak and Sustained Bandwidths 28

Table 1-2 System Bisection Bandwidths 28

Table 1-3 Peripheral Bandwidths 29

About This Guide

This document is a brief and practical orientation to the Origin2000™ system. It is intended for SSEs and customers who need background for Origin2000 installation and diagnostic tasks. The information contained in this document is organized as follows:

- Front matter
- Contents
- List of Figures
- List of Tables
- *About This Guide*, which describes the organization of the manual, references, and typographical conventions
- *Chapter 1*, which gives an overview of the Origin™ architecture
- *Chapter 2*, which gives a board-level description
- *Chapter 3*, which gives an ASIC-level description

The figure on the next page illustrates this organization pictorially.

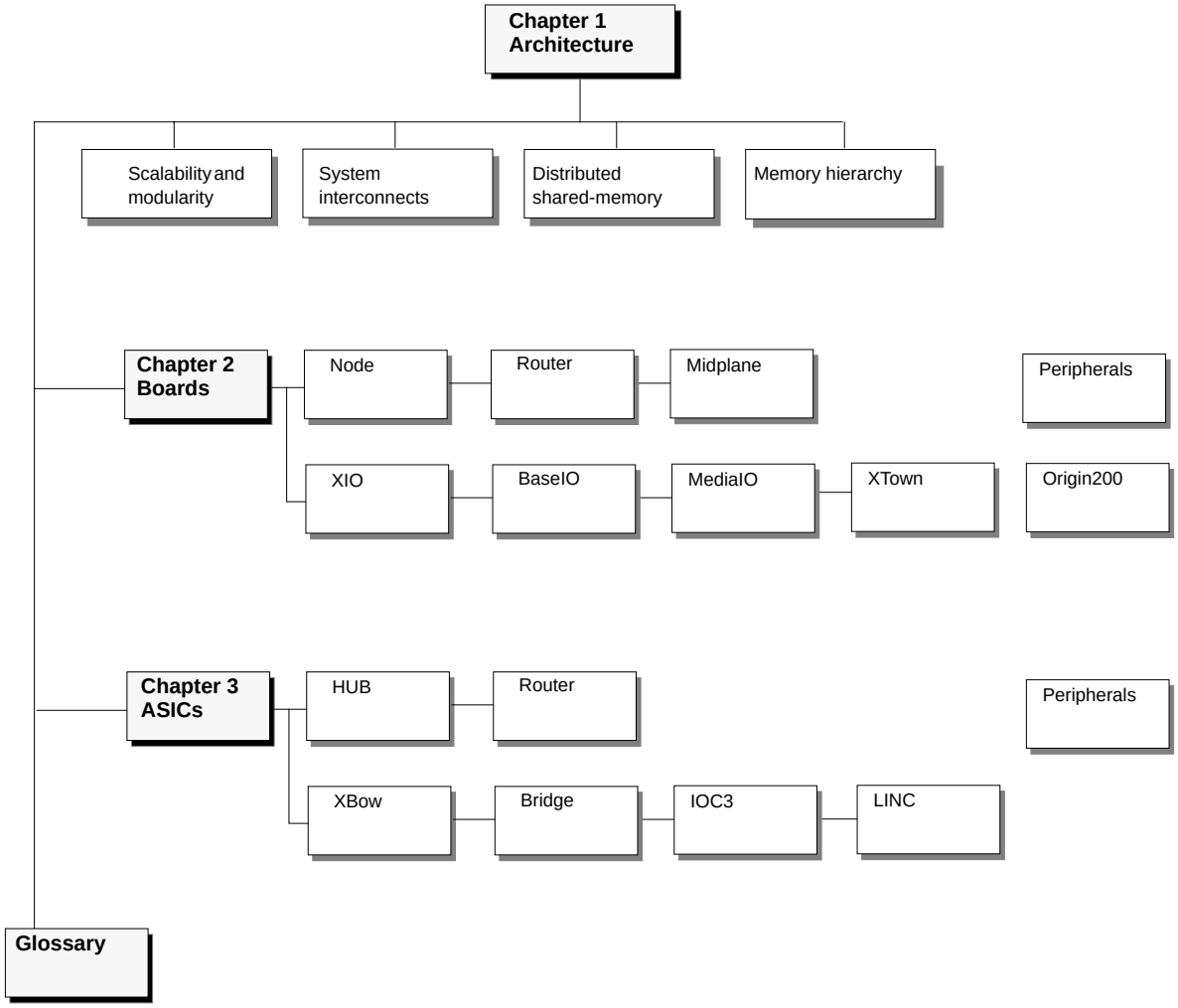

Figure i Organization of this Manual

References and Source Material

Much of the explanatory material in this book was taken from three canons:

- Lenoski, Daniel and Weber, Wolf-Dietrich. *Scalable Shared-Memory Multiprocessing* San Francisco: Morgan Kauffman, 1995.
- Hennessy, John and Patterson David. *Computer Architecture: A Quantitative Approach* San Mateo, California: Morgan Kauffman, 1990.
- Schimmel, Curt. *Unix Systems for Modern Architectures* Menlo Park, California: Addison Wesley, 1994.

James Laudon's *System Specification* and *Cache Coherence Protocol Specification* provided a wealth of information. Both are internal-Silicon Graphics® documents.

The following information was also relevant:

- Kourosh Gharachorloo, Daniel Lenoski, James Laudon, Phillip Gibbons, Anoop Gupta, and John Hennessy. *Memory Consistency and Event Ordering in Scalable Shared-Memory Multiprocessors*. Proceedings of the 17th International Symposium on Computer Architecture, pages 15-26, May 1990.

  `ftp://www-flash.stanford.edu/pub/flash/ISCA90.ps.Z`

- Kourosh Gharachorloo, Anoop Gupta, and John Hennessy. *Revision to Memory Consistency and Event Ordering in Scalable Shared-Memory Multiprocessors*. Technical Report CSL-TR-93-568, Computer Systems Laboratory, Stanford University, April 1993.

  `ftp://www-flash.stanford.edu/pub/flash/ISCA90_rev.ps.Z`

- Daniel Lenoski, James Laudon, Truman Joe, David Nakahira, Luis Stevens, Anoop Gupta, and John Hennessy. *The DASH Prototype: Implementation and Performance*. In Proceedings of the 19th International Symposium on Computer Architecture, pages 92-103, Gold Coast, Australia, May 1992.

  `http://www-flash.stanford.edu/architecture/papers/paperlinks.html`

Thanks also to **Ben Passarelli**, **Rick Bahr**, **Rich Altmaier**, **Ben Fathi**, **Ed Reidenbach**, **Rob Warnock**, **Jim** "Positive-ECL" **Smith**, **Sam Sengupta**, **Dave Parry**, **Robert A. dePeyster**, **Mike Galles**, and **Luis Stevens**.

Finally, thanks to **John Mashey** (mash@mash.sgi.com) for making himself iteratively available during various emergencies.

Typographical Conventions

Italic

- is used for *emphasis*
- is used for *bits*, *fields*, and *registers* important from a software perspective (for instance, *address bits* used by software, *programmable registers*, etc.)

Bold Text

- represents a term that is being **defined**
- is used for **bits** and **fields** which are important from a hardware perspective (for instance **signals** on the backplane, or **register** bits which are not programmable but accessible only to hardware)

For More Information

The following documents provide additional information about the Origin and Onyx2™ systems:

- *Origin and Onyx2 Programmer's Reference Manual*, part number 007-3410-*nnn*
- *IRIX Device Driver Programmer's Guide*, part number 007-0911-*nnn*
- *Origin2000 Deskside Server Owner's Guide*, part number 007-3453-*nnn*
- *Origin2000 Rackmount Owner's Guide*, part number 007-3456-*nnn*
- *Onyx2 Deskside Workstation Owner's Guide*, part number 007-3454-*nnn*
- *Origin200 Owner's Guide*, part number 007-3415-*nnn*

Some of these can be found on the Technical Publications web site,

http://www.sgi.com/Technology/TechPubs/

Comments and Corrections

For comments and corrections, the author can be reached at joeh@sgi.com

Chapter 1

Overview of the Origin Family Architecture

The guide describes the hardware architecture of the Origin family, and its specific implementations:

- The entry-level Origin200™ system, consisting of a maximum of two towers (up to four processors) that can be linked together.
- The deskside/rackmount Origin2000 system, presently consisting of from 1 to 32 processors (with potential future expansion to 128), housed in deskside and rackmount cabinets.

The Origin family is a revolutionary follow-on to the CHALLENGE®-class symmetric multiprocessing (SMP) system. It uses Silicon Graphics' distributed Scalable Shared-memory MultiProcessing architecture, called **S2MP**.

The development path Silicon Graphics' multiprocessor systems is shown in Figure 1-1.

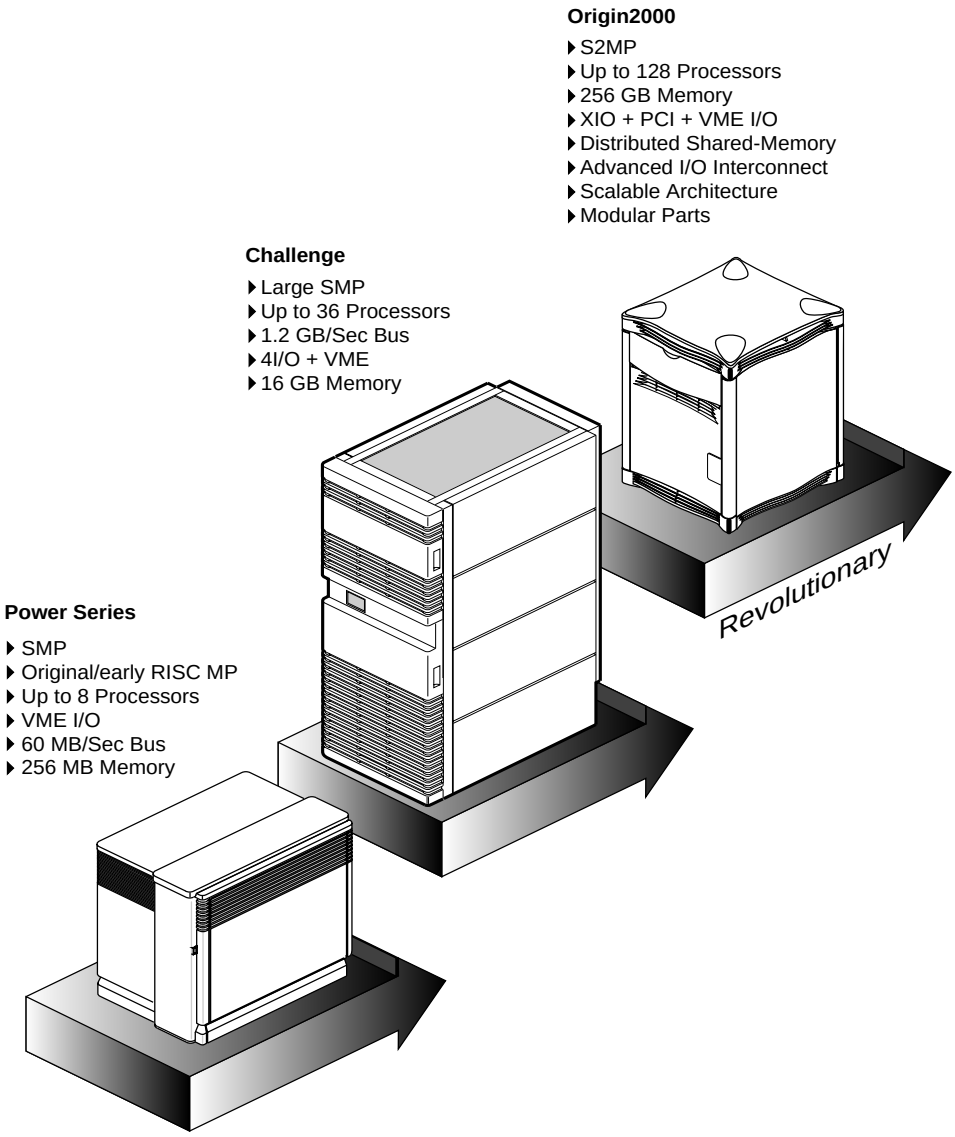

Figure 1-1 Developmental Path of SGI Multiprocessing Architectures

As illustrated in Figure 1-2, Origin2000 is a number of processing nodes linked together by an interconnection fabric. Each processing node contains either one or two processors, a portion of shared memory, a directory for cache coherence, and two interfaces: one that connects to I/O devices and another that links system nodes through the interconnection fabric.

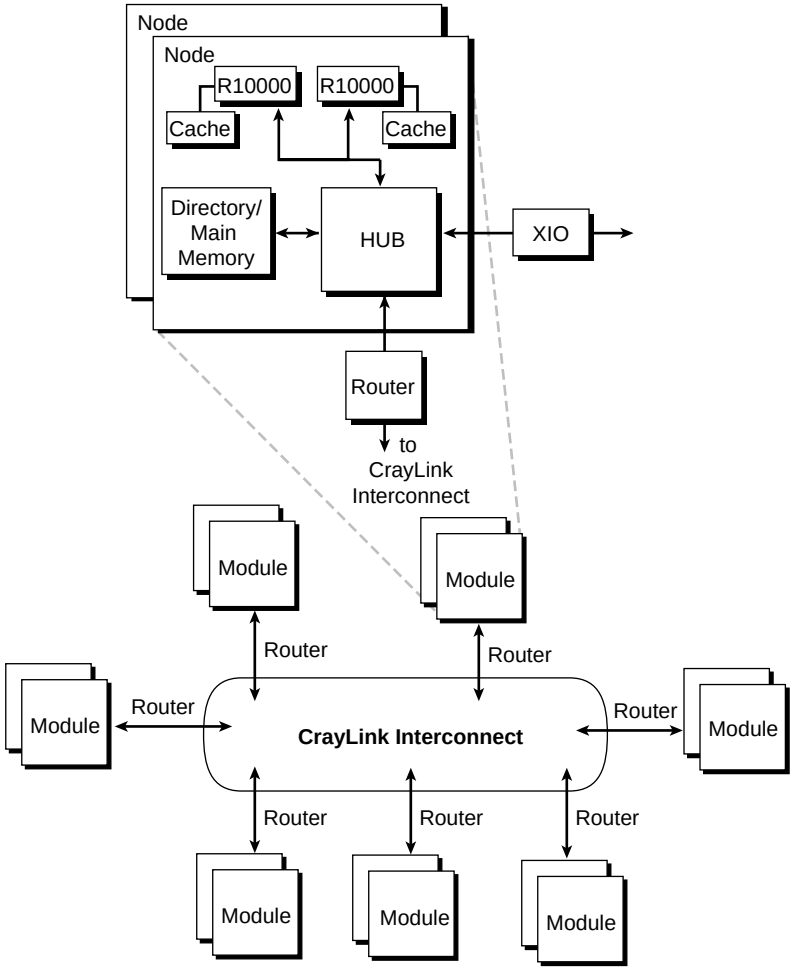

Figure 1-2 Nodes in an Origin2000 System

The interconnection fabric links nodes to each other, but it differs from a bus in several important ways. A bus is a resource that can only be used by one processor at a time. The interconnection fabric is a mesh of multiple, simultaneous, dynamically-allocable — that is, connections are made from processor to processor as they are needed — transactions. This web of connections differs from a bus in the same way that multiple dimensions differ from a single dimension: if a bus is a *one*-dimensional line, then the interconnection fabric is a *multi*-dimensional mesh.

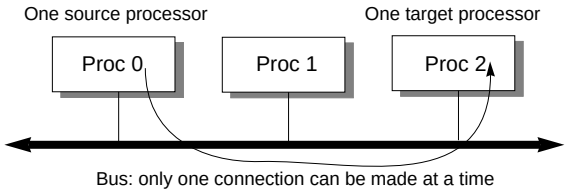

Figure 1-3 Single Datapath Over a Bus

As shown in Figure 1-3, a bus is a shared, common link that multiprocessors must contest for and that only a single processor can use at a time. The interconnection fabric allows many nodes to communicate simultaneously, as shown in Figure 1-4. (Each black box connected to a router ("R") is a node that contains two R10000™ processors.) Paths through the interconnection fabric are constructed as they are needed by Router ASICs, which act as switches.

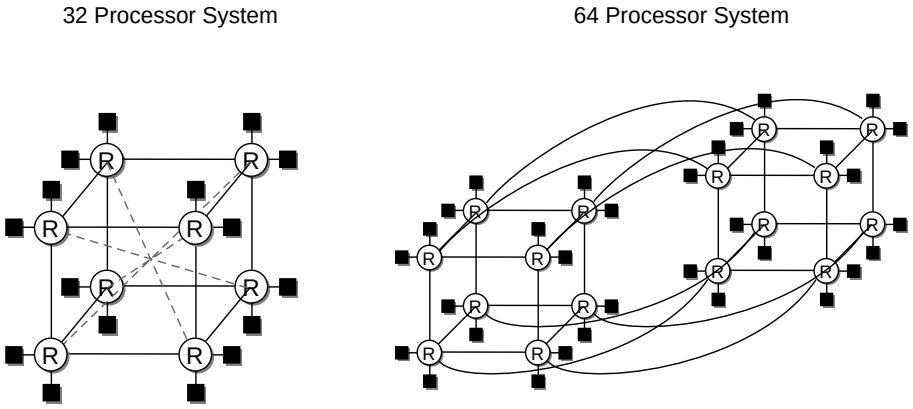

Figure 1-4 Multi-dimensional Datapaths Through an Interconnection Fabric

The Origin2000 system is said to be scalable, because it can range in size from 1 to 128 processors. As you add nodes, you add to and scale the system bandwidth. The Origin2000 is also modular, in that it can be increased in size by adding standard modules to the interconnection fabric. The interconnection fabric is implemented on cables outside of these modules.

The Origin family uses Silicon Graphics' S2MP architecture to distribute shared memory amongst the nodes. This shared memory is accessible to all processors through the interconnection fabric and can be accessed with low latency.

The next sections describe both the Origin200 system and the Origin2000 system.

Origin200

The Origin200 system can consist of one or two towers. The maximum configuration of two towers is connected together by the CrayLink™ interconnection fabric (this is described earlier in this chapter). Each tower has the following:

- a daughterboard, mounted on the motherboard, which holds
  - either one or two R10000 processors
  - 1 or 4 MB of secondary cache for each processor
- a motherboard, which has
  - main memory, ranging from 32 MB to 2 GB
  - SCSI controllers
  - Ethernet controller
- three PCI expansion slots
- a slot for a 5-1/4-inch drive
- slots for five 3-1/2-inch drives
- SCSI, serial, Ethernet ports

Locations of the mother and daughterboards are shown in Figure 1-5 and the locations of the PCI slots are shown in Figure 1-6. Figure 1-7 shows the layout of the memory DIMMs on the motherboard.

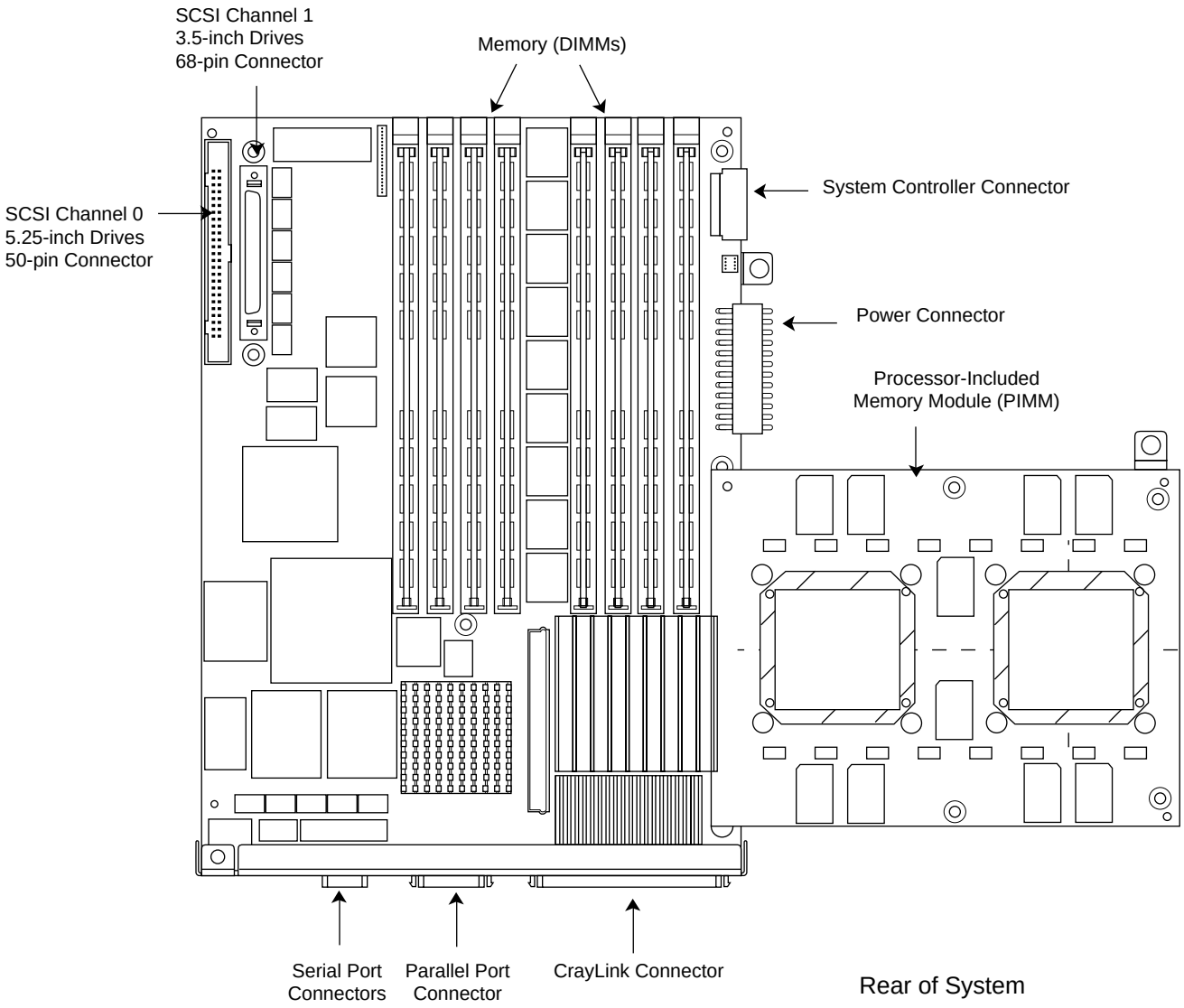

Figure 1-5 Origin200 Mother and Daughterboards

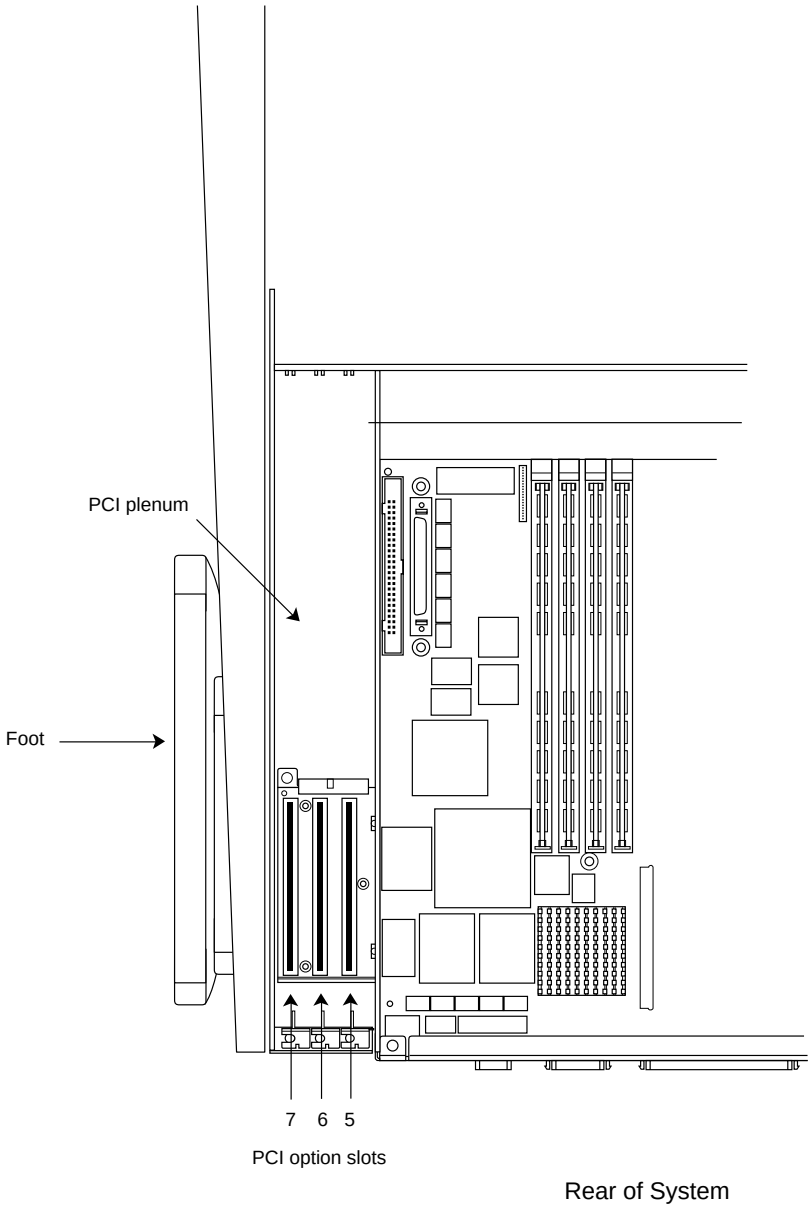

Figure 1-6 Location of Origin200 PCI slots

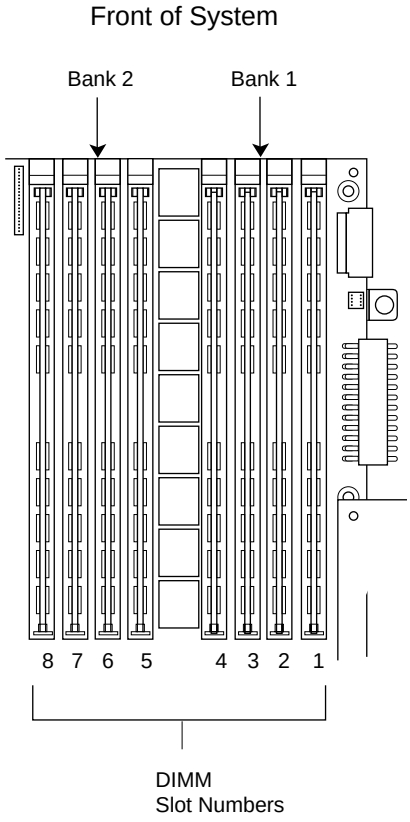

Figure 1-7 Layout of Origin200 Memory

Origin2000

Figure 1-8 is a block diagram of an Origin2000 system showing the central Node board, which can be viewed as a system controller from which all other system components radiate.

