System Architecture
SGI™ Origin™ 3000 Server Series

SGI Confidential & Proprietary Information - For Internal Recipients Only

Document Number 108-0240-002

CONTRIBUTORS

Written by Nancy Heller
Illustrated by Kelly Begley and Nancy Heller
Edited by Allison Gosbin
Production by Rhonda Kunsman

INFORMATION CLASSIFICATION

LIMITED RIGHTS LEGEND

TRADEMARKS

Record of Revision

Version **Description**

001 July 2000

Original printing.

002 December 2000

This revision includes the following technical changes:

- Added a note about the 6- and 8-port R bricks
- Added host interface cards to the P- and X-brick block diagrams
- Added InfiniteReality as a supported board set of the Silicon Graphics Onyx3 graphics system
- Changed the color of the Service Required LED on the front panel of the L1 controller
- Updated the serial number information
- Reduced the number of power supplies in the I/O rack power bay from 5 to 4

Contents

Figures

Tables

Chapter 1

System Features

The SGI Origin 3000 series servers consist of compute nodes, routers, I/O interfaces, and peripheral devices. The compute nodes are linked together by a NUMAlink 3 interconnect that uses routers where required. The compute nodes connect to I/O interfaces by using a Crosstalk2 I/O protocol.

The architecture of the SGI 3000 series servers has the following features:

- Modularity
- Scalability
- Distributed shared memory
- Distributed shared I/O
- Cache-coherent nonuniform memory access
- Reliability, availability, and serviceability (RAS)

1.1 Modularity and Scalability

The SGI 3000 series servers are scalable systems, which means that the customers can scale the system in independent dimensions: computing, I/O, and storage. For example, the computing dimension can range from 2 to 512 processors. The SGI 3000 series servers can also be clustered together to increase the number of processors from 512 to thousands of processors.

The SGI 3000 series servers are also modular systems, which means that the components are housed in building blocks called bricks. These bricks can be added to a system to achieve the desired system configuration. As bricks are added to a system, the bandwidth and performance scale in a manner that is almost linear without significantly affecting system latencies.

1.2 Distributed Shared Memory (DSM)

In SGI 3000 series servers, memory is physically distributed among compute nodes; however, it is accessible to and shared by all compute nodes. The memory that is physically located on a compute node is referred to as local memory; all other memory is referred to as remote memory and is accessed via the NUMAlink 3 interconnect. The total memory within the system is referred to as global memory.

For example, Figure 1-1 shows global memory for a system that contains four compute nodes. Compute node 0 has some memory that is local to its node. Compute nodes 1, 2, and 3 refer to this memory as remote memory and must use the NUMAlink 3 interconnect to access it.

The memory latency, which is the amount of time it takes to retrieve data from memory, is lowest when a processor accesses memory that is local to its compute node.

1.3 Distributed Shared I/O

Like DSM, I/O devices are distributed among the compute nodes (each compute node has an I/O port that can connect to an I/O interface) and are accessible by all compute nodes.