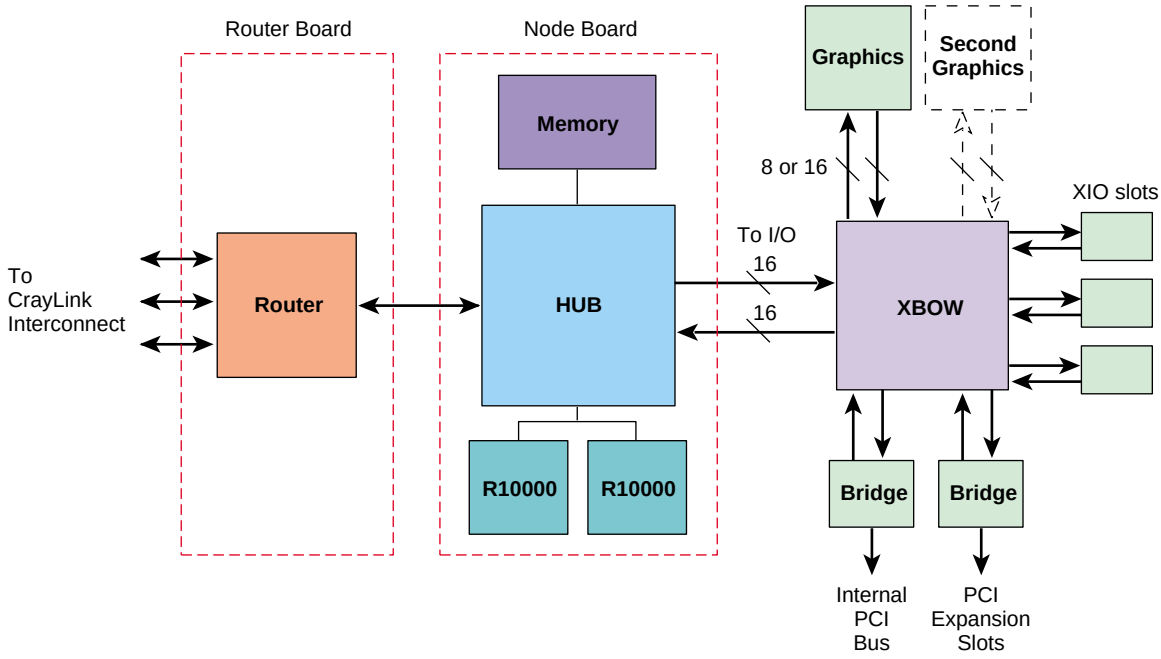

Figure 1-8 Origin2000 Block Diagram

Figure 1-9 is a block diagram of a system with four Node boards; Nodes 1 and 3 connect to Crossbow (XBOW) 1, and Nodes 2 and 4 connect to Crossbow 2. Crossbow 1 connects to XIO boards 1 through 6 and Crossbow 2 connects to XIO boards 7 through 12.

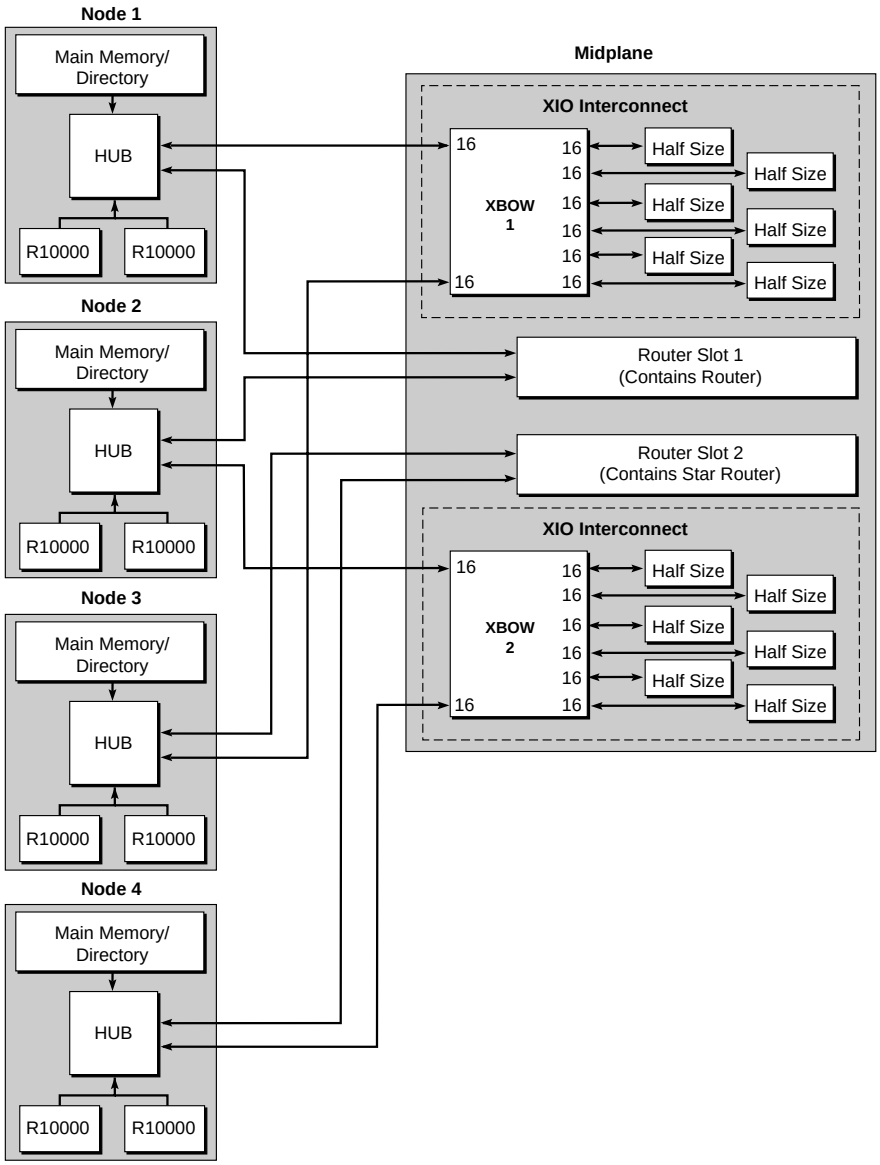

Figure 1-9 Block Diagram of a System with 4 Nodes

An Origin2000 system has the following components:

- R10000 processor(s)
- memory
- I/O controllers
- distributed memory controller (Hub ASIC)
- directory memory for cache coherence
- CrayLink Interconnect
- XIO and Crossbow interfaces

These are linked as shown in Figure 1-8 and Figure 1-9, and all are described in this section.

An exploded view of an Origin2000 deskside system is shown in Figure 1-10. A front view of the enclosed module is shown in Figure 1-11 and a front view with the facade removed is shown in Figure 1-12. A rear view of the deskside chassis, showing the Node and XIO board locations, is shown in Figure 1-13, and Figure 1-14 shows a block diagram of a basic Origin2000 system with a single node connected to XIO and the CrayLink Interconnect.

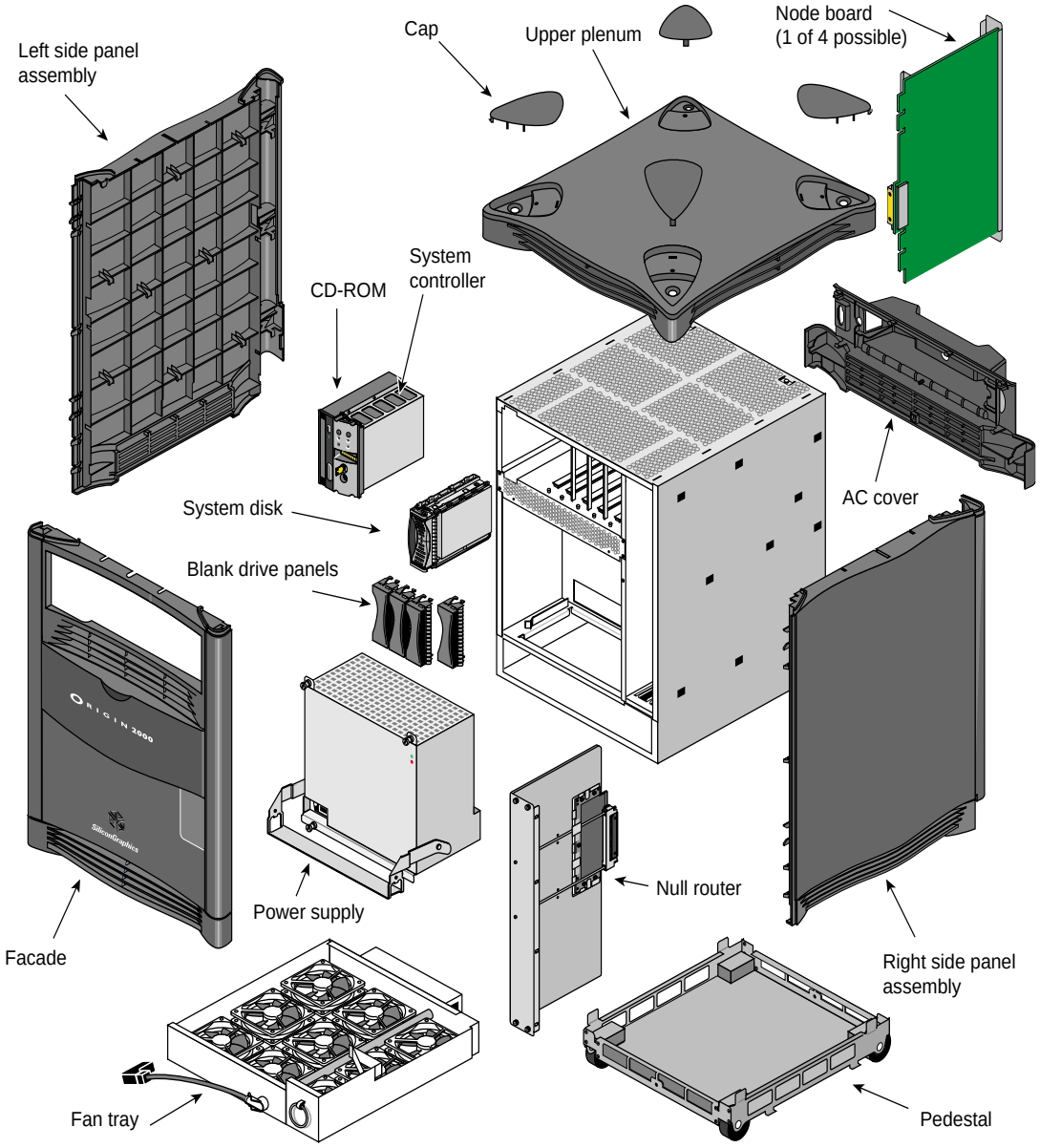

Figure 1-10 Exploded View of the Origin2000 Deskside Chassis

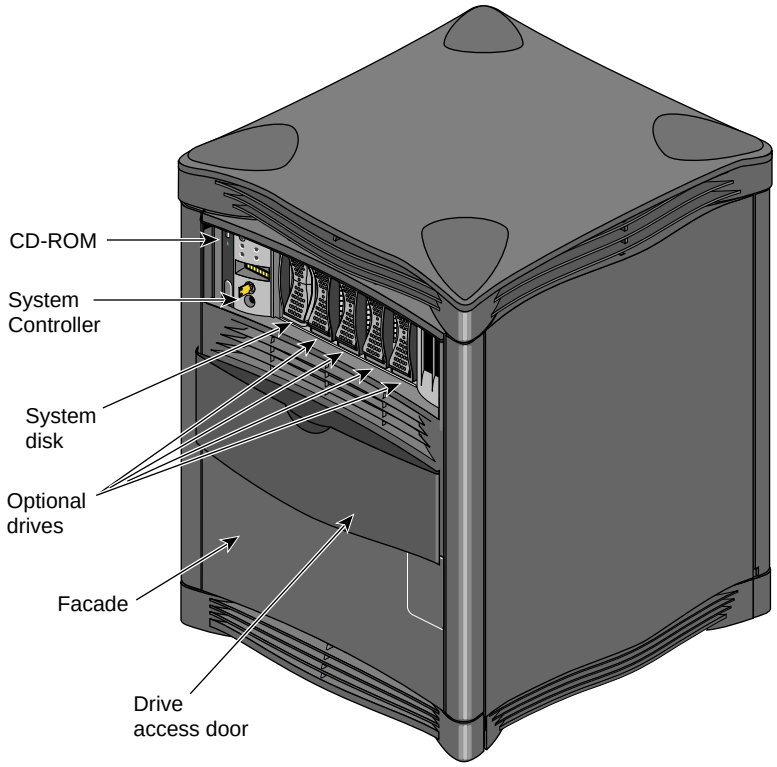

Figure 1-11 Front View of Origin2000 Chassis, with Components

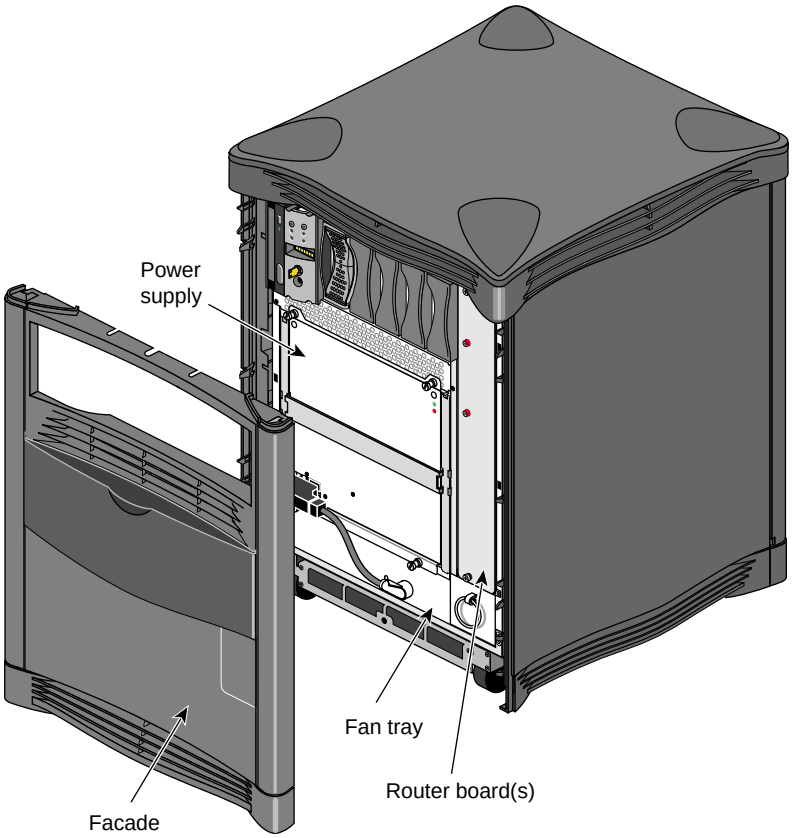

Figure 1-12 Front View of Origin2000 Chassis, Front Facade Removed

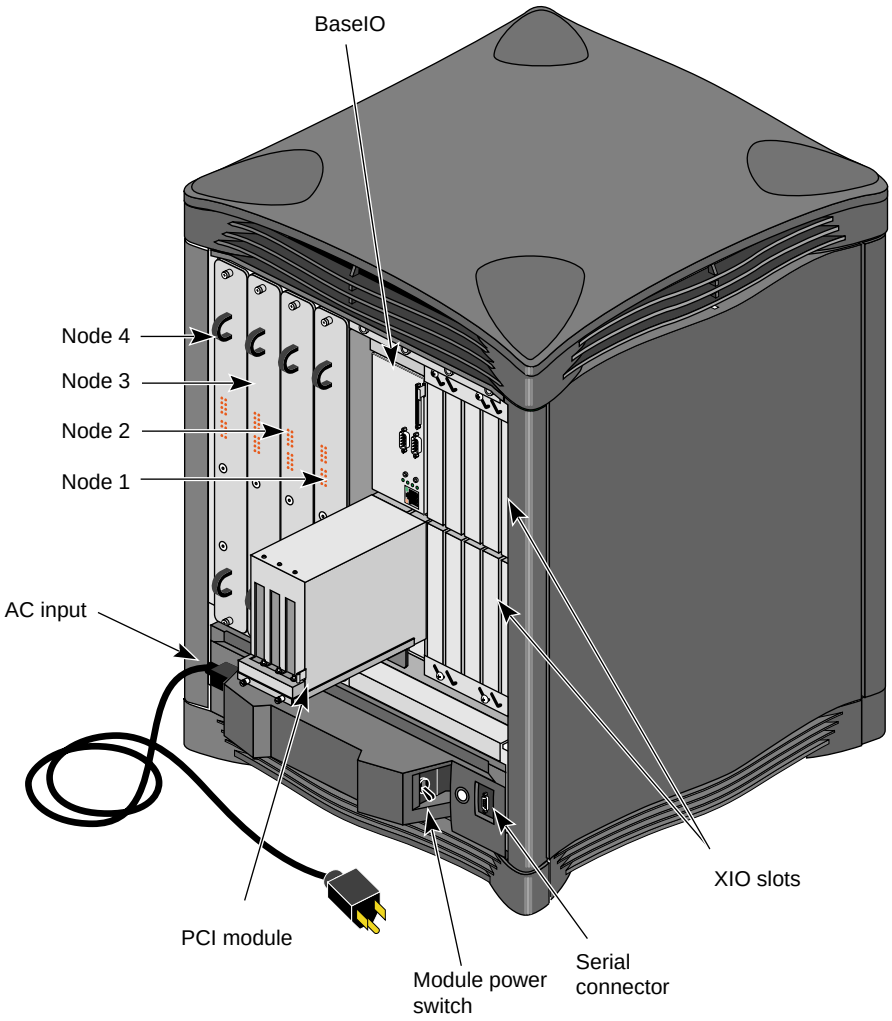

Figure 1-13 Rear View of Origin2000 Chassis

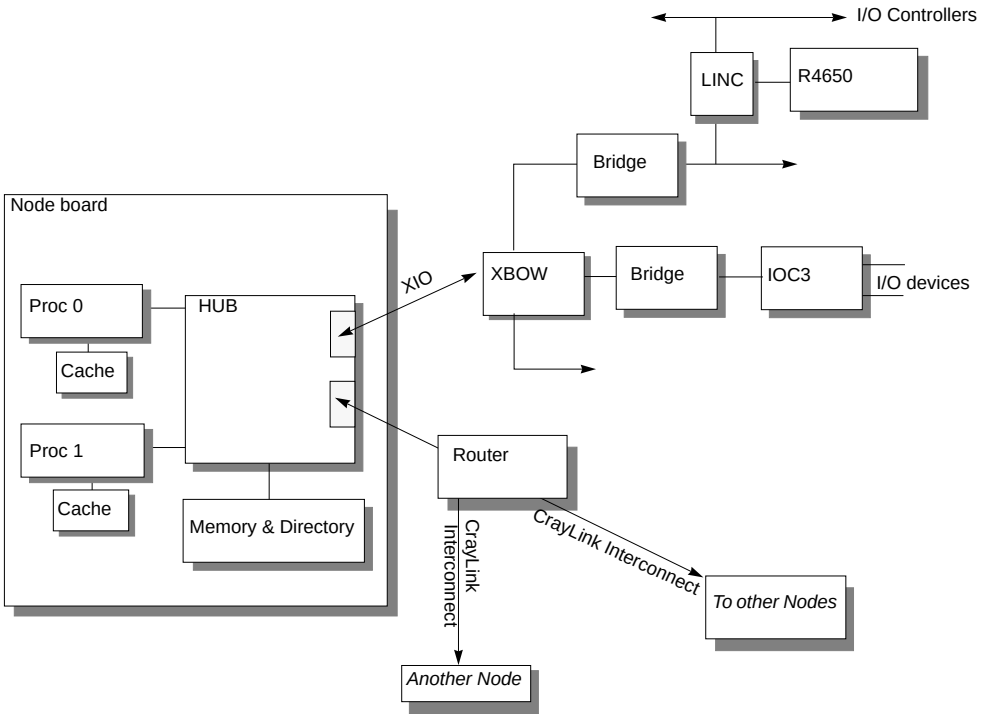

Figure 1-14 Block Diagram of an Origin2000 System

Processor

Origin2000 system uses the MIPS® R10000, a high-performance 64-bit superscalar processor which supports dynamic scheduling. Some of the important attributes of the R10000 are its large memory address space, together with a capacity for heavy overlapping of memory transactions — up to twelve per processor in Origin2000.

Memory

Each Node board added to Origin2000 is another independent bank of memory, and each bank is capable of supporting up to 4 GB of memory. Up to 64 nodes can be configured in a system, which implies a maximum memory capacity of 256 GB.

I/O Controllers

Origin2000 supports a number of high-speed I/O interfaces, including Fast, Wide SCSI, Fibrechannel, 100BASE-Tx, ATM, and HIPPI-Serial. Internally, these controllers are added through XIO cards, which have an embedded PCI-32 or PCI-64 bus. Thus, in Origin2000 I/O performance is added one bus at a time.

Hub

This ASIC is the distributed shared-memory controller. It is responsible for providing all of the processors and I/O devices a transparent access to all of distributed memory in a cache-coherent manner

Directory Memory

This supplementary memory is controlled by the Hub. The directory keeps information about the cache status of all memory within its node. This status information is used to provide scalable cache coherence, and to migrate data to a node that accesses it more frequently than the present node.

CrayLink Interconnect

This is a collection of very high speed links and routers that is responsible for tying together the set of hubs that make up the system. The important attributes of CrayLink Interconnect are its low latency, scalable bandwidth, modularity, and fault tolerance.

XIO and Crossbow (XBOW)

These are the internal I/O interfaces originating in each Hub and terminating on the targeted I/O controller. XIO uses the same physical link technology as CrayLink Interconnect, but uses a protocol optimized for I/O traffic. The Crossbow ASIC is a crossbar routing chip responsible for connecting two nodes to up to six I/O controllers.

What Makes the Origin2000 System Different

The following characteristics make Origin2000 different from previous system architectures (the terms in italics are described in more detail throughout the remainder of this chapter):

- Origin2000 is *scalable.*
- Origin2000 is *modular.*
- Origin2000 uses an *interconnection fabric* to link system nodes and internal *crossbars* within the system ASICs (Hub, Router, Crossbow).
- Origin2000 has *distributed shared-memory* and *distributed shared-I/O.*
- Origin2000 shared memory is kept cache coherent using directories and a *directory-based cache coherence protocol.*
- Origin2000 uses *page migration and replication* to improve memory latency.

See "Scalability and Modularity"

Scalability. Origin2000 is easily scaled by linking nodes together over an interconnection fabric, and system bandwidth scales linearly with an increase in the number of processors and the associated switching fabric. This means Origin2000 can have a low entry cost, since you can build a system upward from an inexpensive configuration.

In contrast, POWER CHALLENGE™ is only scalable in the amount of its processing and I/O power. The Everest interconnect is the E-bus, which has a fixed bandwidth and is the same size from entry-level to high-end.

See "Scalability and Modularity"

Modularity. A system is comprised of standard processing nodes. Each node contains processor(s), memory, a directory for cache coherence, an I/O interface, and a system interconnection. Node boards are placed in both the Origin200 and Origin2000 systems, although they are not identical.

Due to its bus-based design, CHALLENGE is not as modular; there is a fixed number of slots in each deskside or rack system, and this number cannot be changed.

See "System Interconnections"

System interconnections. Origin2000 uses an interconnection fabric and crossbars. 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 — the Crossbow, the Router, and the Hub. Crossbars dynamically link ASIC input ports with their output ports.

In CHALLENGE, processors access memory and I/O interfaces over a shared system bus (E-bus) that has a fixed size and a fixed bandwidth.

See "Distributed Shared Address Space (Memory and I/O)"

Distributed shared-memory (DSM) and I/O. Origin2000 memory is physically dispersed throughout the system for faster processor access. Page migration hardware moves data into memory closer to a processor that frequently uses it. This page migration scheme reduces memory **latency** — the time it takes to retrieve data from memory. Although main memory is distributed, it is universally accessible and shared between all the processors in the system. Similarly, I/O devices are distributed among the nodes, and each device is accessible to every processor in the system.

CHALLENGE has shared memory, but its memory is concentrated, not distributed, and CHALLENGE does not distribute I/O. All I/O accesses, and those memory accesses not satisfied by the cache, incur extra latencies when traversing the E-bus.

See Chapter 2

Directory-based cache coherence. Origin2000 uses caches to reduce memory latency. Cache coherence is supported by a hardware directory that is distributed among the nodes along with main memory. Cache coherence is applied across the entire system and all memory. In a snoopy protocol, every cache-line invalidation must be broadcast to all CPUs in the system, whether the CPU has a copy of the cache line or not. In contrast, a directory protocol relies on point-to-point messages that are only sent those CPUs actually using the cache line. This removes the scalability problems inherent in the snoopy coherence scheme used by bus-based systems such as Everest. A directory-based protocol is preferable to snooping since it reduces the amount of coherence traffic that must be sent throughout the system.

CHALLENGE uses a snoopy coherence protocol.

See Chapter 2

Page migration and replication. To provide better performance by reducing the amount of remote memory traffic, Origin2000 uses a process called **page migration**. Page migration moves data that is often used by a processor into memory close to that processor.

CHALLENGE does not support page migration.

Scalability and Modularity

Origin2000 scalability and modularity allow one to start with a small system and incrementally add modules to make the system as large as needed. An entry-level Origin20000 module can hold from one to four MIPS R10000 processors, and a Origin2000 deskside module can hold from one to eight R10000 processors. A series of these deskside modules can be mounted in racks, scaling the system up to the following maximum configuration:

- 128 processors
- 256 GB of memory
- 64 I/O interfaces with 192 I/O controllers (or 184 XIO and 24 PCI-64)
- 128 3.5-inch Ultra-SCSI devices and 16 6.25-inch devices

As one adds nodes to the interconnection fabric, bandwidth and performance scale linearly without significantly impacting system latencies. This is a result of the following design decisions:

- replacing the fixed-size, fixed-bandwidth bus of CHALLENGE with the scalable interconnection fabric whose **bisection bandwidth** (the bandwidth through the center of CrayLink Interconnect) scales linearly with the number of nodes in the system
- reducing system latencies by replacing the centrally-located main memory of CHALLENGE with the tightly-integrated but distributed shared-memory S2MP architecture of Origin2000.

System Interconnections

Origin2000 replaces CHALLENGE's shared, fixed-bandwidth bus with the following:

See "Interconnection Fabric"

- a scalable **interconnection fabric**, in which processing nodes are linked by a set of routers

See "Crossbar"

- **crossbar switches**, which implement the interconnection fabric. Crossbars are located in the following places:
  - within the Crossbow ASIC connecting the I/O interfaces to the nodes
  - within the Router ASIC forming the interconnection fabric itself,
  - within the Hub ASIC which interconnects the processors, memory, I/O, and interconnection fabric interfaces within each node.

  These internal crossbars maximize the throughput of the major system components and concurrent operations.

Interconnection Fabric

Origin2000 nodes are connected by an interconnection 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 important ways:

- The interconnection fabric is a mesh of multiple point-to-point links connected by the routing switches. 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 as limited by contention, while a bus must be contested for through arbitration.
- More routers and links are added as nodes are added, increasing the interconnection fabric's bandwidth. A shared bus has a fixed bandwidth that is not scalable.
- The topology of the CrayLink Interconnect is such that the bisection 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 fabric 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.

Earlier in this chapter, Figure 1-3 and Figure 1-4 showed how an interconnection fabric differs from an ordinary shared bus. Figure 1-15 amplifies this difference by illustrating an 8-node hypercube with its multiple datapaths. Simultaneously, R1 can communicate with R0, R2 to R3, R4 to R6, and R5 to R7, all without having to interface with any other node.

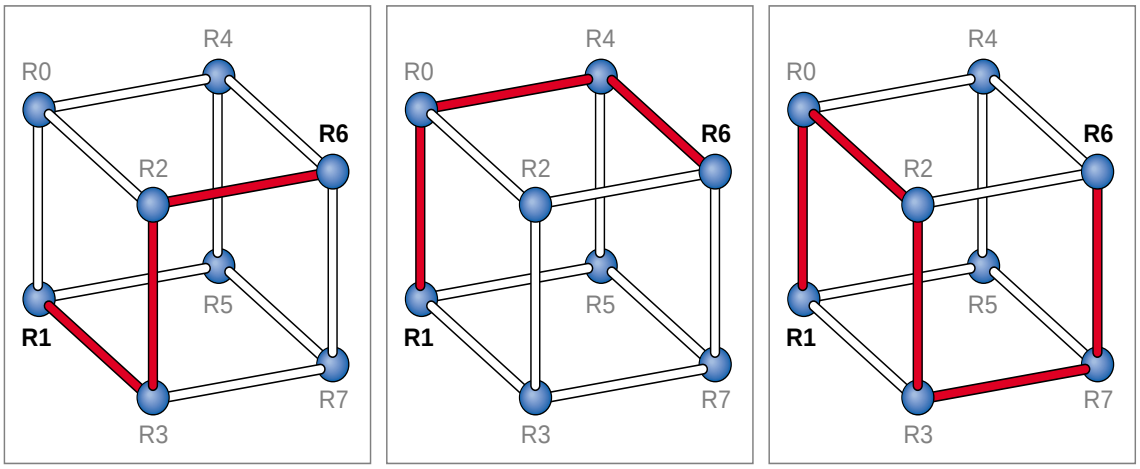

Figure 1-15 Datapaths in an Interconnection Fabric

Crossbar

Several of the ASICs (Hub, Router, and Crossbow) use a crossbar for linking on-chip inputs with on-chip output interfaces. For instance, an 8-way crossbar is used on the Crossbow ASIC; this crossbar creates direct point-to-point links between one or more nodes and multiple I/O devices. The crossbar switch also allows peer-to-peer communication, in which one I/O device can speak directly to another I/O device.

The Router ASIC uses a similar 6-way crossbar to link its six ports with the interconnection fabric, and the Hub ASIC links its four interfaces with a crossbar. A logical diagram of a 4-way (also referred to as four-by-four, or 4 x 4) crossbar is given in Figure 1-16; note that each output is determined by multiplexing the four inputs.

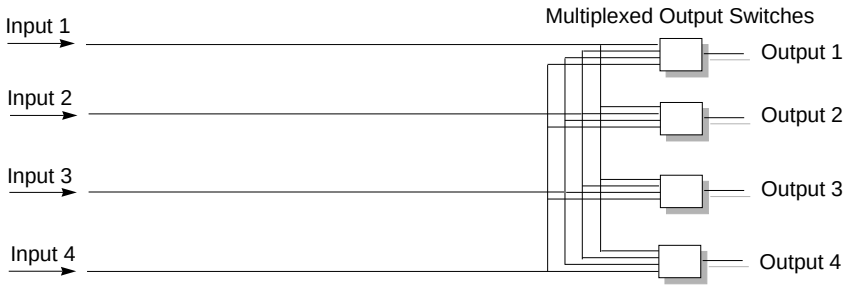

Figure 1-16 Logical Illustration of a Four-by-Four (4 x 4) Crossbar

Figure 1-17 shows a 6-way crossbar at work. In this example, the crossbar connects six ports, and each port has an input (**I**) and an output (**O**) buffer for flow control. Since there must be an output for every input, the six ports can be connected as six independent, parallel paths. The crossbar connections are shown at two clock intervals: `Time=n`, and `Time=n+1`.

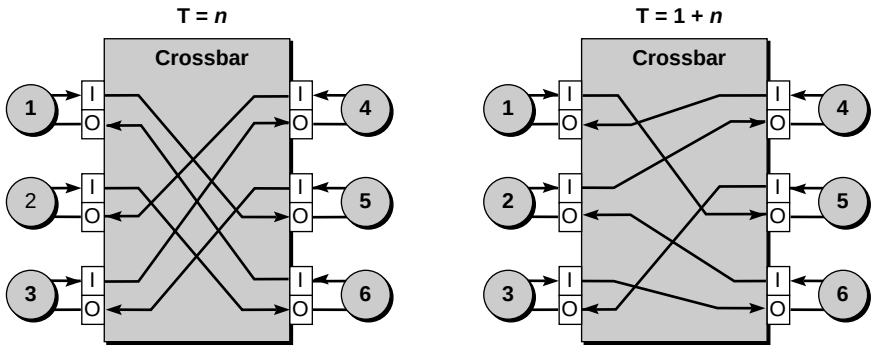

Figure 1-17 Crossbar Operation

At clock `T=n`, the ports independently make the following parallel connections:

- from port 1 to port 5
- from port 2 to port 6
- from port 3 to port 4
- from port 4 to port 2
- from port 5 to port 3
- from port 6 to port 1

Figure 1-17 shows the source (**I**nput) and target (**O**utput) for each connection, and arrows indicate the direction of flow. When a connection is active, its source and target are not available for any other connection over the crossbar.

At the next clock, `T=n+1`, the ports independently reconfigure themselves[1] into six new data links: 1-to-5, 2-to-4, 3-to-6, 4-to-1, 5-to-3 and 6-to-2. At clock intervals, the ports continue making new connections as needed. Connection decisions are based on algorithms that take into account flow control, routing, and arbitration.

[1] Except in the Router, where global considerations are taken into account.

Distributed Shared Address Space (Memory and I/O)

Origin2000 memory is located in a single shared address space. Memory within this space is distributed amongst all the processors, and is accessible over the interconnection fabric. This differs from an CHALLENGE-class system, in which memory is centrally located on and only accessible over a single shared bus. By distributing Origin2000's memory among processors memory latency is reduced: accessing memory near to a processor take less time than accessing remote memory. Although physically distributed, main memory is available to all processors.

I/O devices are also distributed within a shared address space; every I/O device is universally accessible throughout the system.

Origin2000 Memory Hierarchy

Memory in Origin2000 is organized into the following hierarchy:

registers

- At the top, and closest to the processor making the memory request, are the **processor registers**. Since they are physically on the chip they have the lowest latency — that is, they have the fastest access times. In Figure 1-18, these are on the processor labelled P0.

cache

- The next level of memory hierarchy is labelled **cache**. In Figure 1-18, these are the primary and secondary caches located on P0. Aside from the registers, caches have the lowest latency in Origin2000, since they are also on the R10000 chip (primary cache) or tightly-coupled to its processor on a daughterboard (secondary cache).

home memory

- The next level of memory hierarchy is called **home memory**, which can be either local or remote. The access is **local** if the address of the memory reference is to address space on the same node as the processor. The access is **remote** if the address of the memory reference is to address space on another node. In Figure 1-18, home memory is the block of main memory on Node 1, which means it is local to Processor 0.

remote cache

- The next level of memory hierarchy consists of the **remote caches** that may be holding copies of a given memory block. If the requesting processor is writing, these copies must be invalidated. If the processor is reading, this level exists if another processor has the most up-to-date copy of the requested location. In Figure 1-18, remote cache is represented by the blocks labelled "cache" on Nodes 2 and 3.

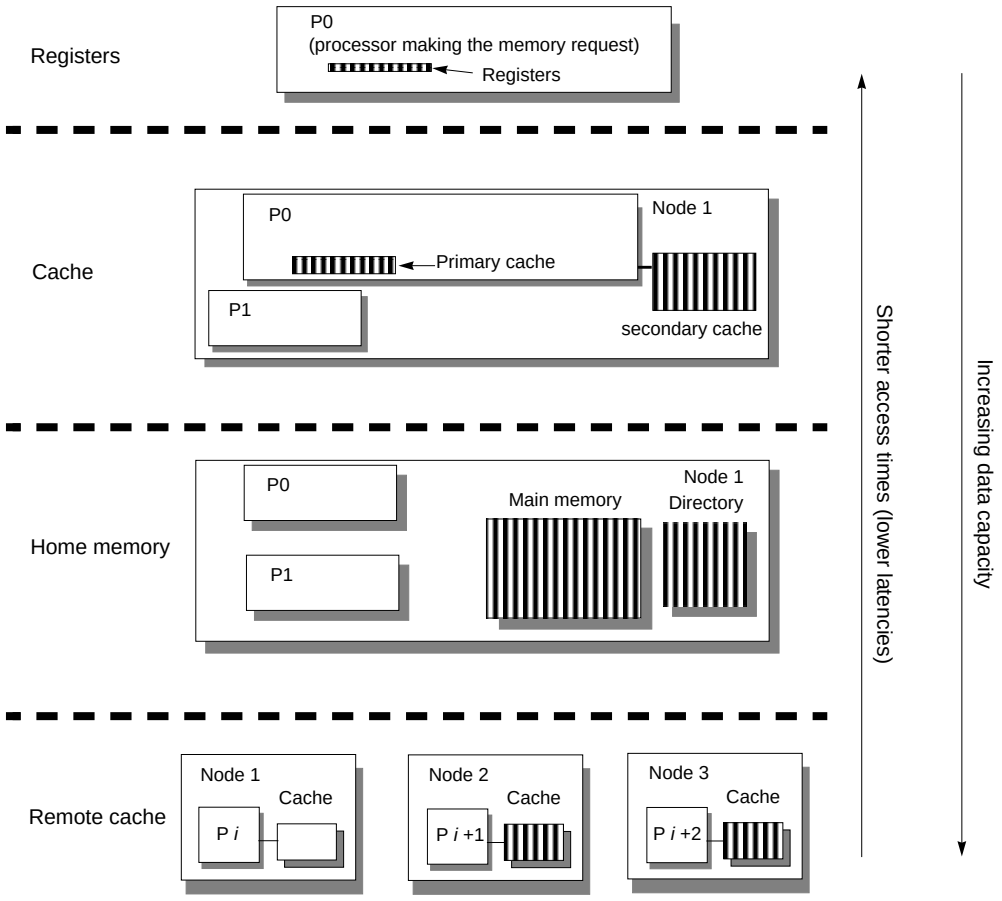

Figure 1-18 Memory Hierarchy, Based on Relative Latencies and Data Capacities

Caches are used to reduce the amount of time it takes to access memory — also known as a memory's **latency** — by moving faster memory physically close to, or even onto, the processor. This faster memory is generally some version of static RAM, or SRAM.

The DSM structure of Origin2000 also creates the notion of local memory. This memory is close to the processor and has reduced latency compared to bus-based systems, where all memory must be accessed through a shared bus.

While data only exists in either local or remote memory, copies of the data can exist in various processor caches. Keeping these copies consistent is the responsibility of the logic of the various hubs. This logic is collectively referred to as a *cache-coherence protocol*, described in Chapter 2.

System Bandwidth

Three types of bandwidth are cited in this manual:

- Peak bandwidth, which is a theoretical number derived by multiplying the clock rate at the interface by the data width of the interface.
- Sustained bandwidth, which is derived by subtracting the packet header and any other immediate overhead from the peak bandwidth. This best-case figure, sometimes called Peak Payload bandwidth, does not take into account contention and other variable effects.
- Bisection bandwidth, which derived by dividing the interconnection fabric in half, and measuring the data rate across this divide. This figure is useful for measuring data rates when the data is not optimally placed.

Table 1-1 gives a comparison between peak and sustained data bandwidths at the Crosstalk and interconnection fabric interfaces of the Hub.

Table 1-1 Comparison of Peak and Sustained Bandwidths

Interface at Hub	Half/Full Duplex	Peak Bandwidth, per second	Sustained Bandwidth per second
Processor	Unidirectional	800 MB	
Memory	Unidirectional	800 MB	
Interconnection Fabric	Full duplex	1.6 GB	1.28 GB
	Half duplex	800 MB	711 MB
Crosstalk	8-bit Full duplex	800 MB	640 MB
	8-bit Half duplex	400 MB	355 MB
	16-bit Full duplex	1.6 GB	1.28 GB
	16-bit Half duplex	800 MB	711 MB

Table 1-2 lists the bisection bandwidths of various Origin2000 configurations both with and without Xpress Links.

Table 1-2 System Bisection Bandwidths

System Size (number of CPUs)	Sustained Bisection Bandwidth without Xpress Links [Peak BW]	Sustained Bisection Bandwidth with Xpress Links [Peak BW]
8	1.28 GB per second [1.6 GB]	2.56 GB per second [3.2 GB][a]
16	2.56 GB per second [3.2 GB]	5.12 GB per second [6.4 GB]
32	5.12 GB per second [6.4 GB]	10.2 GB per second [12.8 GB]
64	10.2 GB per second [12.8 GB]	N/A
128	20.5 GB per second [25.6 GB]	N/A

a. Using a Star Router

Table 1-3 lists the bandwidths of Ultra SCSI and FibreChannel devices.

Table 1-3 Peripheral Bandwidths

Peripheral	Bandwidth
Ultra SCSI	40 MB per second
FibreChannel	100 MB per second

Chapter 2

Origin Family Boards

This chapter contains a description of the following boards and protocol:

- Node board (Origin2000)
- XIO protocol (Origin2000)
- Router board (Origin2000)
- Midplane board (Origin2000)
- BaseIO board (Origin2000)
- MediaIO board (Origin2000)
- Crosstown board (Origin2000)
- Motherboard (Origin200)
- Daughter card (Origin 200)

Node Board

The basic building block of an Origin2000 system is the **Node board**, which plugs into the rear card cage of a deskside enclosure. The Node board contains the Hub ASIC with interfaces to the processor(s), memory, I/O, and CrayLink Interconnect.

Figure 2-1 shows a block diagram of the Node board, with its central Hub ASIC. The bandwidth figures represent peak bandwidth each way through the interface.

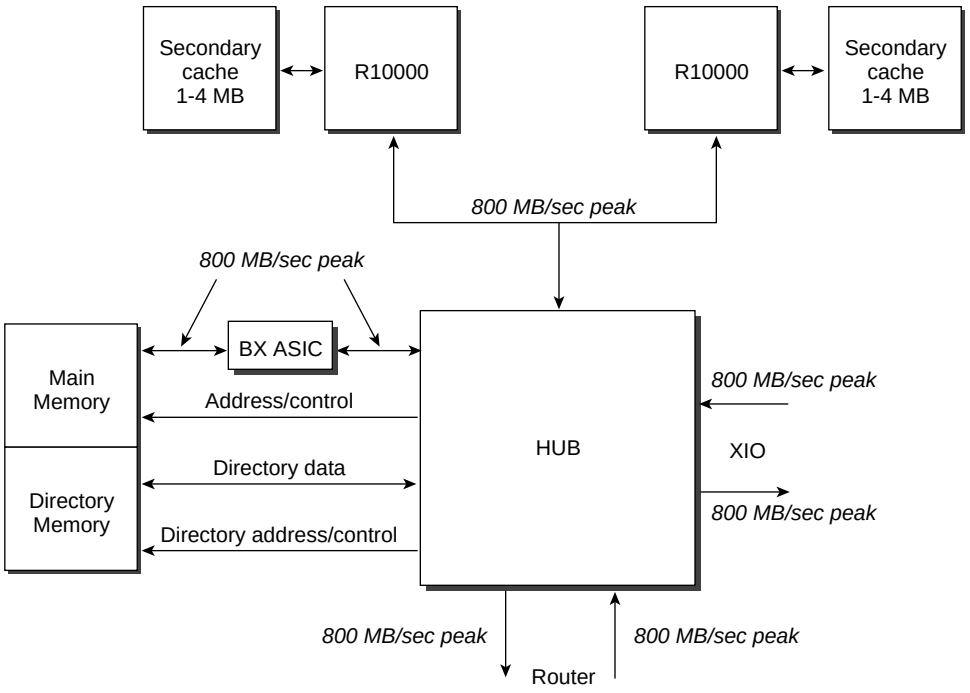

Figure 2-1 Block Diagram of the Node Board

Hub ASIC and Interfaces

The Node board has a central Hub ASIC, to which can be connected:

- either one or two processors
- main memory, and its associated directory memory
- the system interconnection fabric, through a dedicated Router port
- I/O crossbar interconnect, through a dedicated XIO port (single port), or Crossbow ASIC (eight ports)

Bandwidths are given in Chapter 1.

Processors and Cache

The Origin2000 system uses MIPS RISC R10000 64-bit CPU running at 195 MHz. The R10000 has separate 32-KB on-chip set associative primary instruction and data caches. Each CPU has a 1 MB or 4 MB set-associative secondary cache.

Each Node board (also referred to in this text as a "node") is capable of supporting either one or two R10000 processors. In a deskside or rackmounted configuration, each processor is mounted on a **HIMM** (horizontal in-line memory module) together with its primary cache, and either 1 or 4 MB or secondary cache, as shown in Figure 2-2.

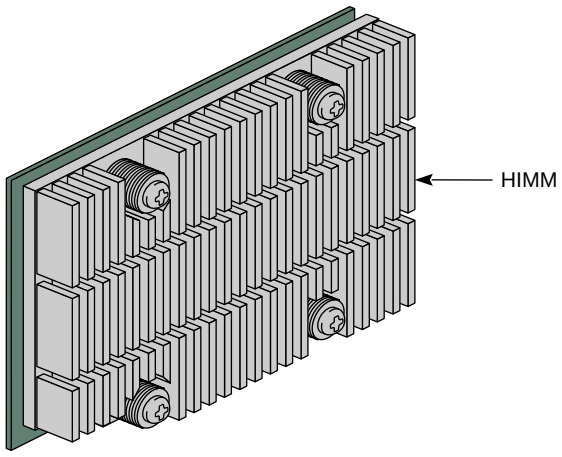

Figure 2-2 Horizontal In-Line Memory Module

Distributed Shared-Memory

As described in Chapter 1, Origin2000 systems use **distributed shared-memory** (DSM). With DSM, main memory is partitioned among processors but is accessible to and shared by all of the processors. Origin2000 divides main memory into two classes: local and remote. Memory on the same node as the processor is labelled **local**, with all other memory in the system labelled **remote**. Despite this distribution, all memory remains globally addressable.

To a processor, main memory appears as a single addressable space containing many blocks, or pages. Each node is allotted a static portion of the address space—which means there is a gap if a node is removed. Figure 2-3 shows an address space in which each node is allocated 4 GB of address space, and Node 1 is removed, leaving a hole from address space 4G to 8G.

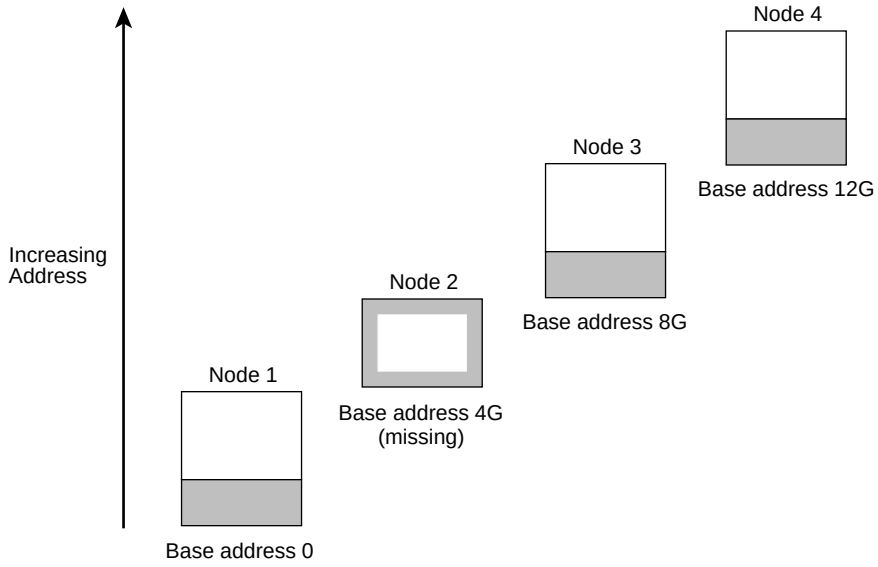

Figure 2-3 Origin2000 Address Space

Main and directory memory are implemented using Synchronous DRAM (SDRAM) parts mounted on dual in-line memory modules (**DIMMs**). Each Node board has main memory that ranges in 8 increments, using pairs of DIMMs. For configurations up to 32 processors (**32P**), directory memory is included in the main memory DIMMs; for configurations larger than 32P, extended directory memory must be added in separate slots, as shown in Figure 2-13.

Memory DIMMs come in a range of sizes. When using 16-Mb parts, each node can support up to 1 GB of memory (in increments of 64 or 128 MB). When using 64-Mb parts, each node can support up to 4 GB of memory (in increments of 512 MB). DIMM increments can be intermixed, as long as increments are made in pairs of DIMMs.

Load/Store Architecture

The R10000 RISC processor uses a load/store architecture. This means data is loaded from memory into processor registers before the processor operates on the data. When the processor has finished, the data is stored back into memory. Only load and store instructions directly access main memory.

This differs from a CISC system, which can have instructions operate on data while it is still in memory. Since memory operations take longer than register operations—and there are typically several memory operations, such as calculating the addresses of the operands, reading the operands from memory, calculating and storing the result—they can negatively impact system latency.

Memory Specifications

Each Node board has 16 memory slots and 8 directory slots. A memory slot holds two memory DIMMs, and the single optional directory DIMM slot is used with an extended directory. These two main memory DIMMs together with the single optional directory DIMM are referred to as a DIMM **bank**.

The DIMM datapath is 144 bits wide—128 bits of data and 16 bits of ECC. In a regular directory, sufficient for 32 processors, a DIMM bank also contains 16 bits of directory memory. For systems that have more than 32 processors, additional directory DIMM must be added to provide the extra 32 bits for an extended directory.

As shown in Figure 2-4, each DIMM bank has two physical banks of SDRAM, and each SDRAM has two logical banks. The four logical banks on a DIMM are interleaved at the 4 KB boundaries, and the 16K page used by Origin2000 spans all four banks on the DIMM.

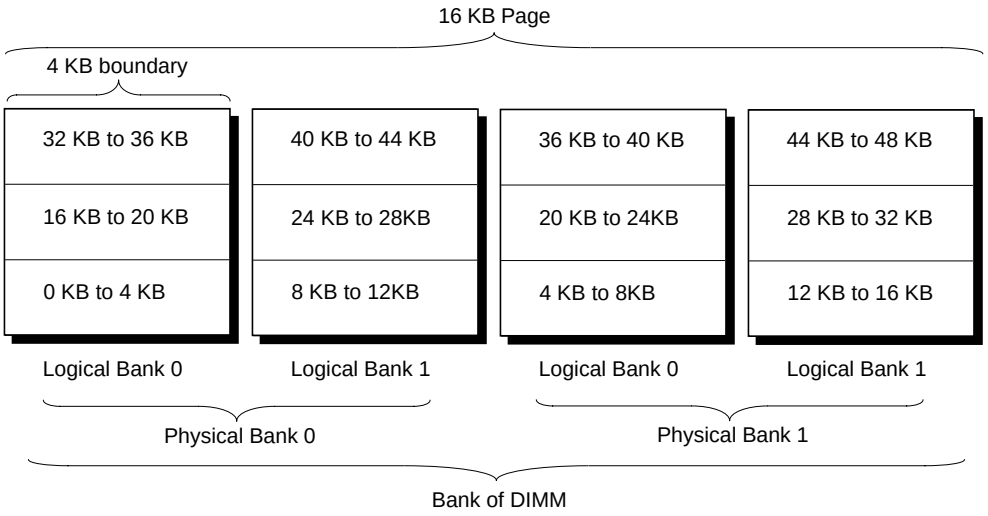

Figure 2-4 Memory Banks and Interleaving

Memory Blocks, Lines, and Pages

The smallest unit of data that can either be present or not present in memory is broadly referred to as a **block;** a block is also the smallest unit of data over which coherence is maintained.