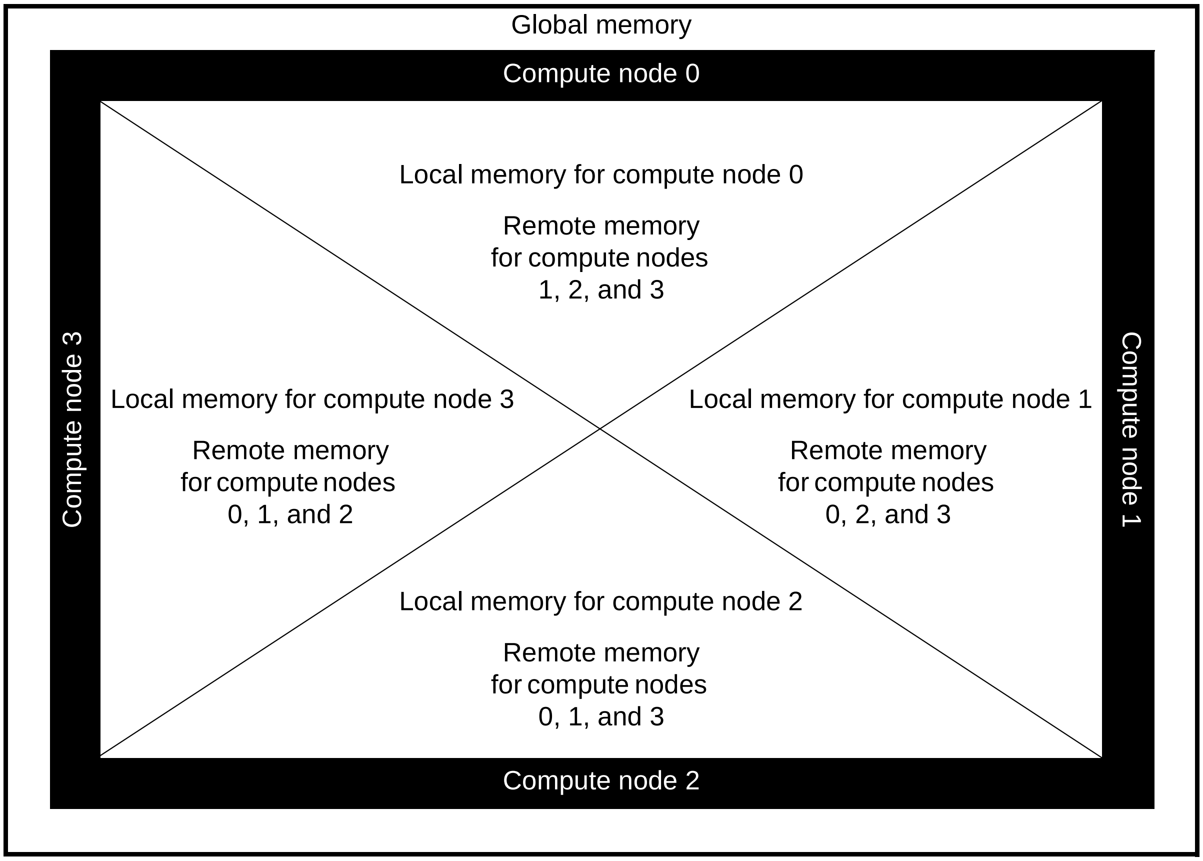

Global memory
Compute node 0
Local memory for compute node 0
Remote memory for compute nodes 1, 2, and 3
Compute node 3
Local memory for compute node 3
Remote memory for compute nodes 0, 1, and 2
Compute node 1
Local memory for compute node 1
Remote memory for compute nodes 0, 2, and 3
Local memory for compute node 2
Remote memory for compute nodes 0, 1, and 3
Compute node 2

Figure 1-1 Distributed Shared Memory

1.4 Cache-coherent Nonuniform Memory Access Architecture

The SGI 3000 series servers support a cache-coherent nonuniform memory access architecture, which is referred to as SGI NUMA (formerly ccNUMA). There are two parts to the SGI NUMA architecture: cache coherency and nonuniform memory access.

1.4.1 Cache Coherency

The SGI 3000 series servers use caches to reduce memory latency. While data only exists in local or remote memory, copies of the data can exist in various processor caches. Cache coherency keeps the cached copies consistent.

To keep the copies consistent, the SGI NUMA architecture uses directory-based coherence protocol. In directory-based coherence protocol, each block of memory (128 bytes) has an entry in a table that is referred to as a directory. Like the blocks of memory that they represent, the directories are distributed among the compute nodes.

Note: A block of memory is also referred to as a cache line.

Each directory entry indicates the state of the memory block that it represents. For example, when the block is not cached, it is in an unowned state. When only one processor has a copy of the memory block, it is in an exclusive state. And when more than one processor has a copy of the block, it is in a shared state; a bit vector indicates which caches contain a copy.

When a processor modifies a block of data, the processors that have the same block of data in their caches must be notified of the modification. The SGI 3000 series servers use an invalidation method to maintain cache coherence. The invalidation method purges all unmodified copies of the block of data and the processor that wants to modify the block receives exclusive ownership of the block.

1.4.2 Nonuniform Memory Access (NUMA)

In DSM systems, memory is physically located at various distances from the processors. As a result, memory access times (latencies) are different or *nonuniform*. For example, it takes less time for a processor to reference its local memory than it does to reference remote memory.

In a NUMA system, program performance is based on proper placement of important data structures. In general, data should be located close to the processor that will access it. If the operating system and application do not place the data correctly, the SGI 3000 series servers use page migration to move frequently accessed data closer to the processor that is accessing it.