More specifically, a block of data in a cache is called a **line**, as in "cache line." In Origin2000, secondary cache lines are fixed in size at 32 words, or 128 bytes. A block of data in main memory can have a different name: a **page**, as in a "page in memory." Main memory page sizes are multiples of 4 KB, usually 16 KB.

Note: In main memory, coherence is maintained over a cache line, not a page.

Depending on the cache,[1] lines are accessed, or **indexed,** using either a virtual or a physical address. Pages in main memory are indexed with a physical address. This process is shown in Figure 2-5.

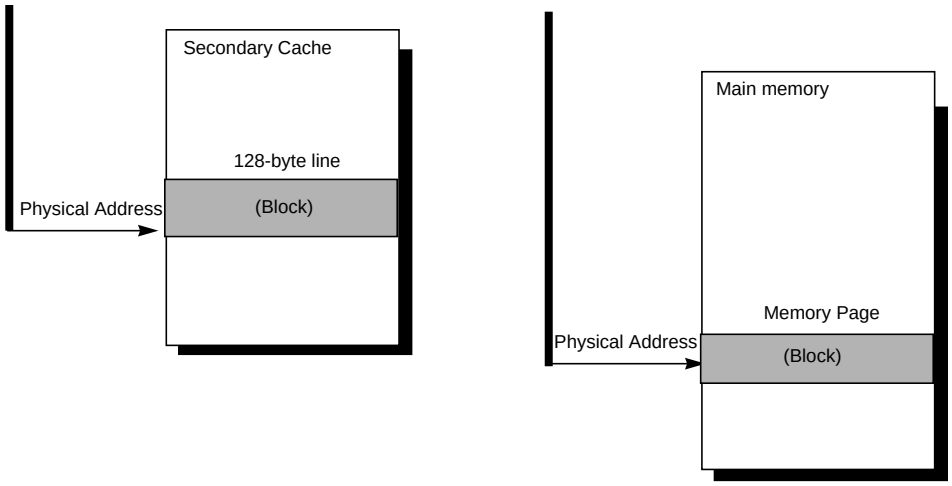

Figure 2-5 Cache Lines and Memory Pages

[1] Primary caches are indexed with a virtual address, while the secondary cache is indexed with a physical address. Physical and virtual addressing are described in the section, "Virtual Memory."

Virtual Memory

Virtual memory, or virtual addressing, is used to divide the system's relatively small amount of physical memory among the potentially larger amount of logical processes in a program. For instance, let's say a system has 32 MB of main memory, and it is being used by 10 users, each of whom have 100 processes. Dividing physical memory equally would limit each user process to a minuscule 32 KB of memory. Alternatively, it would be too expensive to dedicate a full 32-MB address space to each process: this would require 32 GB of physical memory.

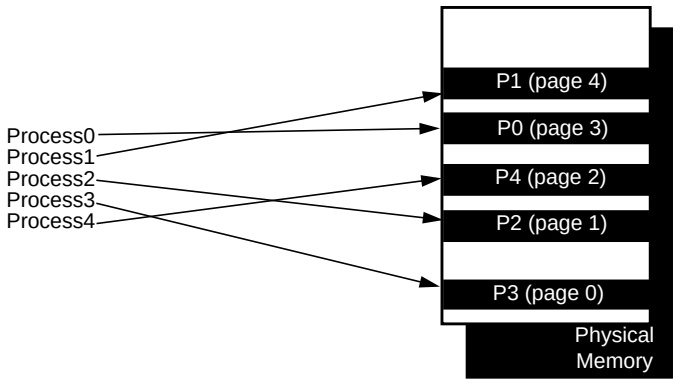

Figure 2-6 Allocating Physical Memory to Virtual Processes

Instead, virtual addressing provides each process with a "virtual" 32 MB of memory, as shown in Figure 2-6. It does this by dividing physical memory into **pages**, and then allocating pages to processes as the pages are needed. In an action called **mapping**, each physical page links or "maps" its physical address to the virtual address of the process using it.

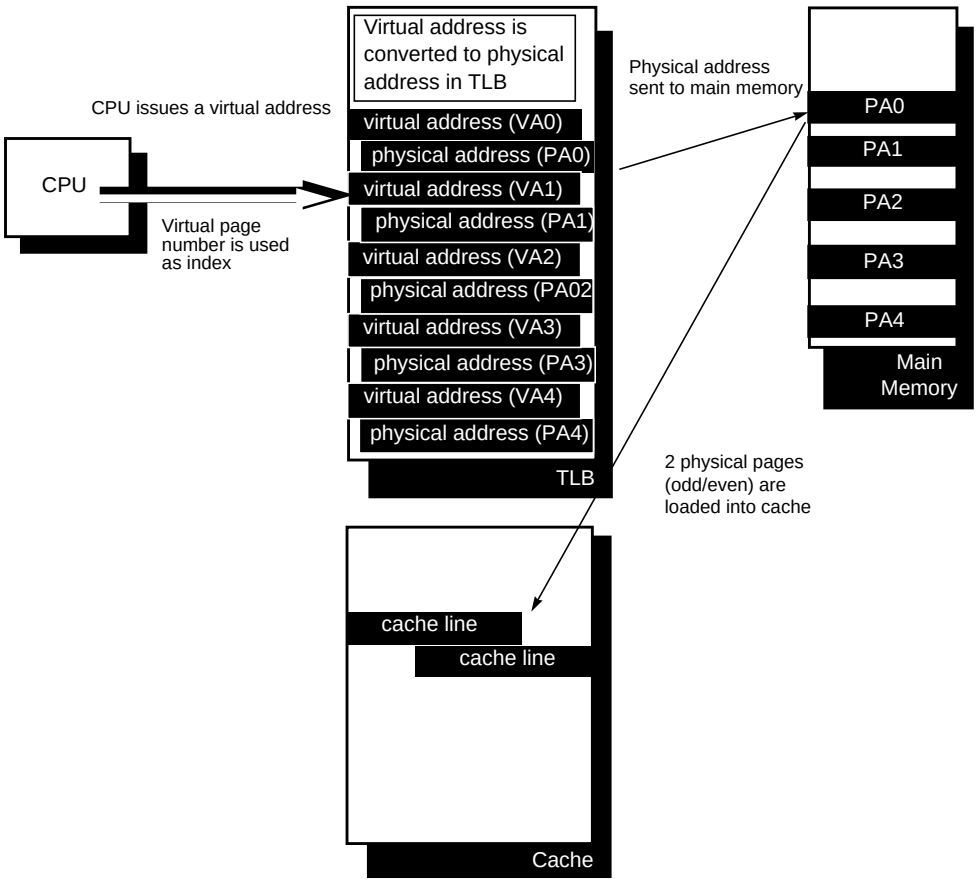

Figure 2-7 Converting Virtual to Physical Addresses

These virtual-to-physical address mappings can be found in several places:

- a software table in main memory called a **page table**; each entry in the table is called a **page table entry**, or PTE
- several R10000 processor registers used for memory management, if the page has been recently retrieved
- a hardware buffer called a **translation lookaside buffer**, which acts as a cache for quick retrieval of recently-used virtual-to-physical address mappings

Translation Lookaside Buffer

References to the page table in main memory can be time consuming. To reduce this latency, a subset of the page table is stored in a fast buffer of registers called a **translation lookaside buffer**. The TLB allows rapid access to the most-recently-used address translations.

Figure 2-8 shows a part of a virtual-to-physical address mapping, as it is contained in the R10000 TLB. Each entry is 256 bits (8 words) long. Notice that each virtual address maps to a pair of physical pages: an even-numbered page (for instance, page 0) and its corresponding odd-numbered page (page 1).

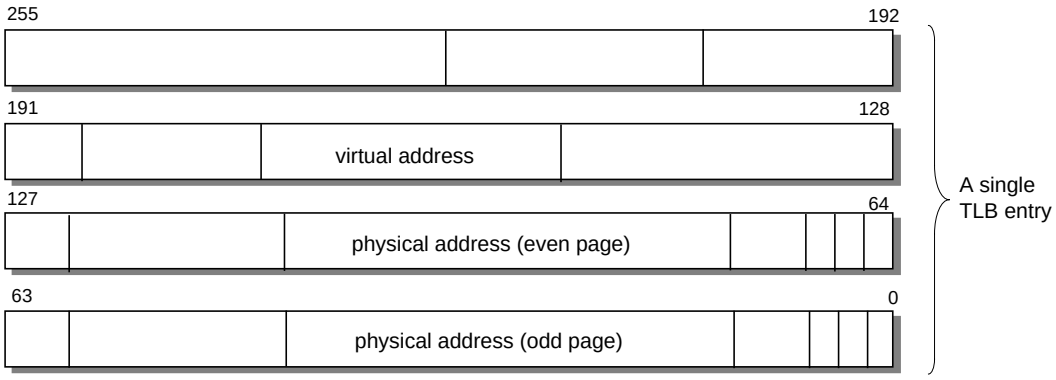

Figure 2-8 Virtual-to-Physical Address Mapping in a TLB Entry

Hits, Misses, and Page Faults

When a process finds a cache line it is looking for, it is said to make a **hit** in the cache. However, sooner or later a process fails to find the line it is looking for in the cache. This is called a cache **miss**. When the process can't find a page in main memory, this failure is called a **fault**, as in "page fault."

In a cache miss or a page fault, the next lower level(s) of memory hierarchy are searched for the missing data.

Cache-Coherence Protocol

Simply put, coherence is the ability to keep data consistent throughout a system. Data coherence in a uniprocessor system is usually managed directly by the processor, so no separate coherence protocol is needed.

A multiprocessor configuration is different. In a system like Origin2000, data can be copied and shared amongst all the processors and their caches. Moving data into a cache reduces memory latency, but it can also complicate coherence since the cached copy may become inconsistent with the same data stored elsewhere. A **cache coherence protocol** is designed to keep data consistent and to disperse the most-recent version of data to wherever it is being used.

Here's an example of cache coherence. It starts when a new block of data is loaded into a single processor's cache. Since there is only one instance of this data in the entire system, the coherence state of the data is said to be **Exclusive**, as shown in Figure 2-9. This state is indicated by setting the *Exclusive* bit in the **directory entry** that is assigned to the memory block (see "Directory-Based Coherence" for a description of directory-based coherence).

Figure 2-9 Exclusive Data

When Processor 1 needs to use this data, it makes a copy and loads the copy into its cache. This means there are now two instances of the same data in the system, which also means the data in Processor 0 is no longer exclusive.

When the copy is made, the directory sets the *Shared* state bit, indicating that this block has a copy somewhere[1] in the system. This is shown in Figure 2-10.

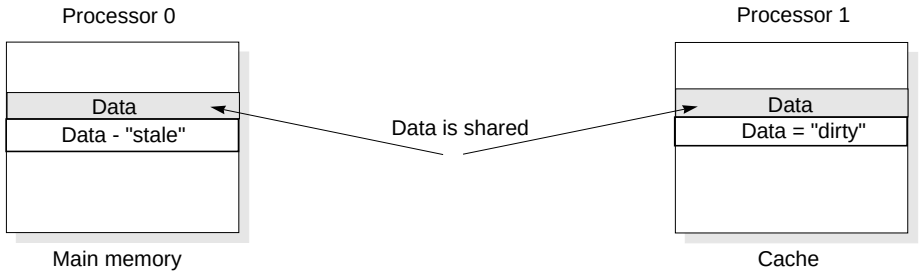

Figure 2-10 Shared Data

Processor 1 uses the data. If the data has been changed, it is referred to as **dirty** and is marked for writeback.[2]

Once Processor 1 is finished using the data, the next step is for Processor 1 to write this data into its cache. However, if this write were allowed to execute there would be two different versions of the same data in the system: the newly-modified data in Processor 1 cache, and the unmodified, or "stale" version in the main memory and cache of Processor 0. Therefore the stale data must be made consistent with the dirty version in Processor 1 cache.

There are two different methods, called **protocols**, of reconciling the inconsistency between the versions of data:

- snoopy-based protocol (see the section titled "Snoopy-Based Coherence").
- directory-based protocol (see the section titled "Directory-Based Coherence").

Origin2000 uses a directory-based coherence protocol.

---

[1] That "somewhere" is indicated by a bit vector in the same directory entry, which points to the node(s) storing the shared copy.

[2] This applies to read/write data only. Since read-only data cannot be modified, it does not need to be written back.

Snoopy-Based Coherence

Challenge is an example of a snoopy-based system. All processors are connected to a single, shared bus. Each processor is responsible for monitoring, or "snooping," this bus for memory reads that might affect the data in its cache—for instance, a write to shared data, or a read of clean data. Each processor is also responsible for broadcasting each of its cache misses to the entire system.

Snoopy protocols are generally used on small-sized systems; given the speed of today's RISC processors, requiring all cache misses to be broadcast can quickly swamp a bus. Obviously this would also limit the growth, or scalability, of the system, since adding more processors to the bus would only serve to saturate it more quickly.

Directory-Based Coherence

Origin2000 uses a directory-based coherence protocol. In a directory-based protocol, each block in memory has an accompanying directory entry. These entries are kept in a table called a **directory**. Since memory is distributed throughout the system, directories and their entries are also distributed.

As shown in Figure 2-11, each **directory entry** contains information about the memory block such as its system-wide caching state, and bit-vector pointing to caches which have copies of the memory block. By checking the state and the bit vector, memory can determine which caches need to be involved with a given memory operation in order to maintain coherence.

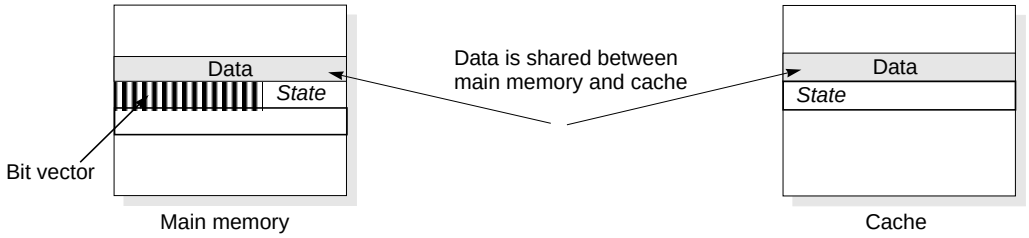

Figure 2-11 Directory-Based Coherence

Directory-based coherence avoids the bandwidth problem of a snoopy protocol by not requiring each processor to broadcast their memory accesses. Instead, only those caches which contain copies of the memory block need to be notified of a memory access, and then only if they are affected. This feature of the directory protocol assures that the scalable bandwidth of the interconnection fabric is not compromised by the support of cache coherence.

Two methods may be used to maintain coherence in a directory-based system: update, and invalidate. Origin2000 uses the invalidate method.

Maintaining Coherence Through Invalidation

An **invalidate** purges all copies of the modified, or dirty, cache line from all other caches in which the line resides. Invalidation is done by setting an *Invalid* bit in the cache line's tag. If an invalidate is executed, the single remaining line becomes exclusively owned by the writing processor. Invalidation allows the processor having exclusive ownership to make further writes to the line without having to notify other caches as each write occurs.

Why Coherence is Implemented in Hardware

Were cache coherence not implemented in hardware, the responsibility for coherence would fall either to the user or to the compiler. Neither of these alternatives are preferable, since:

- leaving its responsibility to the user could greatly complicate the programming effort, and
- leaving its responsibility to the compiler greatly complicates the compiling effort and forces the compiler to conservatively limit the amount of caching.

Designing coherence into the hardware allows the compiler to concentrate on optimizing for latencies.

Memory Consistency

Various models are used to maintain data coherence throughout a computer system. The most stringent model is called *sequential consistency*, which is used in Origin2000.

Using **sequential consistency**,

- all processors issue memory requests in program order
- all memory operations return values that correspond to a sequential ordering of the memory references by the processors.

Ordering is enforced between uncached writes and cached operations, but overlap is still possible due to the design of the R10000.

Directory States

Origin2000 uses four stable states and three transient states for the directory.

- *Unowned*: (uncached) the memory block is not cached anywhere in the system
- *Exclusive*: only one readable/writable copy exists in the system
- *Shared*: zero or more read-only copies of the memory block may exist in the system. Bit vector points to any cached location(s) of the memory block.
- Busy states: *Busy Shared*, *Busy Exclusive*, *Wait*. These three transient states handle situations in which multiple requests are pending for a given memory location.
- *Poisoned*: page has been migrated to another node. Any access to the directory entry causes a bus error, indicating the virtual-to-physical address translation in the TLB must be updated.

Note: Directory poisoning is an architectural feature which is being proposed for future implementations. In implementations where directory poisoning is unavailable, the operating system uses a more conventional TLB shootdown algorithm to handle the TLB updates required by page migration.

Sample Read Traversal of Memory

A read cycle traversing the memory hierarchy is described below.

1. The processor issues a read request to its cache.
   - If data is present, the processor continues execution.
   - If data is not present—a cache miss—the read request is passed on to the home node of the memory (home may be either local or remote).
2. The read request is made to home memory. If block is in home memory, fetch it. Simultaneously, check the coherence state of the block's directory entry.
   - If data is *Unowned*, return data to requestor and update directory entry to *Exclusive.*
   - If data is *Shared*, return the data to requestor and update the directory to *Shared*, and set the appropriate bit in the vector for the requesting processor.
   - If data is *Exclusive*, pass the read request from home memory to the remote memory that owns the *Exclusive* copy. Go to the next level of hierarchy.
3. Remote memory returns the *Exclusive* copy of data directly to the requesting processor. In parallel with this operation, a sharing write back is executed to home memory, updating the directory entry to indicate the fact that both the remote memory and the requesting processor are now sharing this block.

Making the two operations in Step 3 parallel reduces the latency of a remote-memory read request.

Sample Write Traversal of the Memory Hierarchy

A write cycle traversing the memory hierarchy is described below.

1. The processor issues a write request to its cache.
   - If data is present in the cache—a cache hit—and the data is in a *Clean Exclusive* or *Dirty* state, the write can complete immediately and processor execution continue.

- If data is present but not *Clean Exclusive* or *Dirty*, a processor upgrade request is issued to the home memory to obtain exclusive ownership of the data. This home memory can be either local or remote.
- If data is not present—a cache miss—a read-exclusive request is issued to home memory.

2. The read-exclusive or upgrade request is passed to home memory. Home memory can service an ownership request for a memory write of a location that is either *Unowned* or *Shared*.
   - If the write is to a home memory block that is in the *Shared* state, all other copies of the block must be invalidated. The home memory directory has a bit vector which points to any node(s) holding a copy of this block, and invalidations are sent to each of these nodes. The home replies to the write requester of Step 1 by sending either an *Exclusive* data reply in the event of a *Read-Exclusive* request,[1] or an *Upgrade* acknowledgment in response to an *Upgrade* request.
   - If the write is to a home memory block that is in the *Exclusive* state, the read-exclusive request is sent on to the node that owns the *Exclusive* data.
3. Result of a read-exclusive request ripples out to remote memory in the following cases:
   - If the write request is to a *Shared* memory block, remote nodes receive *Invalidate* requests to eliminate their *Shared* copies. As each cache line is invalidated, the node sends an *Invalidate Acknowledge* back to the write requestor.
   - If the write request is to a *Exclusive* memory block, the remote owner of the *Exclusive* data receives a read-exclusive request. The remote node then takes the following two actions:

     It returns the requested data directly to the write requestor.

     It sends a transfer message to the home directory indicating that the write requestor now owns *Exclusive* data.

[1] At this time the home also changes its own state to *Exclusive*.

System Latency

Memory latency is the amount of time it takes a system to complete a read or write to memory. It is quoted using two metrics: access time and cycle time.

- **Access time** is the time between the issuance of a data request and the time when the data is returned.
- **Cycle time** is the time between repeated requests.

Memory latency is reduced by:

- the inherent spatial and temporal locality in caches
- distributing memory, and moving some of it close to each processor
- page migration, in which frequently-accessed data is moved from remote memory to local memory
- the integrated node design and CrayLink Interconnect topology, which reduces the number of chip crossbars and the contention to reach remote memory.

Locality and page migration are describing in the sections below; memory distribution is described in Chapter 1.

Locality: Spatial and Temporal

Cache memory is designed to take advantage of a fundamental property of programming: a program usually spends 90% of its execution time running 10% of its code. This is a result of what is called **locality of reference**: programs tend to reuse data and instructions they recently used.

Programs exhibit two types of locality:

- **temporal**: items that have been recently accessed are likely to be accessed again in the near future
- **spatial**: a program tends to reference items whose addresses are close to each other

Caches are designed to exploit this principle of locality by storing small subsets of main memory for rapid accessibility. Using the two principles of locality described above, a cache contains the most-frequently-used information needed by the processor.

Page Migration and Replication

To provide better performance by reducing the amount of remote memory traffic, Origin2000 uses a process called **page migration and replication**. Read-only pages are **replicated**; read/write pages are **migrated**. Page migration and replication moves data that is often used by a processor into memory close to that processor. Data is moved by a specially-designed **Block Transfer Engine**, (BTE).

Each page has an array of 64 page-reference counters—one counter for each node. These counters track the number of memory references to a line by a given node. During a memory operation, the counter for the requestor is compared against the counter for the home node—*Remote Access* and *Local Access* counters, respectively—to determine which pages should be migrated closer to the using processors. When the difference between the *Access* counters exceeds a certain software-determined threshold, the hardware generates an interrupt to the operating system (OS), indicating an excess number of accesses has been made to the remote page. The OS then to an interrupt handler.

Once in the interrupt handler, the OS uses a migration algorithm to decide whether or not to migrate/replicate the page. If the decision is made to replicate or migrate the page, the page is moved to a physical page on the node that is issuing the remote access. This action is taken while the OS is still in the interrupt handler.

Note: After the page has been moved, its TLB entry must be updated. In implementations where directory poisoning is unavailable, the operating system uses a more conventional TLB shootdown algorithm to handle the TLB updates required by page migration.

As shown in Figure 2-12, the per-page *Local Access* and *Remote Access* counters keep their counts on a *regional* basis. When there are less than 128 CPUs in a system, a region is defined as two processors sharing a hub. When a system is larger than this, a region is defined as eight hubs in the same node. The present Origin2000 system has 64 regions, one for each two-processor node.

The *Access* counters can have two widths, depending upon whether regular or extended directory DIMMs are used. The counter width is 12 bits with regular DIMMs and 20 bits with extended DIMMs.

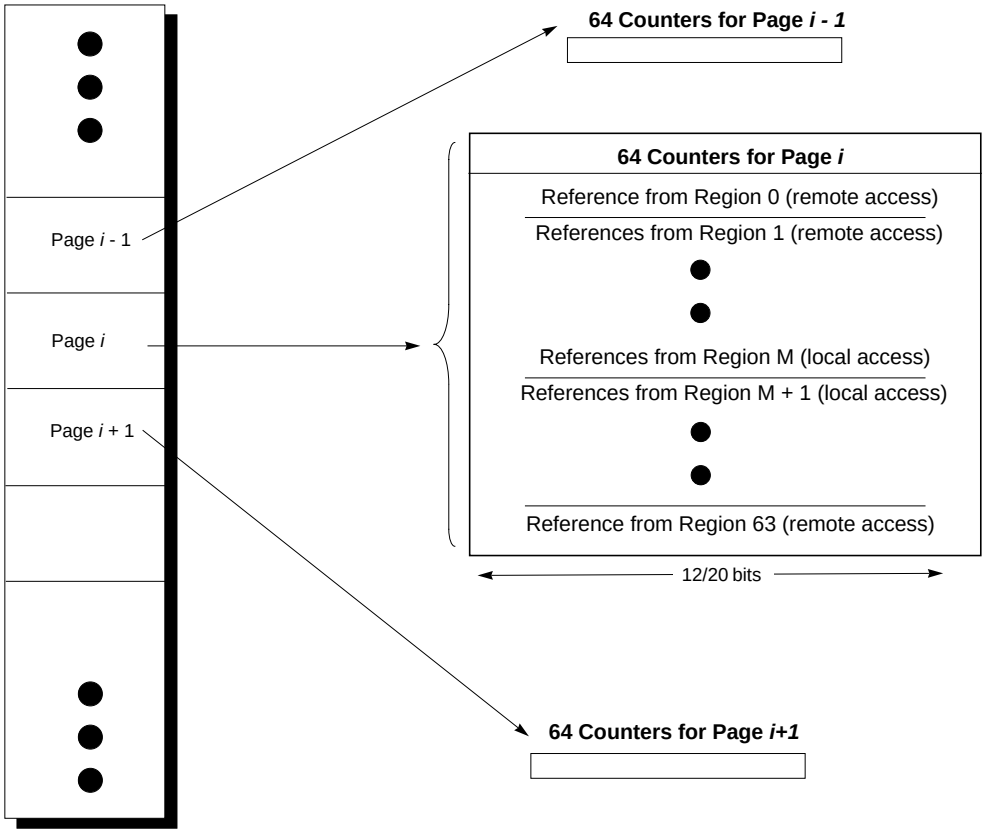

Figure 2-12 Origin2000 Local and Remote Page Access Counters

Directory Poisoning

Note: Directory poisoning is an architectural feature which is being proposed for future implementations. In implementations where directory poisoning is unavailable, the operating system uses a more conventional TLB shootdown algorithm to handle the TLB updates required by page migration.

After the page has been moved, TLBs must be updated to point to the new location of the page. Since transmitting updated address mappings to all TLBs could be quite slow, the source page (the page being copied) is instead "poisoned" during the copy operation, by setting the *Poisoned* state bit in the page's directory entry.

Once the *Poisoned* bit is set, any access to the old TLB entry returns a special bus error to update the TLB. The interrupt handler for this bus error still has the virtual address the CPU was attempting to access, and now uses this virtual address to invalidate, or "shoot down" the TLB entry.[1] The next time the CPU accesses this TLB entry, either an updated or an invalid translation is returned. If an invalid translation is returned, this page is migrated and then the newly-updated translation is loaded into the TLB.

Poisoning the block's directory entry allows global migration of pages without the overhead of having to do a global TLB invalidation of all the relevant TLB entries.

I/O

All processors can also access any I/O port as a location in uncached memory address space. Likewise, DMA access is provided for all DMA devices to all memory in the system.

Global Real-time Clock

One Hub can be designated a clock "master," sending a global clock out through the routers to the rest of the Hubs, which then slave off this clock. An optional BNC connector on the edge of the Node board can feed an external 1 MHz TTL real-time clock to the system, if greater accuracy is needed.

Node Board Physical Connections

Nodes are linked by the interconnection fabric, through a Router card. Node boards can be installed in either graphics or server modules. Each Node board connects to the module midplane through a 300-pin connector.

---

[1] This is also known as a "lazy" TLB shootdown.

A physical view of the Node board is given in Figure 2-13.

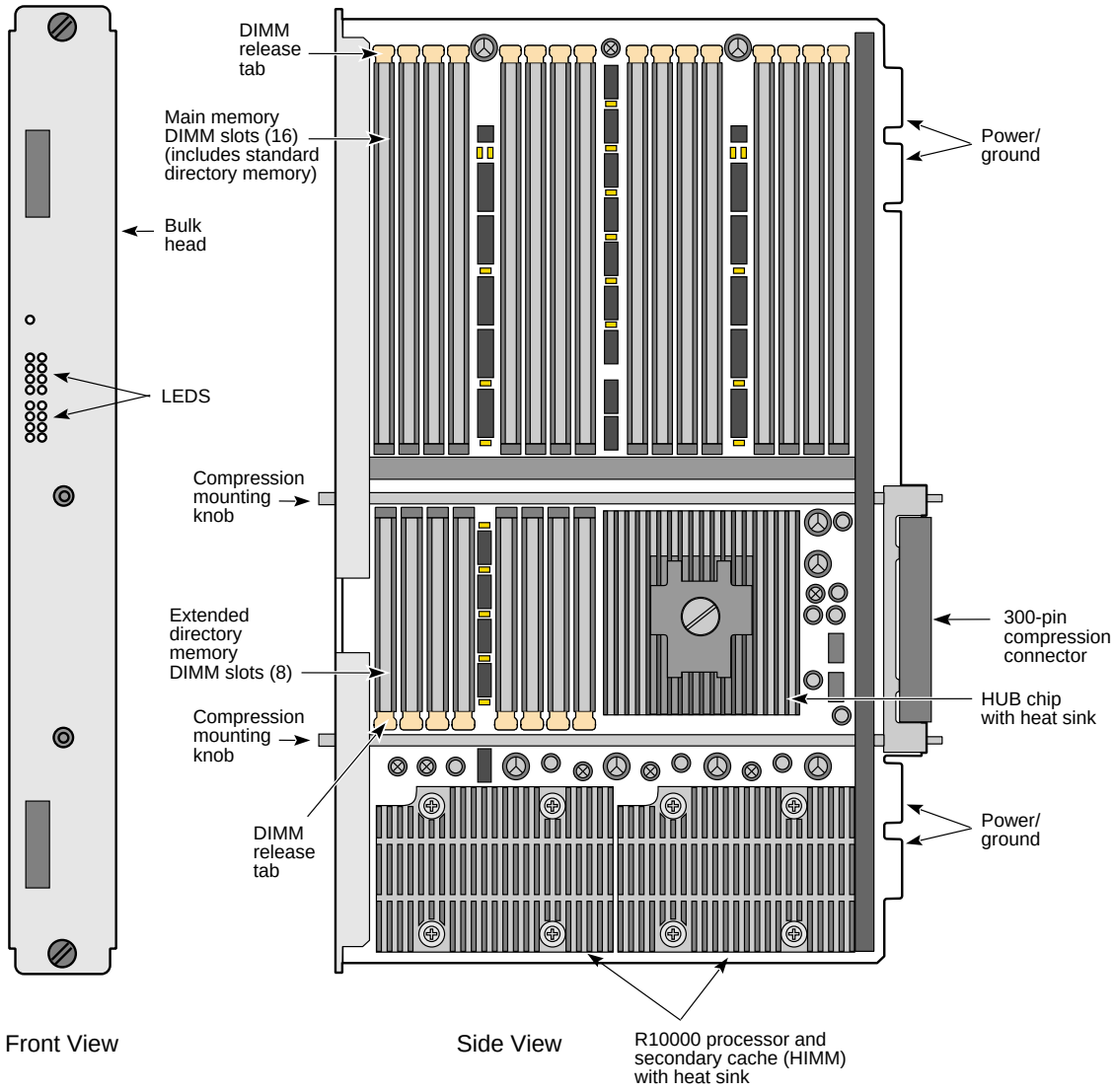

Figure 2-13 Physical View of the Node Board

XIO Protocol and Devices

Origin2000 systems use an advanced input-output (I/O) subsystem, consisting of a number of high-speed XIO links. XIO supports a wide range of Silicon Graphics and third-party I/O devices.

XIO bandwidth is given in Chapter 1 of this document.

Distributed I/O

XIO is distributed, with an I/O port on each Node board. As with Origin2000 distributed memory, each I/O port is accessible by every CPU. I/O is controlled either through the single-port XIO-protocol link on the Node board, or through an intelligent crossbar interconnect on the Crossbow ASIC.

Crossbow Expansion

A Crossbow ASIC expands the single XIO port to a total of eight ports: six are used for I/O and two connected to Node boards. Ports using the XIO protocol can be programmed for either 8 or 16 bits communications. The electrical interface for XIO is the same as that used by CrayLink Interconnect.

XIO Devices — Widgets

The form-factor for XIO widgets[1] may vary. Typically a widget is a single board, either half-size (10-inch x 6.5-inch x 1-inch) or full-size (10-inch x 13-inch x 1-inch); however it is possible for a widget to include a daughter board.

XIO can also run outside an enclosure, using the Crosstown protocol, which is described in the section titled “Crosstown Board.”

[1] I/O devices are pseudonymously referred to as “widgets.”

Crossbow Configuration

An example of a Crossbow (XBOW) configuration, with 4 CPUs (2 Node boards) and 6 I/O ports, is given in Figure 2-14. In this example the Crossbow is presented in a graphics configuration, connected to two graphics boards as well as four XIO boards.

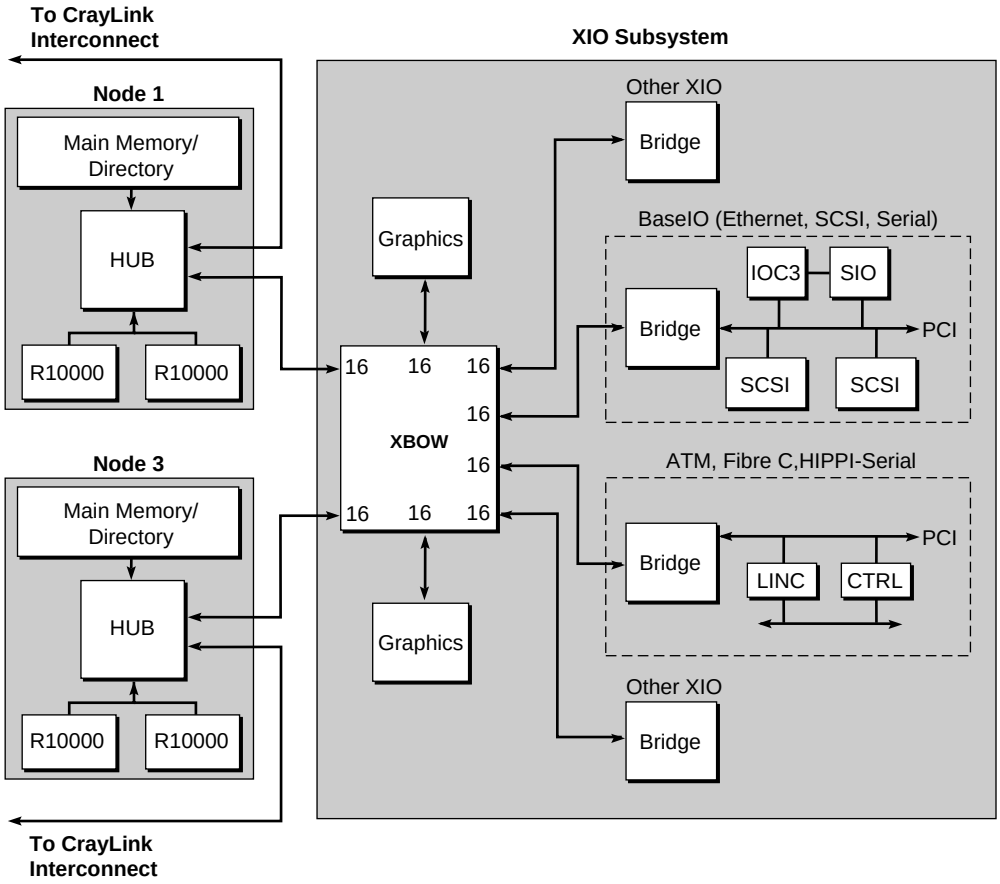

Figure 2-14 Crossbow Connections

A number of interface ASICs are available to link the Crossbow ports to PCI, VME, SCSI, Ethernet, ATM, FibreChannel, and other I/O devices. The interface ASICs include the IOC3, LINC, and Bridge ASICs, all of which are described in Chapter 3.

Router Board

Router boards physically link the Hub ASIC on the Node board to the CrayLink Interconnect. The CrayLink Interconnect provides a high bandwidth, low latency connection between all the Node boards. Central on a Router board is the Router ASIC which implements a full 6-way non-blocking crossbar switch.

Location of a Router board in a system is shown in Figure 2-15, and a physical view of the Router board is given in Figure 2-16.

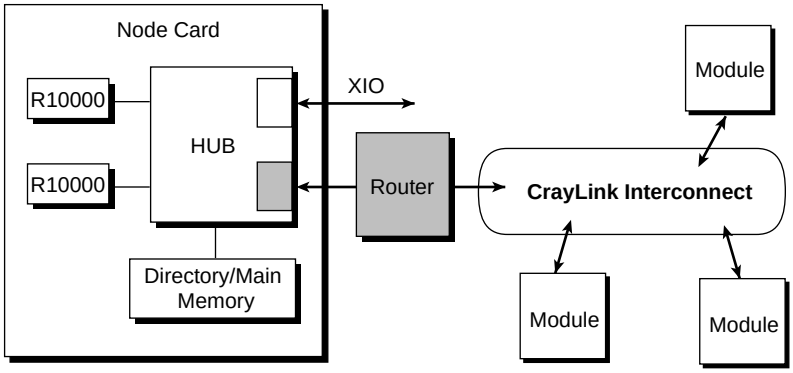

Figure 2-15 Location of a Router Board in an Origin2000 System

The Router crossbar allows all six of the Router ports to operate simultaneously at full-duplex; each port consists of two unidirectional data paths. The Router board also includes a set of protocols which provides a reliable exchange of data even in the face of transient errors on links, manages flow-control, and prioritizes data so that older data is given precedence over newer data.

Bandwidth figures are given in Chapter 1 of this document.

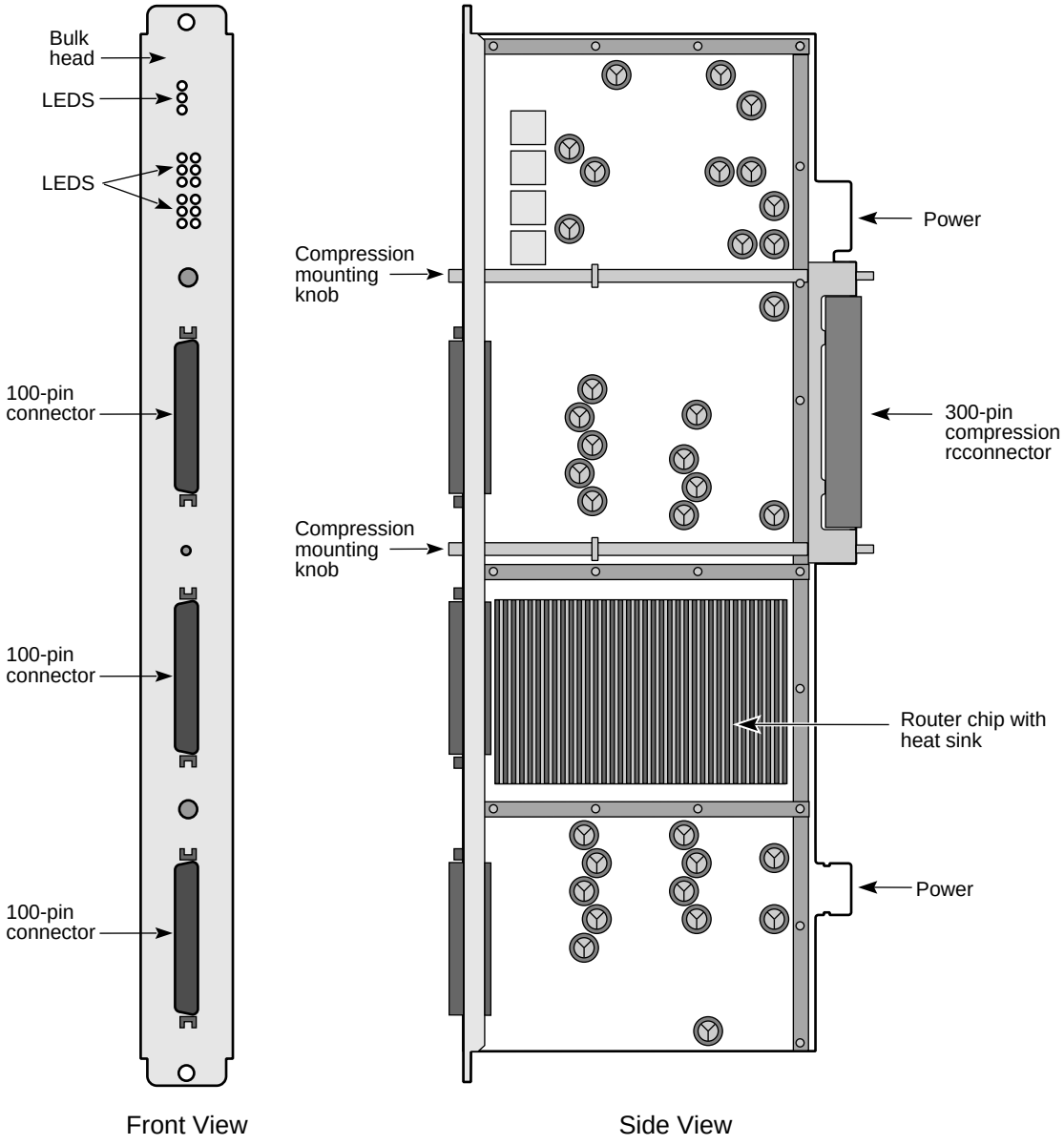

Figure 2-16 Physical View of the Router Board

Types of Router Boards

There are four types of Router boards:

- The *Null Router* connects two Node boards in the same deskside enclosure (up to four R10000 processors). The Null router configuration cannot be expanded.
- The *Star Router* connects up to four Node boards to each other. 3- and 4-Node configurations are created using a Standard router with a Star router.
- The *Standard Router* cables together from 2 to 32 Node boards (from 2 to 64 CPUs), physically located in from 1 to 8 enclosures.
- The *Meta 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 that will be available sometime in the future.

Programmable tables within the Router ASICs control packet routing through the CrayLink Interconnect. These tables allow for partial configurations (system sizes which are not 2^n) and reconfigure around broken links or inoperative modules.

SGI Transistor Logic (STL)

Each port on the Router board consists of a pair of 16-bit unidirectional links that provide 800 MB/sec of peak bandwidth each way (1.6 GB/sec bidirectional). The links use a Silicon Graphics-developed low-swing CMOS signalling technology (**STL**).

STL provides direct, very high-speed (2.5 ns cycle time) ASIC-to-ASIC communication *inside* an enclosure. These STL links can be buffered with differential ECL transmitters and receivers to drive cables to run *outside* the enclosure, connecting to other modules.

Note: Proper operation of the external modules requires correct grounding of the attached modules. Proper site preparation is essential for reliable operation.

Connectors

As shown in Figure 2-17, a Standard Router board has six CrayLink interconnections; three are used for STL midplane links, three are used for external PECL cable links.

The three midplane connections are built into 300-pin connectors. The three cable connectors are on the board bulkhead. Two of the STL connectors support links to Node boards, and the third supports a link to a second Router board in the module (if present). Each of the external connectors supports a cable connection to another Router board, either in the same module or in another module.

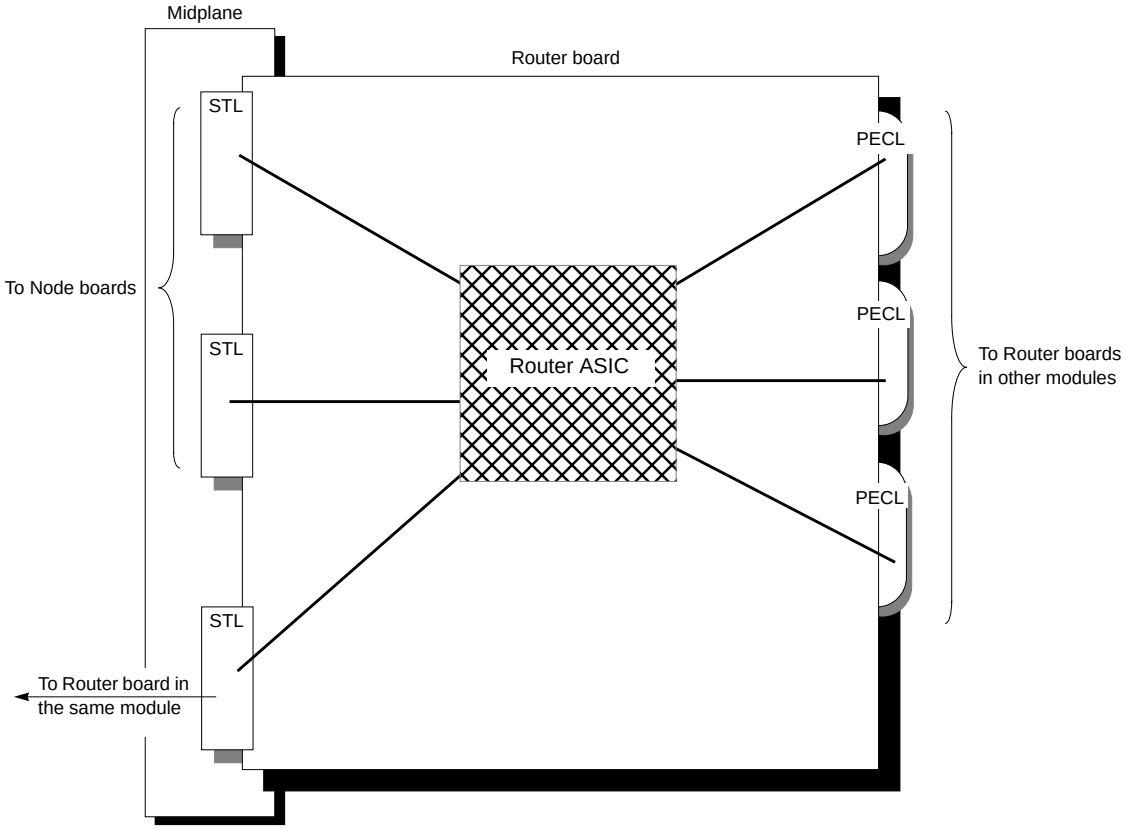

Figure 2-17 Routing Board Connectors

Xpress Links

Additional CrayLink interconnections, called **Xpress links**, can be installed between Standard Router ports for increased performance (reduced latency and increased bandwidth).

Xpress links can only be used in systems that have either 16 or 32 processors. Each Xpress link provides additional bandwidth of 800 MB/second each way (1.6 GB bidirectionally, peak).

Figure 2-18, Figure 2-19 and Figure 2-20 illustrate 16-, 24- and 32-processor systems, respectively.

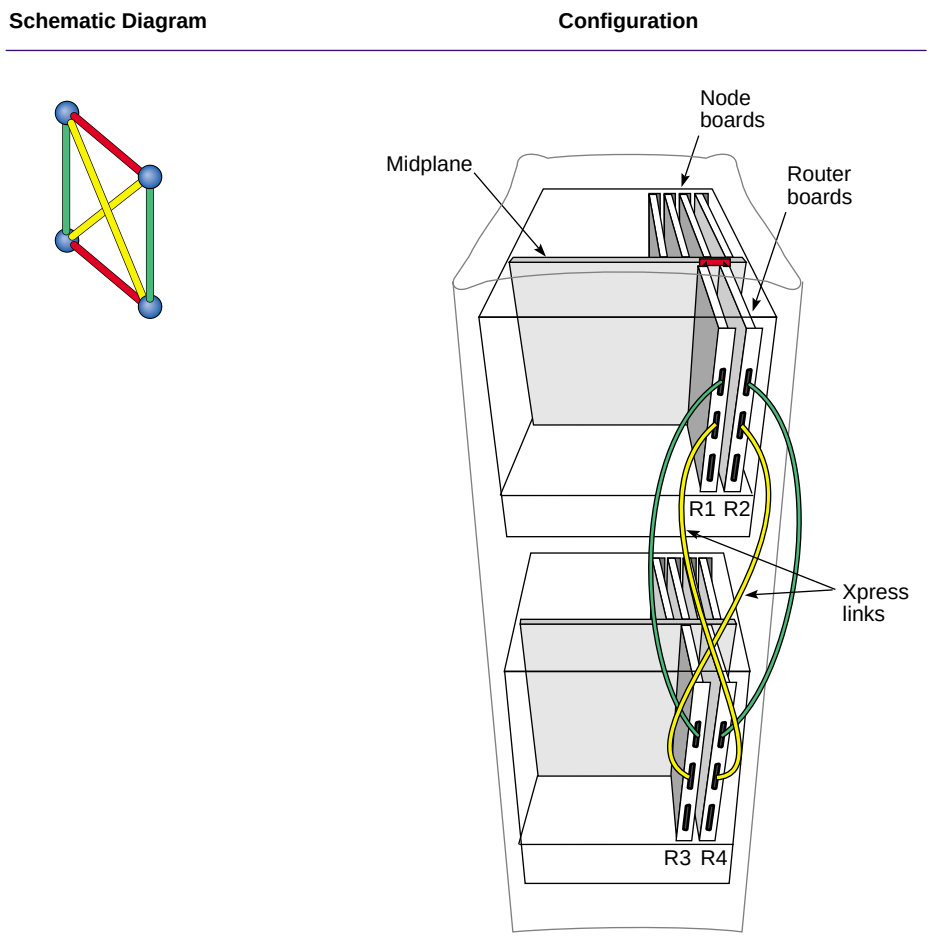

Figure 2-18 16P System Using Xpress Links

Schematic Diagram	Configuration

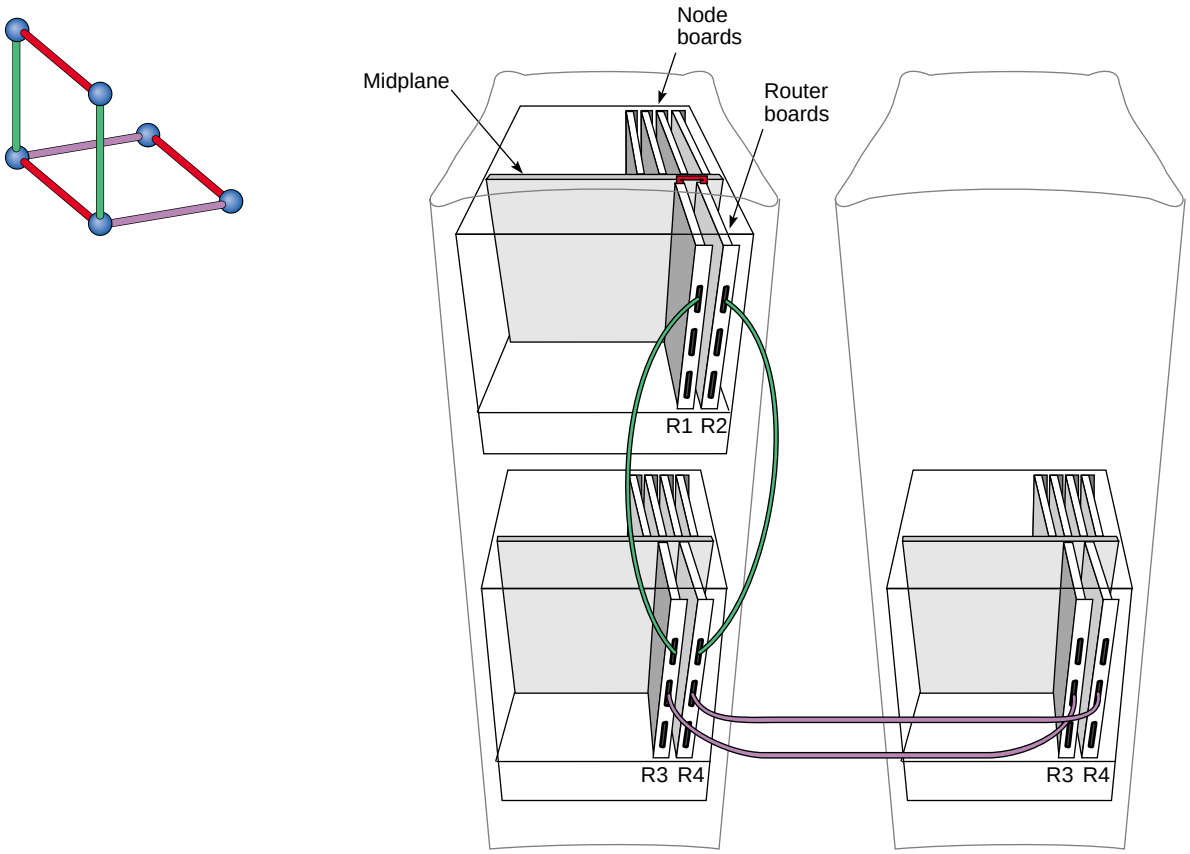

Figure 2-19 24P System

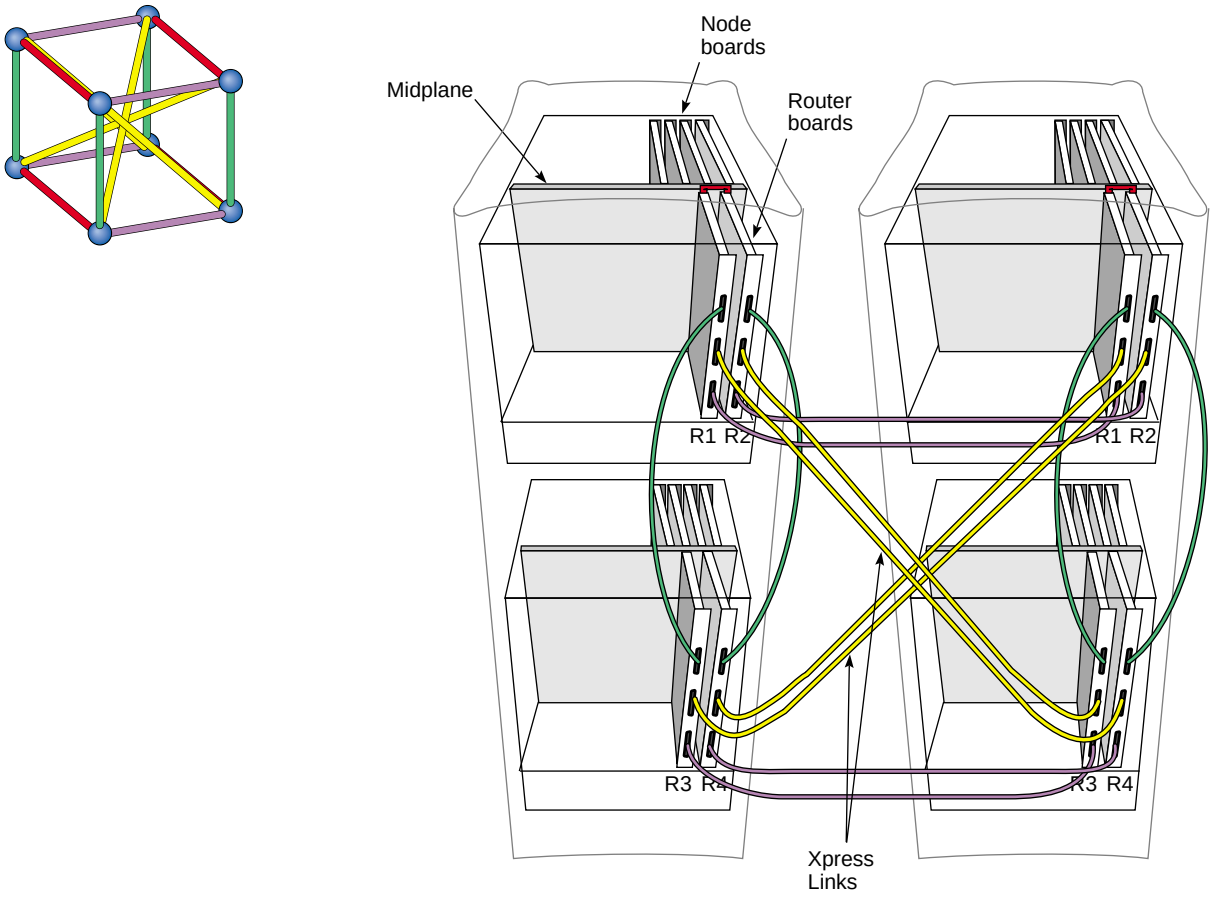

Figure 2-20 32P System Using Xpress Links

Midplane Board

The Origin2000 Midplane board is a backplane that has been moved to the center of the deskside enclosure. Figure 2-21 shows the physical location of the midplane in relation to the deskside chassis. Also shown is the Node board placement in the midplane.