1.5 Reliability, Availability, and Serviceability (RAS)

The SGI 3000 series server components have the following features to increase the reliability, availability, and serviceability of the systems:

- Power supplies are redundant and can be hot swapped
- Bricks have over-current protection
- Fans are redundant, can be hot swapped, and run at multiple speeds (speed increases when temperature increases or when a single fan fails)
- Controllers monitor the internal power and temperature of the bricks
- Controllers automatically shutdown bricks to prevent overheating
- Memory and secondary cache are protected by SECDED (single-bit error correction and double-bit error detection)
- NUMAlink 3 interconnect network is protected by cyclic redundancy check (CRC)
- Primary cache is protected by parity
- Each brick has failure LEDs that indicate the failed part
- LEDs are readable via the system controllers
- System controllers can perform monitoring and maintenance activities
- PCI cards can be added to the system without powering down the brick
- IRIX has improved reliability
- System has a local FRU analyzer
- All system faults are logged in files
- Memory can be scrubbed when a single-bit error occurs
- Automatic testing occurs after you power up the system (power-on self tests or POST)

  Note: These tests are also referred to as power-on diagnostics or POD.
- Boot times are minimized
- System supports remote console and maintenance activities
- System supports partitioning
- System supports ESP (Embedded Support Partner), which is a tool that monitors the system; when a condition occurs that may cause a failure, it notifies the appropriate personnel

Chapter 2

System Functionality

The functionality of the SGI Origin 3000 series servers is provided by the following components: compute nodes, NUMAlink 3 interconnect, I/O interfaces, peripheral devices, and graphics systems (refer to Figure 2-1). The SGI 3000 series servers also contain system controllers that monitor and control the system.

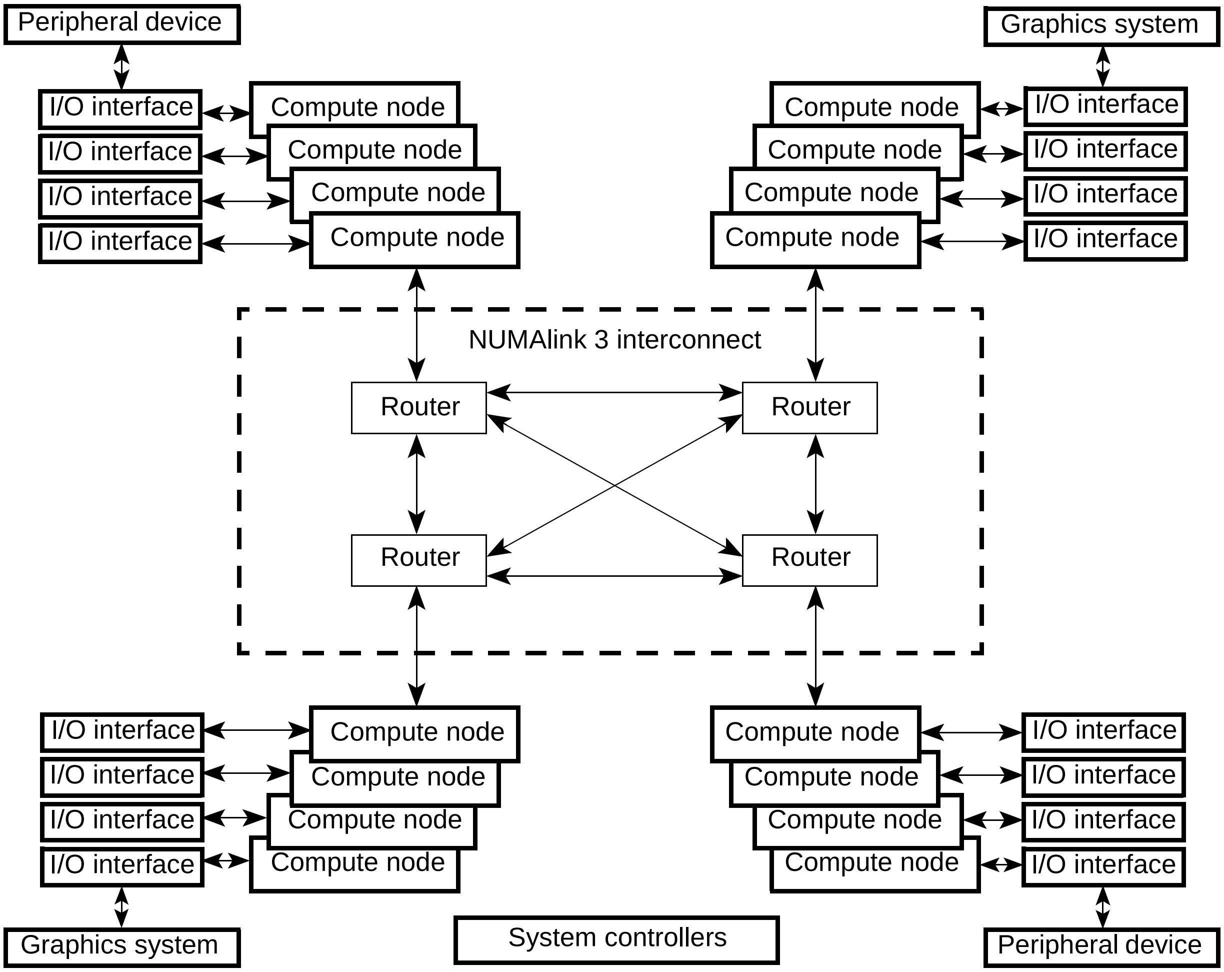


Figure 2-1 SGI 3000 Series Servers Block Diagram

2.1 Compute Node

The compute node, which is also referred to as the C brick, provides the compute functionality for the SGI 3000 series servers (refer to Figure 2-2). It contains:

- Two or four processors
- Primary and secondary cache
- Local memory (main memory and directory memory)
- A hub application-specific integrated circuit (ASIC)

Note: The hub is also referred to as Bedrock.