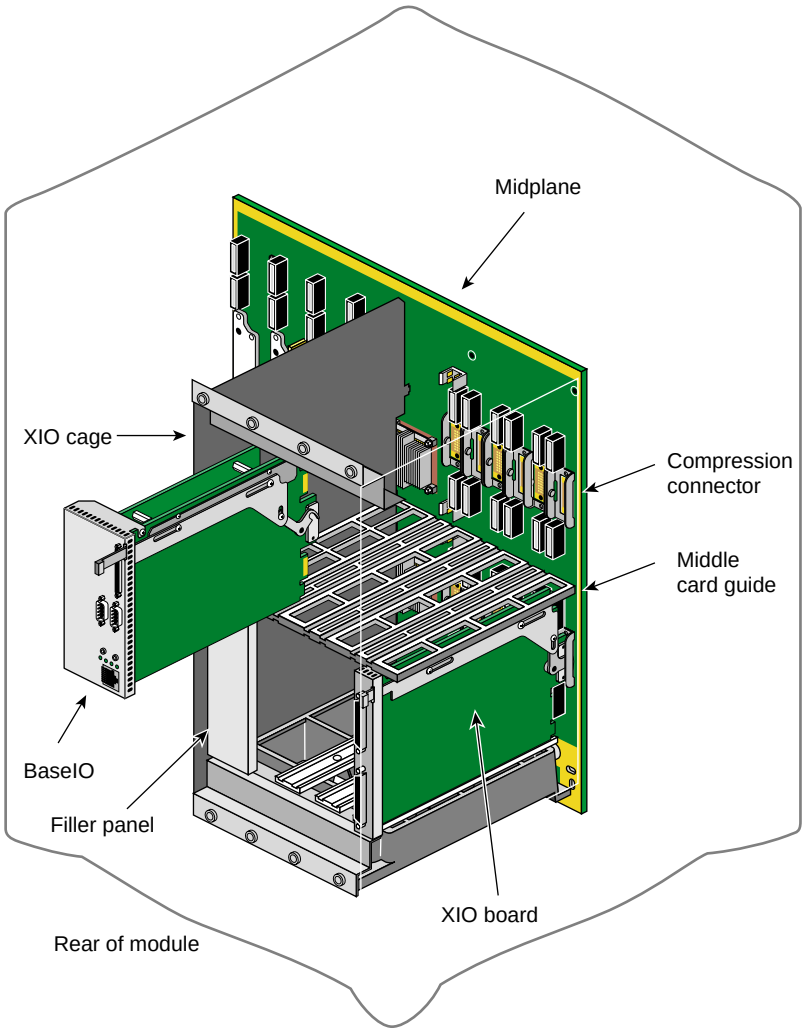

Figure 2-21 Physical Location of the Midplane in Deskside Enclosure

Functionally, the midplane provides the following:

- a standard system clock for both the XIO and CrayLink interconnection
- STL links for the CrayLink Interconnect and XIO links within the module
- power distribution throughout the system
- system control signals and additional real-time clock distribution
- digital media sync
- SCSI connections

Physically, the components located on the midplane are:

- four 300-pin STL connectors for four Node boards
- two 300-pin connectors for two Router boards
- twelve 96-pin connectors for twelve half-size XIO boards
- five connectors for five wide, single-ended Ultra-SCSI disk drives
- one connection for a 680 MB quad-speed CDROM
- one connection for a System Controller
- two SysClocks:
  - 400 MHz CrayLink Interconnect clock for Hub and Router ASICs
  - 400 MHz I/O clock for Hub, Crossbow, and Bridge ASICs

A a front view of the Origin2000 midplane is shown in Figure 2-22, and a rear view of the midplane board is shown in Figure 2-23.

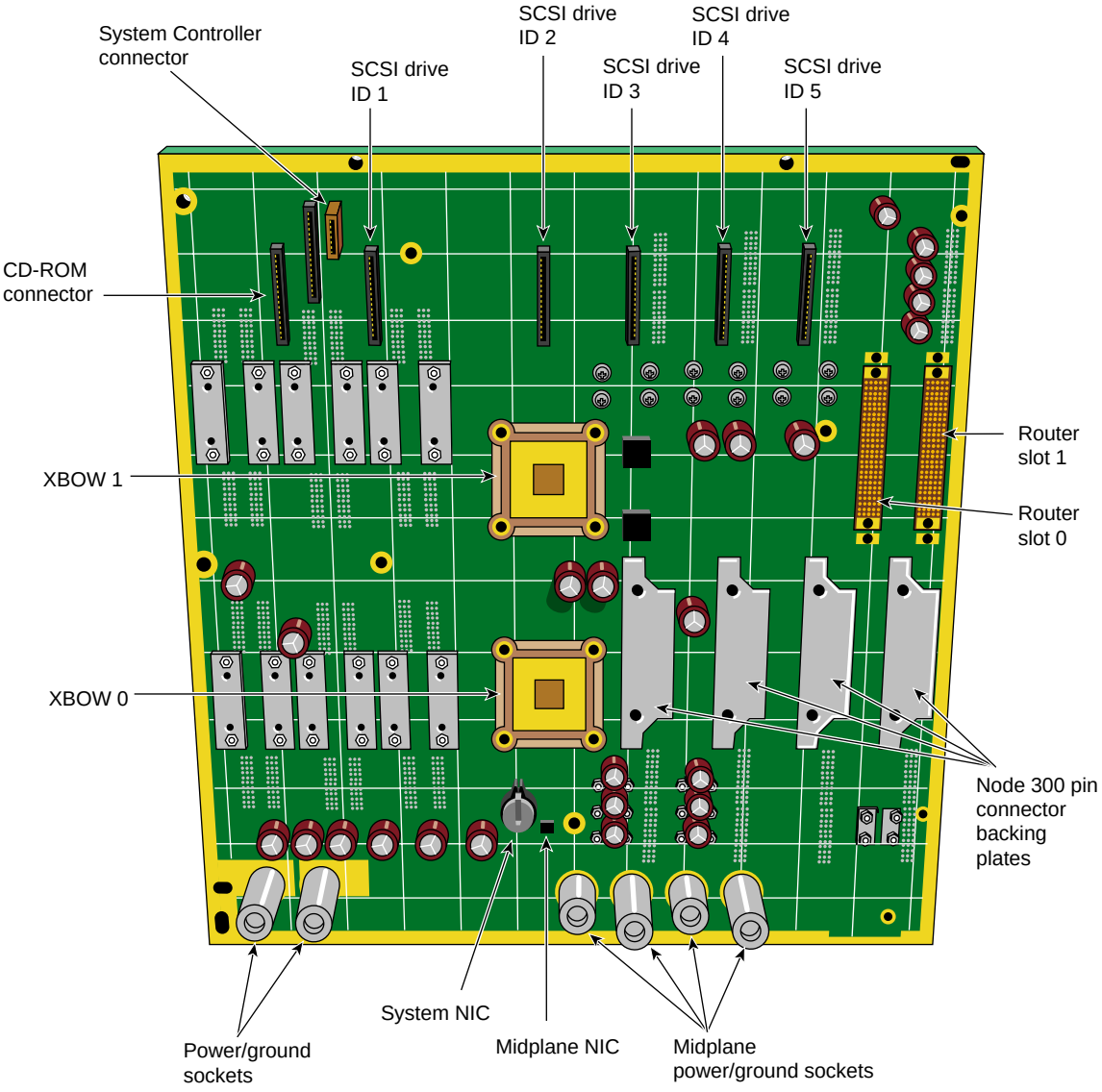

Figure 2-22 Front View of the Midplane

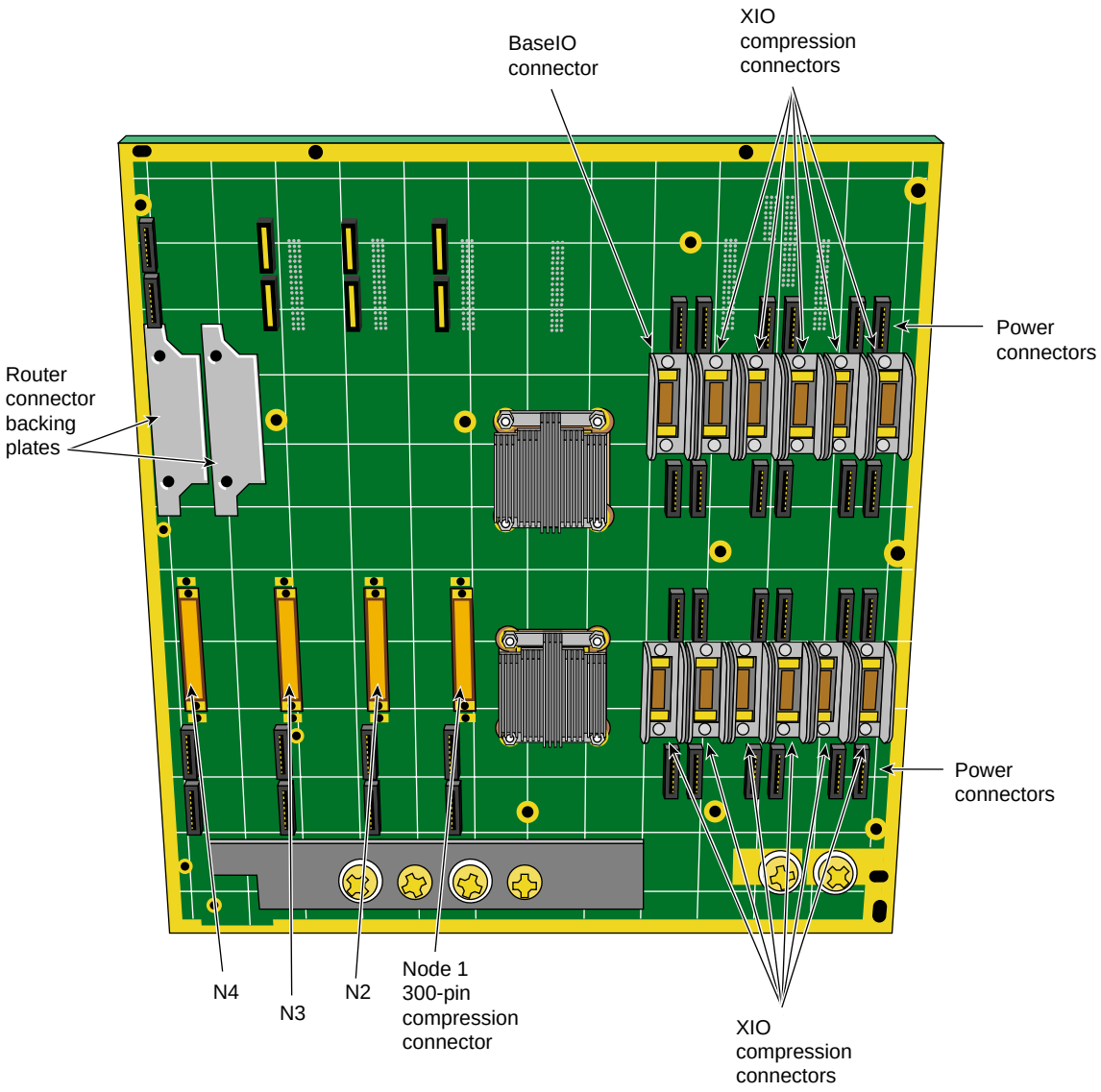

Figure 2-23 Rear View of the Origin2000 Midplane Board

BaseIO Board

The BaseIO board contains those I/O functions which come standard in each base system enclosure:

- one 10/100 Base TX Fast Ethernet link, with auto-negotiation (compliant with 802.3u)
- two 460-Kbaud serial ports, composed of dual, independent UARTS
- one external, single-ended, wide SCSI (compliant with X3.131-1994) port
- one internal Fast 20 SCSI (compliant with X3.131-1994) port
- one real-time interrupt output, for frame sync
- one interrupt input
- Flash PROM
- NVRAM
- Time-of-Day clock

The logical location of a BaseIO card, connected to a Crossbow ASIC, is illustrated in Figure 2-24. The block diagram of the BaseIO board is shown in Figure 2-25.

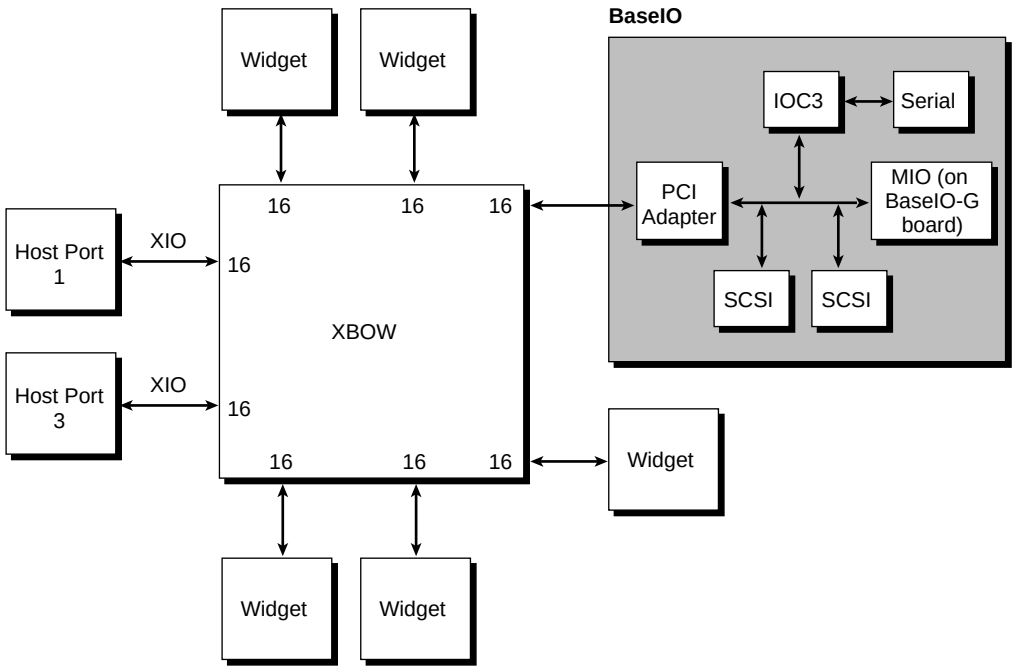

Figure 2-24 Logical Location of an BaseIO Board in an Origin2000 System

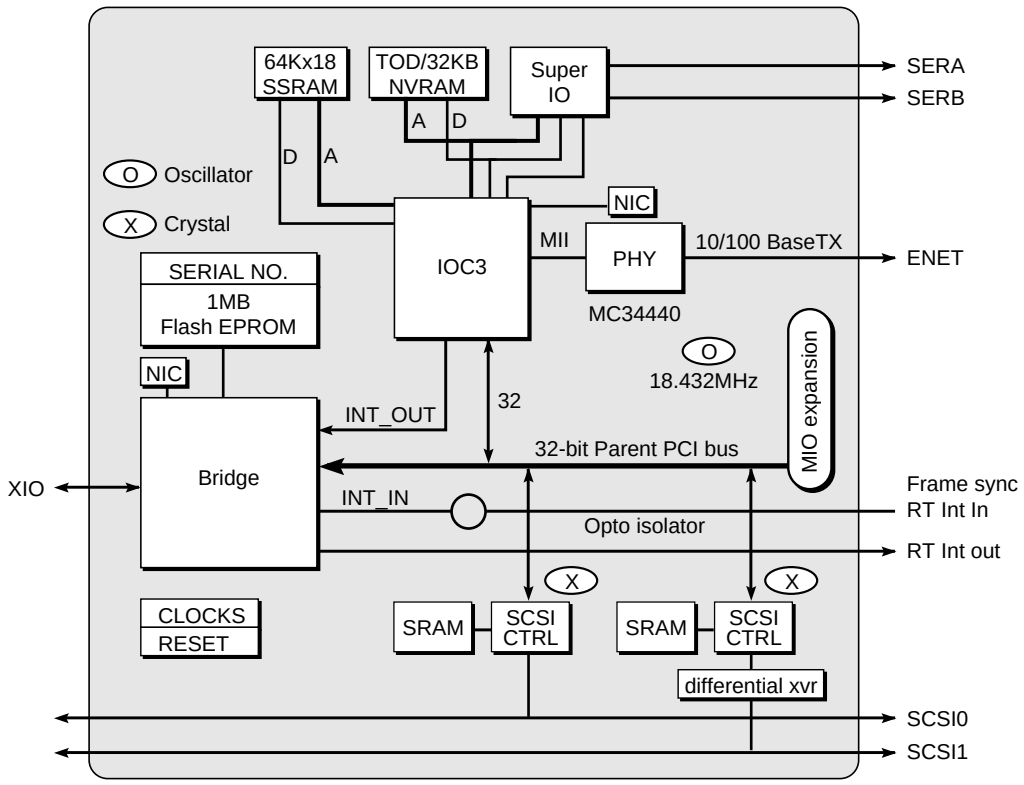

Figure 2-25 BaseIO Board Block Diagram

Due to the size of the BaseIO board, only one can be installed in each deskside enclosure. In a multi-enclosure system, there can be as few as one BaseIO board for the entire system, or there may be a BaseIO board installed in each deskside enclosure. Enclosures which do not have a BaseIO board installed can install other XIO cards in midplane BaseIO slot.

The physical layout of the BaseIO board is shown in Figure 2-26.

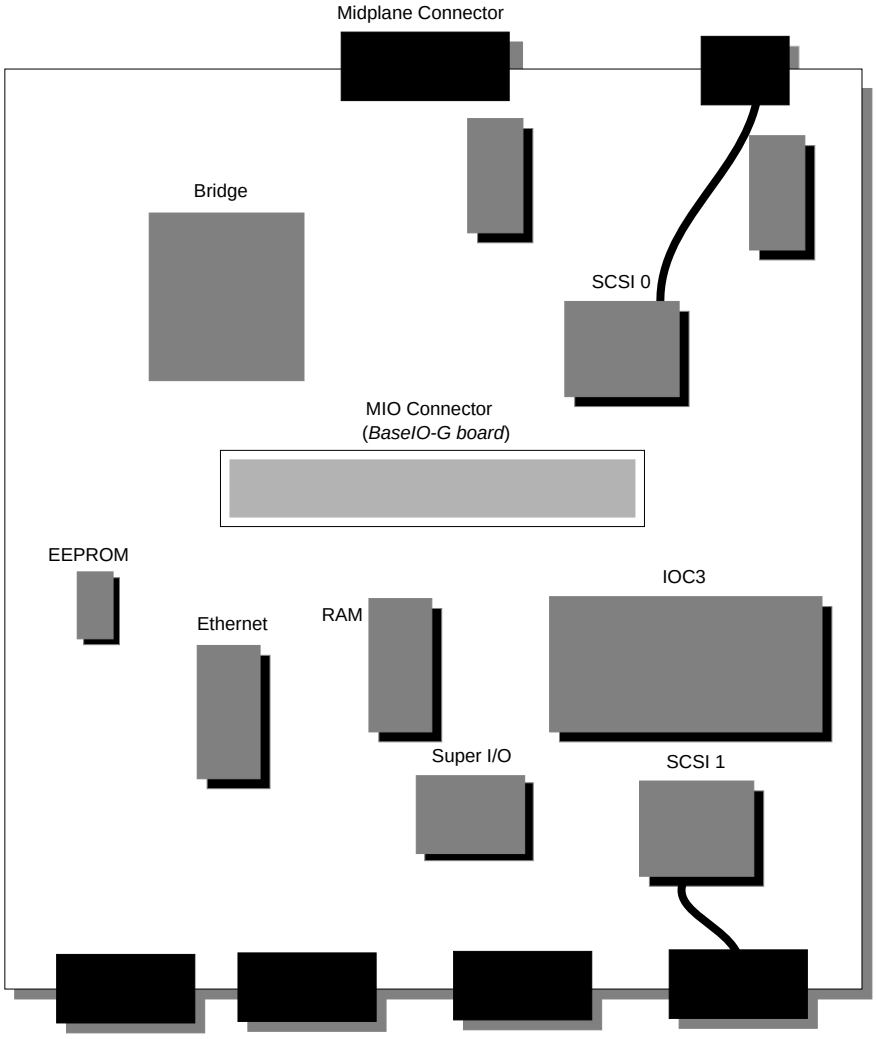

Figure 2-26 Physical Layout of the BaseIO Board

A Media IO (MIO) daughter card can be plugged into a modified BaseIO board, called the **BaseIO-G board**. The MIO card is plugged into the BaseIO-G board's PCI bus to provide additional audio and serial ports.

MediaIO (MIO) Board

The Media IO board (MIO) is mounted as a daughterboard on a specially-configured BaseIO board, called the **BaseIO-G board**. The MIO is primarily used in graphics systems to provide additional audio, serial, keyboard/mouse, and parallel ports. Specifically, the MIO board adds the following to an BaseIO-G board:

- one IEEE 1284 parallel port
- two 460-Kbaud serial ports, dual independent UARTS
- one audio analog stereo input port
- one audio analog stereo output port
- one audio AES3/AES11/SPDIF digital input port
- one audio AES3/AES11/SPDIF digital output port
- one audio Alesis ADAT/Optical SPDIF digital fiber input port
- one audio Alesis ADAT/Optical SPDIF digital fiber output port
- one keyboard port
- one mouse port

The majority of audio functions are provided by the RAD ASIC.

Dues to its size, there can only be one BaseIO/BaseIO-G board per deskside enclosure. Since an MIO board is mounted as a daughterboard on the BaseIO-G board, there can also only be one MIO board in a deskside enclosure. If more than one MIO is desired, it must be added to a multi-enclosure system in which additional BaseIO-G boards can been installed as well.

The BaseIO-G and MIO are treated as an atomic module when MIO functions are used.

The logical location of an MIO card, connected to an BaseIO-G board, is shown in Figure 2-24. A block diagram of the MIO board is shown in Figure 2-27.