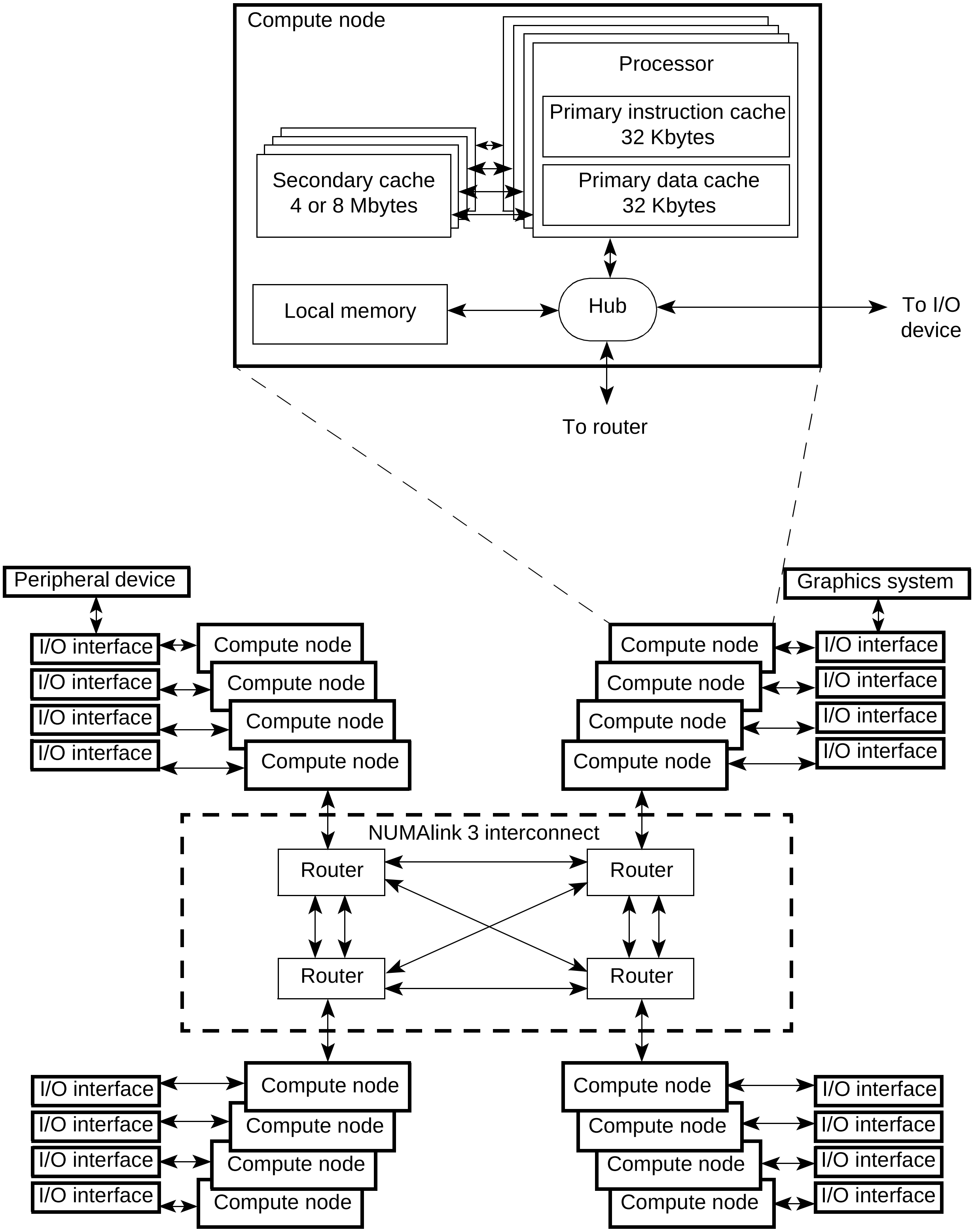


Figure 2-2 SGI 3000 Series Server Block Diagram with Detailed Compute Node (C Brick)

2.1.1 Processor

The SGI 3000 series servers support the MIPS R12000 and MIPS R12000A processors and their successors. The R12000 processor operates at 360 MHz; the R12000A processor operates at 400 MHz.

An SGI 3000 series processor implements the 64-bit MIPS-IV instruction set architecture. It fetches and decodes four instructions per cycle and issues the instructions to five fully pipelined execution units. It predicts conditional branches and executes instructions along the predicted path.

An SGI 3000 series processor uses a load / store architecture, in which the processor does not operate on data that is located in memory; instead, it loads the memory data into its registers and then operates on the data. When the processor is finished manipulating the data, the processor stores the data back in memory.

The processors are physically located on processor integrated memory modules (PIMMs). Each PIMM contains two processors. Each compute node can contain one or two PIMMs (two or four processors).

2.1.2 Primary and Secondary Cache

To reduce memory latency, the processor has access to two on-chip 32-Kbyte primary caches (one cache is for data and the other cache is for instructions) and an off-chip secondary cache. The primary caches are located within the processor for fast, low-latency access of instructions and data. The secondary cache, which is located on the PIMM, consists of 4 or 8 Mbytes of standard synchronous static random access memory (SSRAM). The size of the secondary cache depends on the processor type.

2.1.3 Local Memory

Each compute node has from 512 Mbytes to 8 Gbytes of local memory, which includes main memory and directory memory for cache coherence. Local memory can consist of 1 to 8 banks, which are referred to as banks 0 through 7 (refer to Figure 2-3). Two banks are composed of two dual-inline memory modules (DIMMs) that contain double data rate synchronous dynamic random access memory (DDR SDRAM). For example, Banks 0 and 1 reside on DIMM 0 and DIMM 1.