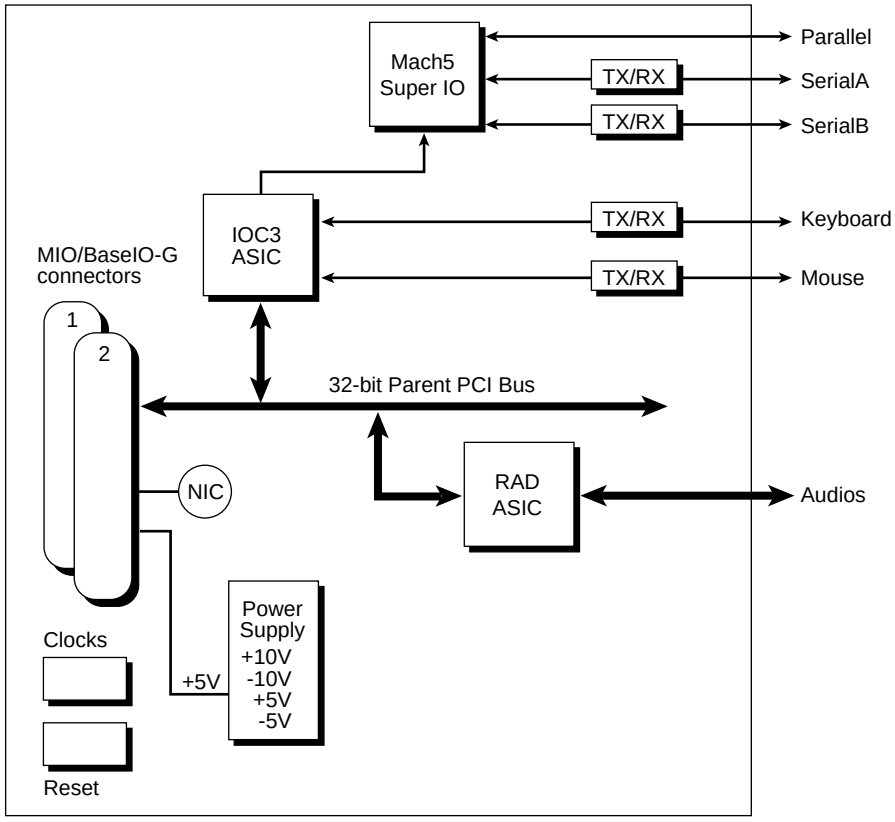

Figure 2-27 MIO Board Block Diagram

Crosstown Board

XIO can also run outside an enclosure, using the Crosstown (KTOWN) conversion board. For example, an XIO slot may be occupied by a Crosstown board, which contains an STL-to-3.45V PECL converter and a Crosstown cable attachment; the result is to convert the XIO STL link to differential signal levels. This connection can support XIO devices up to 3 meters away. Crosstown is primarily used to support graphics configurations.

A block diagram of an Origin2000 system using a Crosstown link is shown in Figure 2-28.

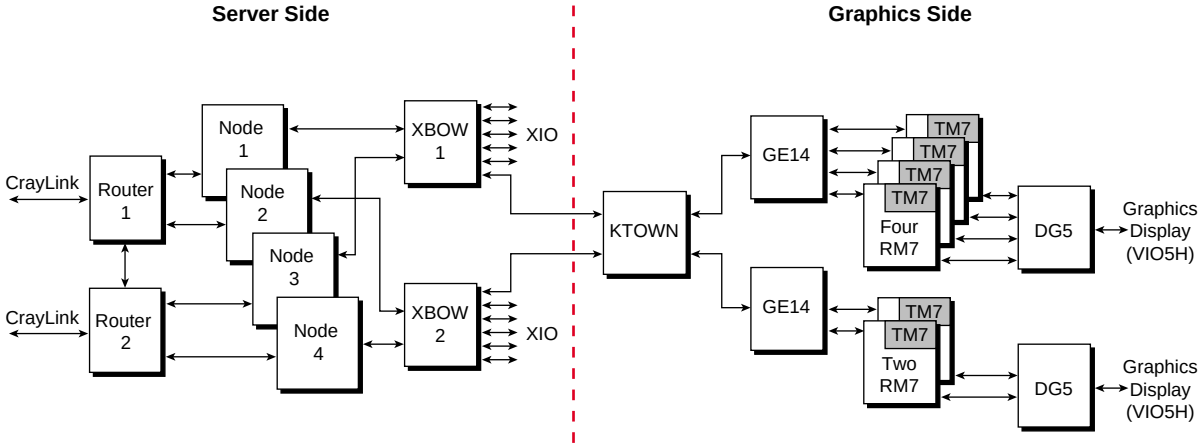

Figure 2-28 Crosstown Link

Origin200 Mother and Daughter Boards

An illustration of the Origin200 mother and daughter boards is given in Chapter 1. Each mother board can hold 1 to 2 R10000 processors each with 1 or 4 MB of secondary cache. The processors and cache memory are located on a separate daughterboard which mounts to the motherboard.

The motherboard can hold from 32 MB to 2 GB of main memory.

Chapter 3

Origin Family ASICs

Each Origin2000 module contains three key types of ASIC: Hub, Router, and Crossbow/XIO. Within an Origin2000 system, each of these ASICs is responsible for a critical element of information transfer. The ASICs and their locations in a system are:

- Hub chip: located on each Node board.
- Router chip: located on each Router board in CrayLink Interconnect.
- Crossbow chip: located on the midplane as part of the XIO interconnect
- Bridge, LINC, and IOC3 chips: located on XIO boards linked to the I/O interface

Figure 3-1 shows interconnections between these ASICs and the communication protocols that run on the interconnections from the Hub ASIC.

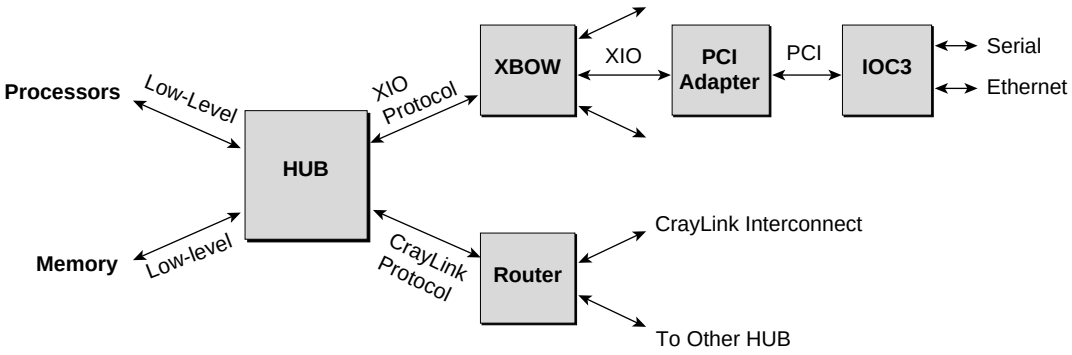

Figure 3-1 ASIC Protocols

The Hub ASIC is the core of the node. It interconnects to processors, memory, CrayLink Interconnect, and the XIO interconnect.

- From its processor interface, the Hub chip supports a low-level bus protocol similar to the shared-bus protocol on the CHALLENGE E-bus.
- To the interconnection fabric, the Hub chip supports a high-level messaging protocol called CrayLink Interconnect, which implements the cache-coherent distributed shared-memory.
- To the I/O interconnect, the Hub chip ports a separate high-level protocol called XIO, optimized for I/O devices which are not using the cache-coherence protocol.
- A memory/directory protocol interfaces with the high-speed, interleaved memory system. It supports directory memory and page migration counters.

Figure 3-1 also shows Router chip and Crossbow chip interconnections. Note that the Crossbow chip uses the XIO protocol at both its input ports and output ports. Similarly, the Router chip uses the CrayLink Interconnect protocol at both its input and output ports.

Finally, the figure shows how a Bridge chip interconnects with the Crossbow and a communications controller on an XIO board; in this case, the controller is the IOC3 chip. The Bridge takes Crossbow as input, and interfaces it with the PCI protocol. The controller supported by the Bridge (the IOC3) takes PCI as input and interfaces it to Ethernet and serial.

Hub ASIC

The Origin2000 Architecture defines the theoretical model of a distributed, shared-memory multiprocessor system employing from 1 to 1024 processors.[1] Origin2000 has a single address space with cache coherence applied across the entire system.

Origin2000 is organized into a number of nodes; each node contains up to two processors, a portion of the global memory, a directory to maintain cache coherence, an interface to the I/O subsystem, an interface to the other nodes on the system, and a Hub ASIC which links all of these subsystems through a crossbar.

Hub Interfaces

The **Hub ASIC** can be viewed as the core of the node. Physically, it is located on the Node board, and the Hub is responsible for connecting all four interfaces of the node together:

- processor
- memory
- XIO
- CrayLink Interconnect through a Router board

These are described briefly below.

Processor
- Each node can have up to two R10000 processors linked to the Hub.

Memory
- A portion of the distributed, shared main memory is connected to the Hub, together with directory memory used for cache coherence and page migration counts. This memory can range in size up to 4 GB.

I/O
- Either one or two Hubs can be connected to each the Crossbow ASIC.

CrayLink Interconnect
- Either one or two hubs can be connected to a Router board, which in turn links the Hub to other nodes connected to the system-wide interconnection fabric.

These four interfaces are interconnected by an internal crossbar, as shown in Figure 3-2. The interfaces on the Hub communicate by sending messages through the crossbar.

[1] The initial release of the product supports 128 CPUs.

The Hub controls *intranode* communications between the node's subsystems, and also controls *internode* communications with other Hub ASICs in other nodes. The Hub converts internal messages, using a request/reply format, to and from the external message format used by the XIO or CrayLink Interconnect port. All internal messages are initiated by processors and I/O devices.

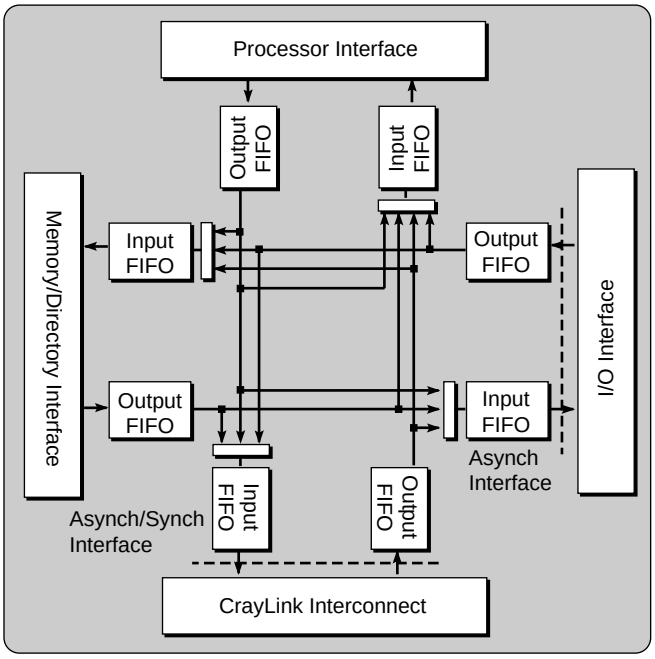

Figure 3-2 Block Diagram of a Hub ASIC

The four node interfaces act as individual controllers for their subsystems. Each interface takes inputs from an external source and converts the inputs to an internal (intra-Hub) message.

Messages can be classified as requests or replies. Each input and output FIFO associated with a Hub chip interface is logically divided into two queues; one to process requests, the other for replies. The cache coherence protocol together with the separate logical request and reply paths guarantee that deadlock is avoided.

These messages (read, write, etc.) are converted by the respective interfaces into CrayLink Interconnect requests to the appropriate target — memory or I/O interface.

Depending upon the directory state, the target either replies or sends additional requests, such as an intervention or an invalidate, to other interfaces within the same or different hubs.

For example, suppose a processor sends a programmed I/O (**PIO**) message to a local I/O device. The message is received at the Hub chip processor interface, converted to intra-Hub format, and passed through the Hub to the I/O interface. At the I/O interface, the message is converted to XIO format and placed on the local XIO interconnect.

As shown in Figure 3-2, each Hub interface has two first-in/first-out (**FIFO**) buffers: one for incoming messages and one for outgoing messages. The FIFOs provide buffering between the Hub chip and the devices to which it connects. When empty, the FIFOs provide bypassing for lower latency.

Cache Coherence

The Hub ASIC provides cache coherence by:

- implementing the processor portion of the Origin2000 distributed-coherence protocol
- converting XIO memory requests to the Origin2000 distributed-coherence protocol
- maintaining the directory caching information and migration counts for all sections of main memory

Static Partitioning of I/O

Either one or two Hubs can be connected to a Crossbow ASIC. In a dual-Hub (or **dual-host**) configuration, the six remaining XIO ports are **statically partitioned** between the two Hubs; that is, the I/O devices are assigned to one or the other of the Hubs/Node boards.

Access to an XIO device allocated to a particular hub is always made through that hub. If a remote processor accesses the device, it first communicates the request over the CrayLink Interconnect to the owning hub, which then relays the request over XIO. A statically-allocated device is local only to the Hub that owns it; to all other Hubs the I/O device is a remote device, even to a Hub physically sharing the same Crossbow ASIC.

Router ASIC

The **Router ASIC** is a 6-port dynamic switch which forms the interconnection fabric connecting the nodes. Physically, the Router chip is located on a Router board, which plugs into the deskside midplane opposite the XIO and Node boards. A block diagram of the Router ASIC is given in Figure 3-3.

Functionally, the Router ASIC does the following:

- Determines the most-efficient connection of receive to send ports, given the set of received messages, and dynamically switches connections between any of the six pairs of ports, through the 6-way crossbar.
- Communicates reliably by using the CrayLink Interconnect link-level protocol (**LLP**) to other routers and hubs.
- May route different messages to the same destination through different paths, for greatest speed and efficiency (adaptive routing).
- To reduce latency, routes messages without having to receive the entire message (wormhole routing).
- Buffers CrayLink Interconnect messages.

The block diagram shows the source-synchronous drivers (SSD) and receivers (SSR) that multiplex and demultiplex the high-speed external connection (400 MHz) to the router internal frequency (100 MHz).

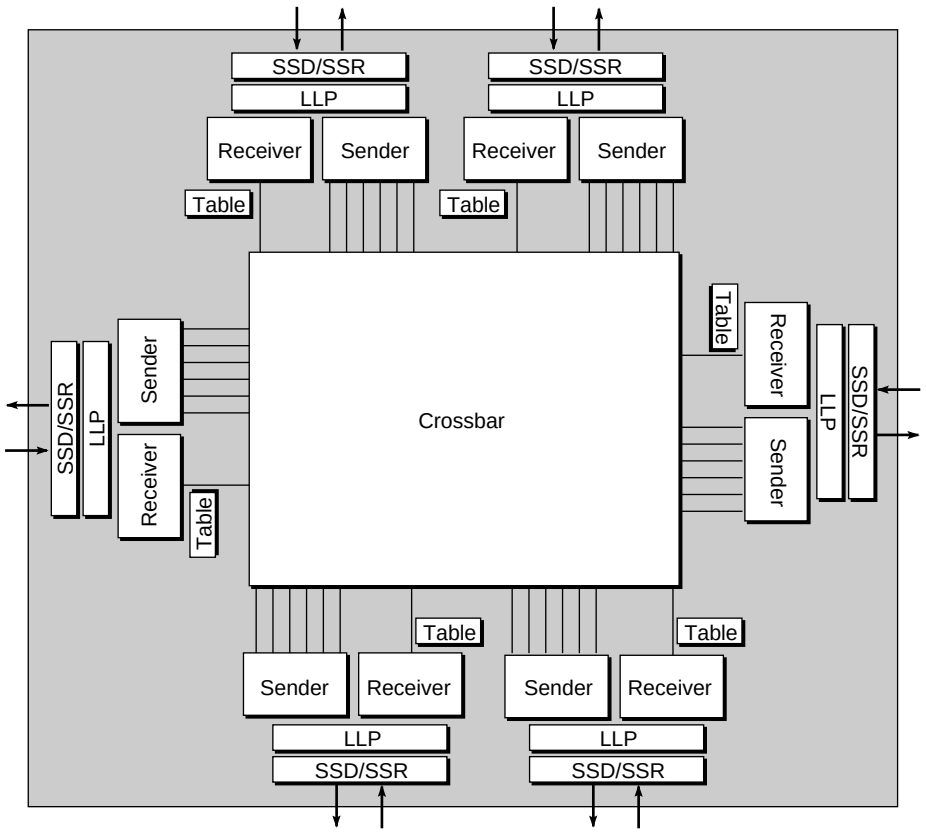

Figure 3-3 Block Diagram of the Router ASIC

As shown earlier in Figure 2-17, three of the Router ports are STL, for rapid communication between ASICs inside the enclosure. The other three Router ports are differential PECL, for communication between modules over cables outside the enclosure. The three internal router ports use single-ended STL signalling to minimize pin count. The three external router ports use differential PECL which provides better noise immunity on external links.

SSD/SSR

The source synchronous drivers/receivers create and interpret the high speed, source synchronous signals used for communication between ASICs within the enclosure. They convert 64 bits of data transmitted at the core frequency of 100 MHz to 16 bits of data transmitted at 400 MHz.

Link-Level Protocol (LLP)

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 using CCITT CRC code, and correction is made by retransmission through a sliding window protocol. Both 8- and 16-bit links are supported.

Router Receiver and Sender

The router receiver accepts data from the LLP, manages virtual channels, and forwards data to the router tables and the router sender. Dynamically Allocated Memory Queues (**DAMQs**) are used for efficient message handling under heavy loads. Bypass logic is provided for performance under light loads (see the section titled “Router Crossbar”). Logic also “ages” packets should they fail to make progress, giving higher priority to older packets.

The router sender drives data to the LLP for transmission to other chips. It also manages CrayLink Interconnect credits, which are used for flow control.

Routing Table

The routing table provides static routing information for messages as they pass through the interconnection fabric.

To minimize delays, routing table lookup is made in a pipelined fashion. Each router determines the direction the message is to take when it enters the next router. The routing table provides flexible routing and configurations with other than 2^n nodes.

Router Crossbar

The router crossbar contains a series of hand-optimized multiplexers which control data flow from receiver to sender ports. Message bypassing during periods of light loading allows a message to pass through the router with minimal latency. When bypassing is not possible, a wavefront arbiter determines the optimal path.

An “aging protocol” gives priority to older messages over those more recently arrived.

Crossbow (XBOW) ASIC

The Crossbow ASIC has a dynamic crossbar switch which expands the dual-host XIO port to six 16-bit I/O ports. Each I/O port can run in either 8- or 16-bit mode, with rate-matching buffers to decouple 16-bit to 8-bit ports. At least one Crossbow port must connect and a maximum of two Crossbow ports can connect to a host. The Crossbow ASIC uses the XIO protocol at all of its ports. There are two Crossbow ASICs on each deskside midplane.

A functional view of a Crossbow ASIC, with dual hosts and six half-size XIO boards, is shown in Figure 3-4. (CrayLink Interconnect cabling is not shown in this illustration.)

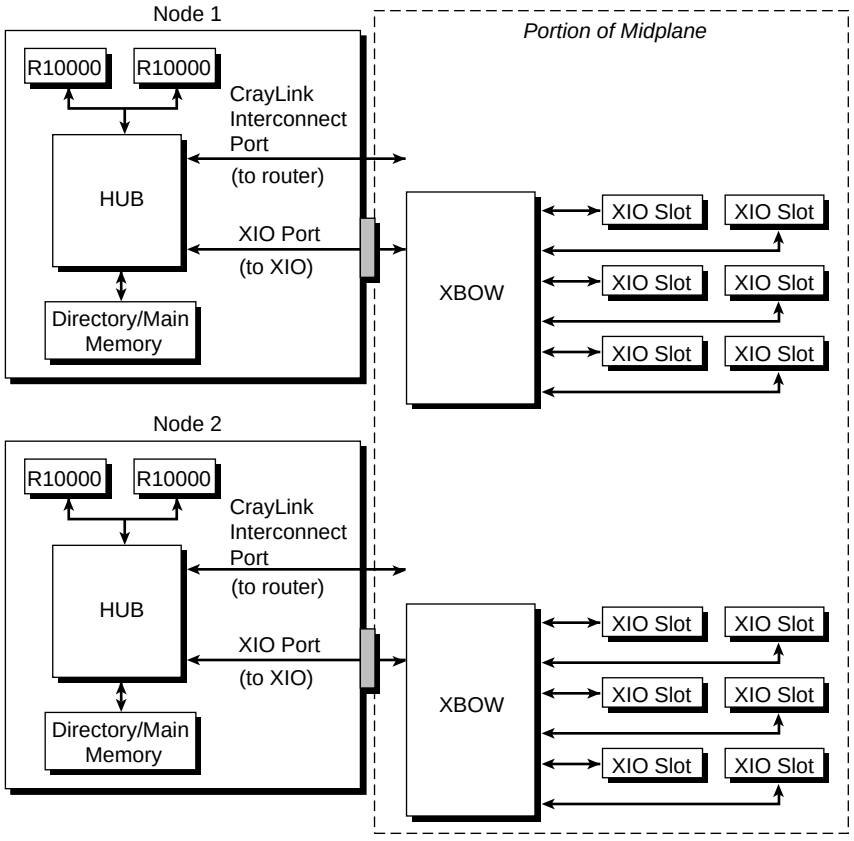

Figure 3-4 Functional Location of Crossbow ASIC

In this dual-host configuration, the six remaining XIO ports are statically partitioned between the two Node boards; if one of the host nodes is inoperable, the second host node can be programmed to take control of all the XIO ports. Note that the Node boards connect to ports 1 and 3 of the Crossbow, and the remaining six ports connect to the XIO widgets.

As described earlier, a **widget** is a generic term for any device connected to an XIO port. The Crossbow ASIC contains a crossbar switch that dynamically connects individual ports to particular I/O widgets (host, graphics board, serial I/O), as shown in Figure 3-5. The Crossbow decodes fields in XIO messages to determine control and destination information.

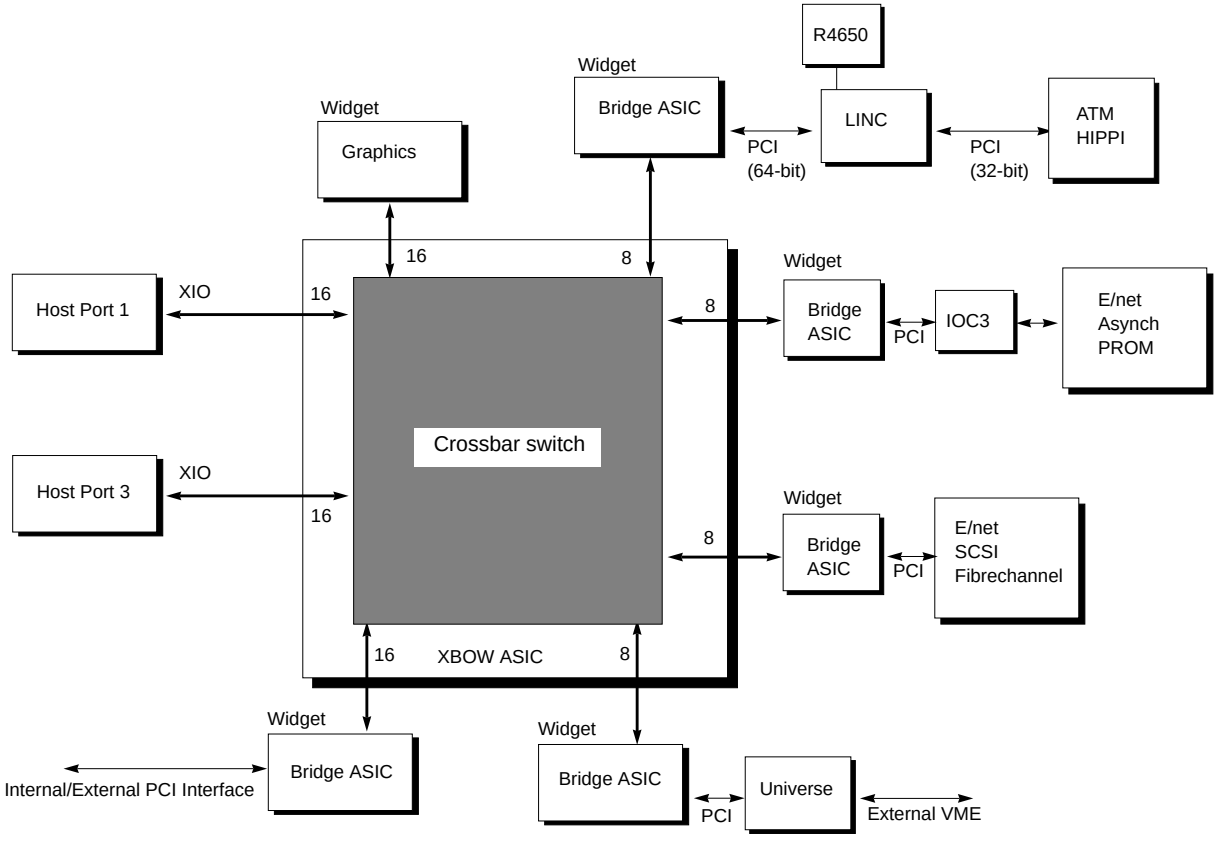

Figure 3-5 Block Diagram of a Crossbow ASIC, Showing Eight Ports Connected to Widgets

Bridge ASIC

The **Bridge** ASIC (labelled "PCI Adapter" in Figure 3-6) is physically located on an XIO board. It converts the XIO link to the PCI bus protocol. The Bridge ASIC also provides address mapping, interrupt control, read prefetching, and write-gathering.

Peak bandwidth of the Bridge ASIC is 800 MB/sec on an XIO link and 266 MB/sec on the PCI link. There is a Bridge ASIC present on every I/O widget board.

An illustration of an I/O subsystem with a Bridge ASIC is given in Figure 3-6.

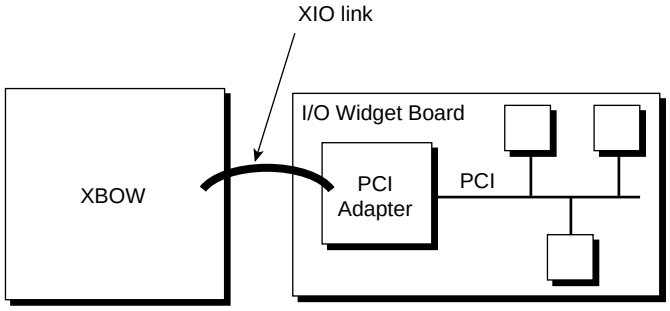

Figure 3-6 Bridge ASIC

IOC3 ASIC

The **IOC3** ASIC takes PCI output from the Bridge ASIC and converts it to standard I/O protocols for Ethernet, parallel I/O, and serial I/O. The IOC3 ASIC is physically located on the XIO BaseIO and MediaIO boards.

LINC ASIC

The LINC ASIC is designed to support intelligent controllers and optimize throughput with a variety of scatter and gather functions.

The LINC ASIC is located on the HIPPI-Serial, ATM, and OC boards; the ASIC converts the 64-bit PCI protocol to 32-bit PCI protocol. Using an IDT R4650 MIPS processor running at 132 MHz, the LINC has a 64-bit host-side "parent" PCIbus (**PPCI**) which provides:

- 64-bit addressing that is used for system DMA access
- 32-bit addressing that is used for peer-to-peer DMA and PIO accesses

The LINC ASIC also supports a 32-bit "child" PCIbus (**CPCI**) for attaching interface devices; the CPCI provides Request/Grant and interrupt support for up to two devices.

A block diagram of an I/O subsystem with both a Bridge ASIC and two LINC ASICs is given in Figure 3-7.

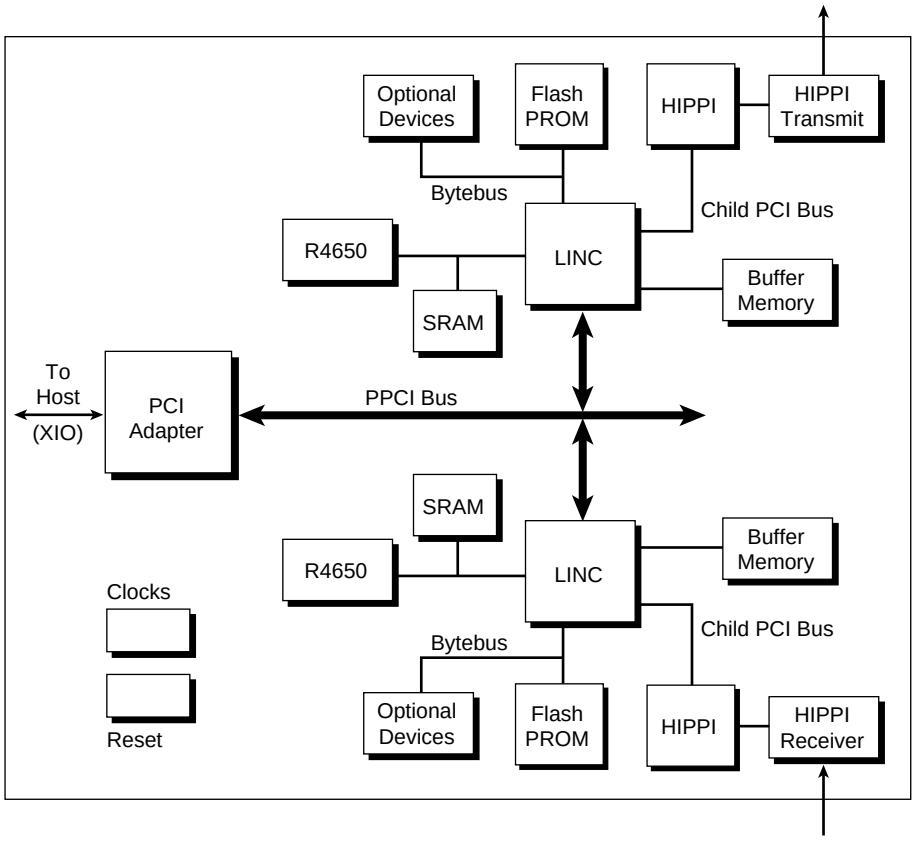

Figure 3-7 Block Diagram of LINC ASICs With Bridge ASIC

LINC ASIC also supports:

- SDRAM buffer memory for local DMA
- 4 MB to 64 MB memory configurations, using 16 Mb or 64 Mb parts
- arbitrary byte access
- two DMA engines for data movement between the PPCI and buffer memory
- an 8-bit bus (Bytebus) to both the Flash PROM and to slave peripherals
- 1-, 2-, and 4-byte access to the generic I/O device space
- 1-, 2-, and 4-byte reads and 1-byte writes to the Flash PROM
- 22 bits of address for generic devices, and 22 bits for the PROM
- optional byte parity
- one interrupt input
- interrupt dispatch logic to the local processor, structured to minimize dispatch overhead
- interrupt dispatch support to the host PCIbus
- 32 mailbox locations for host-to-firmware communication, with 8 bytes per mailbox located in buffer memory, and interrupt generation upon write. Mailboxes are on 64-KB boundaries for safe assignment to user code
- processor *Control/Status* register for host management
- mode pin to control autonomous boot; if inhibited, the host can release boot to PROM or to buffer memory
- configuration registers for PPCI and control registers for CPCI
- parity generation and checking on all ASIC boundaries (optional on the Bytebus)
- support for bi-endian communication: the R4650, SDRAM, Bytebus, and all internal registers are Big Endian; byte- and word-swapping is supported on PPCI (DMA) and CPCI (DMA and PIO) per-transaction
- four LED pins
- four generic I/O pins

Glossary

100Base-TX

See *Fast Ethernet*.

adaptive routing

A mechanism by which different messages can be routed to the same destination through different paths.

Application-Specific Integrated Circuit (ASIC)

Integrated circuit designed for a specific task.

asynchronous

Transfer of information in which the source and target are not synchronized. See also *synchronous*.

Asynchronous Transfer Mode (ATM)

Networking standard for transfer of multimedia data (voice, graphics, video). Specifies exchange of data through standardized, fixed-length cells. Because cells are uniform, they can be switched efficiently across many types of LANs and WANs.

B

byte, as in MB (megabyte)

b

bit, as in Mb (megabit)

bandwidth

Capacity to pass data. Usually measured as megabytes per second (MB/sec).

bidirectional

Transfer of information in both directions.

Bridge chip

An ASIC that converts the XIO protocol to PCI protocol. A Bridge chip is contained on an XIO board. See also *XIO*.

cache coherence

Mechanism that ensures that cached copies of memory locations are kept consistent.

Challenge

Architecture used in previous high-end systems.

CrayLink Interconnect

(1) Interconnection fabric that links Node boards. Consists of high-speed links and, for 4P and larger configurations, routers. (2) Protocol used at the message layer in CrayLink Interconnect and within the Hubs for high-speed transfer of messages between Node boards. See also *Node board, Link Level Protocol (LLP), Positive ECL (PECL), physical layer, STL,* and *Crossbow chip.*

crossbar

Switching circuit that creates multiple point-to-point connections. Also referred to as "*n* x *n* switch," or "*n*-by-*n* switch"; it keeps *n* items communicating at full-speed with *n* other items.

Crossbow chip

An 8-port ASIC that controls the dataflow through the set of XIO links. Contains a crossbar and circuits for flow control, routing, and arbitration. Depending on the type of system, one or two Crossbow ASICs are located on the module midplane. See also *crossbar*, *micropacket*, and *XIO*.

Crosstown Converter (XC) chip

ASIC that converts STL protocol to PECL, and vice versa.

Crosstown Converter board

An XIO board that connects XIO interface to an external chassis. Crosstown interface is required, for instance, to connect an Origin2000 module to a graphics chassis. Contains an XC chip and connectors for external cabling.

DIMM

Dual In-Line Memory Module

directory memory

A dedicated memory array on each Node board which supports the Origin2000 cache-coherence protocol (directory-based cache coherence). The system uses directory memory to track caches that contain a particular data item. If a change is made to the data item, all processors that are caching that location and that are indicated by the directory are notified. See also *Distributed Shared-Memory,* and *cache-coherence*.

Distributed Shared-Memory (DSM)

Main memory that is both distributed and shared. For systems that employ DSM, sections of main memory are distributed with each processor. Although distributed, each section of main memory is accessible by all processors through an interconnection fabric.

Fast Ethernet

Any of several standards for 100 Mbit/sec Ethernet (for example, 100Base-T, 100Base-TX). Each uses the same collision detection scheme (CSMA/CD) but uses a different transmission medium or message format.

Fast 20

A 16-bit wide differential SCSI bus running at a peak transfer rate of 40 MB/sec.

Fiber Distributed Data Interface (FDDI)

Networking standard for a fiber-optic LAN. Specifies a ring network with up to 1000 access points. The circumference of ring can be up to 120 miles, and the maximum baud rate is 100 Mbit/sec.

full-size (FS) XIO board

Form-factor (the shape) of a full-size I/O board. Dimensions of an FS XIO board are 10.5 x 1 3x 1 in.

g

giga (billion): 10^9, or 1,000,000,000, as in gB (one billion bytes)

G

giga: 2^{30}, or 1,073,741,824, as in GB (1,073,741,824 bytes)

half-size (HS) XIO board

Form-factor (the shape) of a half-size I/O board. Dimensions of an HS XIO board are 10.5 x 6.5 x 1 in.

High Performance Parallel Interface (HIPPI)

A high-speed interface used over relatively short distances. It was developed at Los Alamos National Laboratory and is now ANSI-standard X3T9/88-127. HIPPI is ideal for transfer of large volumes of data.

HIMM

Horizontal In-Line Memory Module. In a deskside or rackmounted configuration, this module holds processor and memory and is connected to the Node board.

hop count

Number of Router ASICs a message must pass through to go from source to target node. See also *micropacket* and *Router board*.

Hub chip

ASIC that is the interface controller on Node board. It contains four major Node board subsystems: processor interface(s), memory and directory interface, I/O (XIO) interface, and CrayLink Interconnect. See also *CrayLink Interconnect*, and *XIO*.

hyper-

Existing in a space of one or more dimensions. See also *hypercubes*.

hypercubes

Mathematical model that defines *n*-dimensional cubes. For Origin2000 configurations, all topologies are either a hypercube (up to 64P) or fat hypercube (fatcube; 128P).

interconnection fabric

A set of switches in a given topology (hypercube, mesh, etc.) that interconnect the nodes of a system.

IOC3 chip

ASIC that converts the PCI protocol to standard I/O protocols; for example, converts PCI to Ethernet, serial, or parallel. The IOC3 receives its input from the Bridge ASIC. See also *Bridge chip* and *IO6 board*.

I06 board

An XIO board that provides Origin2000 with basic I/O functions; for example, Ethernet, serial, and SCSI ports. Can contain the Bridge, MIO, or IOC3 chip. See also *XIO*.

IP27

Product title for the Node board on Origin2000 System.

IP29

Product title for the motherboard on the Origin200 System.

k

kilo (thousand): 10^3, or 1,000, as in kB (one thousand bytes)

K

kilo: 2^{10}, or 1024, as in KB (1024 bytes)

LAN (Local Area Network)

A geographically-limited data communications network that allows interconnection of terminals, microprocessors and computers, usually within physically-adjacent buildings. Ethernet and FDDI are examples of LANs.

latency

Amount of time it takes to complete a request and receive a reply (typically to memory, or to an I/O device).

LINC chip

An ASIC designed to support intelligent controllers on some XIO devices; for instance, ATM, HIPPI.

Link Level Protocol (LLP)

A protocol that handles error-checking and error-recovery to ensure error-free transmission of data across all CrayLink Interconnect and XIO links. See *XIO*, *CrayLink Interconnect*, *Router chip*.

m

mega (million): 10^6, or 1,000,000, as in mB (one million bytes)

M

mega: 2^{20}, or 1,048,576, as in MB (1,048,576 bytes)

MAC

Media Access Control; messages sent over a LAN have a unique six-byte identifier called a MAC.

message

Unit of information transfer.

messaging

Means by which information is transferred; or, the act of transferring information.

Meta Router

A module that contains up to eight Router boards and is used to form the metacube level of interconnection in Origin2000 for systems having more than 32 routers.

micropacket

Format of data transferred over the link layer within CrayLink Interconnect and XIO. A micropacket is the minimum size of data transfer, and error-checking is made over micropacket quantities. A micropacket consists of two 64-bit doublewords of data, and two 16-bit halfwords of control information. The control information includes check bits, sequence numbers (transmit and receive), and an 8-bit sideband.

The data portion of a micropacket contains a message header and optional message data. The 8-bit sideband is used by the message layer for framing and flow control. Together, the sideband and data portion of a micropacket can be considered a *message segment*. Micropackets are encoded when a message is injected into the LLP module which encapsulates the CrayLink Interconnect and XIO links.

When passing from the link layer to the physical layer, micropackets are converted from 64 bits to 16 bits. This 16-bit data is transmitted at 400 MHz, which is four times the core data rate of 100 MHz, by the Source Synchronous Driver (SSD). The data is demultiplexed by the Source Synchronous Receiver (SSR).

modular

Constructed from standardized units. In the case of Origin2000, modules are configurable in the field.

module, Origin2000

Enclosure that contains Origin2000 components, such as Node, XIO, graphics, and Router boards. There are two configurations of Origin2000 modules: server and graphics.

multiprocessing

Simultaneous processing by two or more processors in one computer system.

node

Functional unit in Origin2000 module that contains processors, memory, XIO interface, and CrayLink Interconnect interface. Physically, a node is implemented on a Node board.

Node board

Board that contains a node of the Origin2000 system. Each Node board has one or two R10000 processors with their cache(s), a section of main memory with its dedicated directory memory, an interface to system I/O, and an interface to the CrayLink Interconnect. Subsystems on the Node board are connected through a large ASIC called the Hub chip. See also. See also *Hub chip*, *R10000 chip*, *CrayLink Interconnect*.

Non-Uniform Memory Access (NUMA)

A characteristic of DSM systems. In DSM systems, sections of main memory are located at various distances from a given processor. As a result, memory access times (latencies) can be *non-uniform*. See also *Distributed Shared-Memory* and *latency.*

null-Router

Jumper board used to connect two Node boards directly to each other in a 4P system.

PCAR

External expansion module for industry-standard PCI boards.

Peripheral Component Interconnect (PCI)

I/O standard for connection of system peripherals. Specifies a PCI bus whose major features include: multiple bus masters (boards arbitrate to "own" the bus); autoconfiguration (bus is configured automatically); high peak bandwidth (264 MB/sec); interrupt sharing among boards.

PIMM

Processor Included Memory Module. In an entry-level configuration, this module holds processor and memory and is connected to the Node board.

physical layer

Layer within CrayLink Interconnect and XIO protocols that establishes physical link over the transmission medium. See also *CrayLink Interconnect*, *transmission medium,* and *XIO.*

Positive ECL (PECL)

Differential signal levels that are used in external cables for CrayLink Interconnect and XIO (Crosstown) interconnections.

protocol

Set of rules that governs the transfer of information.

R10000 chip

Fifth-generation MIPS Rx000-series processor. Also called the T5. The R10000 is based on RISC 64-bit 4-way superscalar technology and is contained on a single chip (whereas, for instance, the R8000® processor is a multichip module).

Router board

Board that contains the Router chip, associated circuits and a CrayLink Interconnect port. For expandable systems, one Router board is required for each two nodes. See also *Router chip*.

Router chip

ASIC that routes messages on CrayLink Interconnect. Contains crossbar and circuits for flow control, routing, and arbitration, and is physically located on a Router board. See also *crossbar, Router board*, and *CrayLink Interconnect.*

scalable

A situation in which one item increases in proportion to another. For example, CrayLink Interconnect is scalable; as you add nodes to the interconnection fabric, you increase CrayLink Interconnect capacity and performance. Origin2000 systems are scalable with respect to cost; an entry-level system has a low fixed cost, and the system cost scales as you add more processors.

SGI Transistor Logic (STL)

High-speed, low-voltage, unidirectional signals used for communication between ASICs inside a module. STL signals can be converted to PECL signals for communicating between modules over a cable. See also *Positive ECL (PECL)*, *physical layer*, and *XIO*.

SIMM

Single In-line Memory Module

source

Sender of a message. See also *target*.

source synchronous

A transmission in which clock accompanies data.

STL

See *SGI Transistor Logic (STL)*

synchronous

A transmission of data in which both source and target are synchronized.

system, Origin200

An Origin2000 system is composed of one or two towers, within which are either one or two processor/memory daughtercards mounted on a motherboard and containing main and directory memory. The Origin200 system has three PCI slots. In a dual-tower configuration, the two towers can be linked by the CrayLink Interconnect.

system, Origin2000

An Origin2000 system is composed of one or more server or graphics modules, within which processor/memory nodes are linked by an interconnection fabric.There are two types of system: deskside, and rackmounted. See also *module, Origin2000.*

system controller

A board that monitors and controls system functions (for example, system cooling and power consumption). There are two levels of system controller: entry-level, and Multi-Module System Controller.

T5

See *R10000 chip.*

target

Receiver of a message. See also *source.*

topology

A pattern of interconnection between nodes. In Origin2000, *n*-dimensional versions of a hypercube topology are used to link nodes into a coherent system.

transmission medium

Conductor(s) over which signals are passed.

VME

I/O standard for connection of system peripherals. Specifies a bus called VMEbus. Predates PCI.

widget

A generic term for any device connected to an XIO port.

Wormhole Routing

In wormhole routing, a message that passes through the switches of the interconnection network is referred to as a “worm,” since it can be segmented into micropackets. The worm can stretch across several nodes and links at any one time. As soon as the head is received, each intermediate switch moves it towards the intended port without waiting for the entire message to arrive. See also *adaptive routing*.

XIO

(1) The interconnection in each Origin2000 module that provides Node boards with access to I/O devices. Controlled by Crossbow chip. (2) Protocol used on XIO interface. Implements high-speed transfer of messages across connections established by Crossbow. Runs over LLP at the link layer and STL at the physical layer. See also *crossbar*, *Crossbow chip*, *micropacket,* and *CrayLink Interconnect.*

Index

A

access time, 48
address space
  in a Node's memory, 34
  Origin2000 system, 25
architecture
  load/store, 35

B

bandwidth
  bisection, 27, 28
  Origin2000 system, 27
  peak, 27, 28
  peripheral, 29
  sustained, 27, 28
BaseIO board
  description, 67
BaseIO-G board, 70, 71
bisection bandwidth, 27, 28
  in Origin2000 system, 20
bit vectors
  as used in coherence, 43
blocks
  as used in memory, 37
  memory, 34
Block Transfer Engine (BTE), 49
Bridge ASIC, 76
  XIO link, 86
BTE, Block Transfer Engine, 49
Busy
  directory state, 45

C

cache
  coherence
    description of, 41
    in Origin2000 system, 19
  hit, 40
  in the Origin2000 system, 25
  memory hierarchy, 25, 26
clock, global, 51
coherence
  as it uses invalidation, 44
  bit vectors, 43
  directory-based, 42
  directory entry, 43
  hardware, 44
  snoopy-based, 42
  state bits, 43
consistency, sequential, 45
CrayLink Interconnect
  Origin2000 system, 17
crossbar
  Crossbow ASIC, 85
  Origin2000 system, 21, 23
  Router ASIC, 83
  Router board, 55

Crossbow ASIC, 53, 76, 84
  configuration, 54
  connections, 9
  crossbar, 85
  I/O Ports, 84
  Origin2000 system, 17
  widget, 85
Crosstown board
  description, 73
Crosstown protocol, 53
cycle time, 48
  STL, 57

D

DAMQ, Dynamically Allocated Memory Queue, 82
daughterboard, Origin200, 5
devices, XIO, 53
DIMM
  (dual in-line memory module), 35
directory-based coherence protocol, 42, 43
directory entry, 41
directory memory, 35
  configuration, 36
  Origin2000 system, 17
directory poisoning, 51
directory states
  Busy, 45
  Exclusive, 45
  Poisoned, 45
  Shared, 45
  Unowned, 45
distributed I/O, 53
distributed shared-memory (DSM), 34
  as used in Origin2000 system, 19
DSM
  (distributed shared-memory), 34
dual in-line memory module (DIMM), 35
Dynamically Allocated Memory Queue (DAMQ), 82

E

Exclusive
  directory state, 41, 45
Express Links, 59
extended directory memory, 35

F

fault in memory, 40

H

hardware coherence, 44
HIMM
  (horizontal in-line memory module), 33
  Node board, 33
hit in cache, 40
home memory
  in the Origin2000 system, 25
  memory hierarchy, 25
horizontal in-line memory module (HIMM), 33
Hub ASIC, 76
  cache coherence, 79
  description, 32, 33
  interfaces, 77
  in the Origin2000 system, 17
  intranode communications, 78

I

indexing
  as used in memory, 37
interconnection fabric, 3, 21
  parallel datapaths, 4
  performance versus a bus, 4
  system interconnections, 19
invalidation
  as used in coherence, 44
I/O
  Bridge ASIC, 86
  Crossbow ASIC, 84
  distributed, 53
  in the Origin2000 system, 17
  Node board, 51
  statically-partitioned with HUB ASIC, 79
IOC3 ASIC, 76, 87

L

latency
  access time, 48
  cycle time, 48
  memory, 48
  system, 48
LINC ASIC, 87
line
  as used in memory, 37
link level protocol (LLP), 82
load/store architecture, 35
locality
  of reference, 48
  spatial, 48
  temporal, 48
local memory, 26, 34

M

mapping
  memory, 38
maximum configuration
  Origin2000 system, 20
  Origin200 system, 5
MediaIO board
  description, 71
memory
  access time, 48
  address space, 34
  blocks, 34, 37
  consistency, 45
  cycle time, 48
  directory, 35, 36
  distributed shared-, 34
  extended directory, 35
  indexing, 37
  in the Origin2000 system, 16
  latency, 48
  line, 37
  local, 34
  mapping, 38
  Origin200, 5
  page fault, 40
  pages, 34, 37
  page table, 39
  page table entry, 39
  read cycle, 46
  remote, 34
  slots, number of, 36
  virtual, 38
  write cycle, 46
memory hierarchy, 25, 26
  cache, 25, 26
  home memory, 25
  in the Origin2000 system, 25
  local memory, 26
  processor registers, 25
  remote caches, 25

Meta Router, 57
Midplane board
  description, 63
migration, page, 49
modularity, 5
  Origin2000 system, 18, 20
motherboard, Origin200, 5

N

Node board
  block diagram, 32
  cache, 33
  description, 32
  global clock, 51
  HIMM, 33
  Hub ASIC, 33
  in the Origin2000 system, 9
  I/O, 51
  memory, 34
  physical connections, 51
  processors, 33
Null Router, 57

O

Origin2000 system
  address space, 25
  bandwidth, 27
  BaseIO board, 67
  bisection bandwidth, 20
  cache, 25
  cache coherence, 19
  CrayLink Interconnect, 17
  crossbar, 21, 23
  Crossbow, 17
    connections, 9
  Crosstown board, 73
  directory coherence protocol, 19
  directory memory, 17
  distributed shared-memory, 19
  home memory, 25
  Hub ASIC, 17, 32, 33
  interconnection fabric, 3, 19, 21
  I/O, 17
  maximum configuration, 20
  MediaIO board, 71
  memory, 16
    hierarchy, 25
  Midplane board, 63
  modularity, 5, 18, 20
  Node board, 9, 32
  overview, 9
  page migration, 19
  PCI bus, 17
  processing node, 3
  R10000 processors, 11, 16
  registers, 25
  remote caches, 25
  Router ASIC, 23
  Router board, 55
  scalability, 5, 18, 20
  snoopy-based coherence protocol, 19
  system interconnections, 19
  what the system consists of, 1
  XIO boards, 9
  XIO interfaces, 17
Origin200 system
  daughterboard, 5, 73
  maximum configuration, 5
  memory, 5
  motherboard, 5, 73
  overview, 5
  PCI expansion slots, 5
  R10000 processors, 5
  what the system consists of, 1
overview, of the Origin2000 system, 9
overview, of the Origin200 system, 5

P

page fault in memory, 40
page migration, 49
  Origin2000 system, 19
page replication, 49
pages
  as used in memory, 34, 37
page table
  as used in memory, 39
page table entry (PTE), 39
parallel datapaths, 4
PCI expansion slots
  in the Origin2000 system, 17
  in the Origin200 system, 5
PCI protocol, 76
peak bandwidth, 27, 28
peripheral bandwidth, 29
physical connections, Node board, 51
Poisoned
  directory state, 45
poisoning, directory, 51
processing node, 3
processor registers, 25
processors
  Origin2000 system, 11
  Origin200 system, 5
protocol
  Crosstown, 53
  link level, 82
  PCI, 76
  XIO, 53
protocol, cache coherence, 41
  directory entry, 41
  Exclusive state, 41
  Shared state, 42
protocol, directory, 42, 43
protocol, snoopy, 42, 43
PTE, page table entry, 39

R

R10000 processors
  in the Origin2000 system, 16
read cycle, 46
registers
  in the Origin2000 system, 25
remote caches
  in the Origin2000 system, 25
  memory hierarchy, 25
remote memory, 34
replication, page, 49
Router ASIC, 76, 80
  crossbar, 83
  Dynamically Allocated Memory Queue (DAMQ), 82
  link level protocol, 82
  Origin2000 system, 23
  receiver, 82
  routing table, 83
  sender, 82
  source synchronous driver/receiver, 82
Router board
  connectors, 58
  CrayLink interconnections, 58
  crossbar, 55
  description, 55
  Express Links, 59
  Meta Router, 57
  Null Router, 57
  Standard Router, 57
  Star Router, 57
routing table
  Router ASIC, 83

S

S2MP architecture, 1
  used for distributing memory, 5
scalability, 5
  Origin2000 system, 18, 20
Scalable Shared-memory MultiProcessing architecture (S2MP), 1
SDRAM, 36
sequential consistency, 45
SGI Transistor Logic (STL), 57
Shared
  directory state, 42, 45
snoopy-based coherence protocol, 42, 43
  in the Origin2000 system, 19
spatial locality, 48
Standard Router, 57
Star Router, 57
state bits
  coherence, 43
STL, SGI Transistor Logic, 57
sustained bandwidth, 27, 28
system latency, 48

T

temporal locality, 48
TLB, translation lookaside buffer, 39, 40
translation lookaside buffer (TLB), 39, 40

U

Unowned
  directory state, 45

V

virtual memory, 38

W

widgets (XIO), 53
  Crossbow ASIC, 85
write cycle, 46

X

XIO
  Crossbow ASIC, 53
  distributed I/O, 53
  Origin2000 system, 17
  widgets, 53
XIO boards
  in the Origin2000 system, 9
XIO devices, 53
XIO protocol, 53

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-3439-002.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
  - On the Internet: techpubs@sgi.com
  - For UUCP mail (through any backbone site): *[your_site]*!sgi!techpubs
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 415-965-0964
- To send your comments by **traditional mail**, use this address:

  Technical Publications
  Silicon Graphics, Inc.
  2011 North Shoreline Boulevard, M/S 535
  Mountain View, California 94043-1389