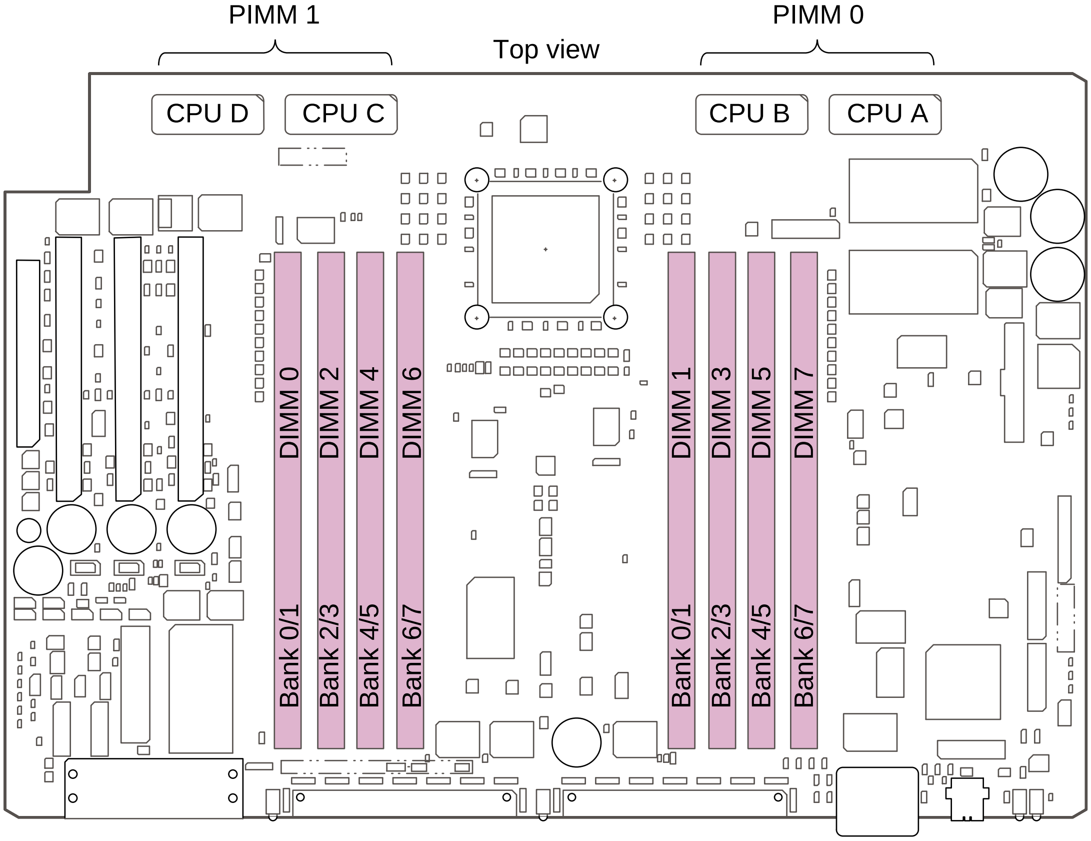


Figure 2-3 Memory Banks of a Compute Node

The SGI 3000 series servers support three DIMM sizes: 256 Mbyte, 512 Mbyte, and 1 Gbyte. The 256-Mbyte DIMM is referred to as a standard DIMM. The 1-Gbyte DIMM is referred to as a premium DIMM. The 512-Mbyte DIMM can be a standard or premium DIMM.

Premium DIMMs are required by large systems (more than 128 processors) that need additional directory memory; however, these DIMMs can be used in any system configuration. Table 2-1 lists the specifications for the three DIMM sizes.

You can increase or decrease the size of memory by adding or removing the two DIMMs that compose a bank pair (for example, Banks 0 and 1, Banks 2 and 3, Banks 4 and 5, or Banks 6 and 7). The two DIMMs that make up a bank pair must be the same size; however, each of the four bank pairs within a compute node can be a different memory size.

Table 2-1 Memory DIMM Specifications

DIMM Capacity	Number of Chips per DIMM	Chip Capacity	Chip Width	Total Memory Capacity
256 MB	18 chips for main memory 1 chip for standard directory memory	128 Mbit	8 bits for main memory 16 bits for directory memory	2 banks - 512 MB 4 banks - 1 GB 6 banks - 1.5 GB 8 banks - 2 GB
512 MB	36 chips for main memory 2 chips for standard directory memory 4 chips for premium directory memory	128 Mbit	4 bits for main memory 8 bits for directory memory (standard or premium)	2 banks - 1 GB 4 banks - 2 GB 6 banks - 3 GB 8 banks - 4 GB
1 GB	36 chips for main memory 4 chips for premium directory memory	256 Mbit	4 bits for main memory 8 bits for directory memory	2 banks - 2 GB 4 banks - 4 GB 6 banks - 6 GB 8 banks - 8 GB

The DIMM data path to main memory is 144 bits wide: 128 data bits and 16 error correction code (ECC) bits (refer to Figure 2-4). The standard DIMM data path to directory memory is 16 bits. The premium DIMM data path to directory memory is 32 bits.

Note: For directory memory, a word is 32 bits (standard) or 64 bits (premium). These bits are retrieved by two read operations. For example, two 16-bit read operations occur from the standard DIMMs to retrieve the 32-bit directory word; six of these bits are ECC bits. For the premium DIMMs, two 32-bit read operations retrieve the 64-bit word; seven of these bits are ECC bits.

The control signals instruct the DIMMs to read data from or write data to the memory chips. The address indicates where the data should be read from or written to.

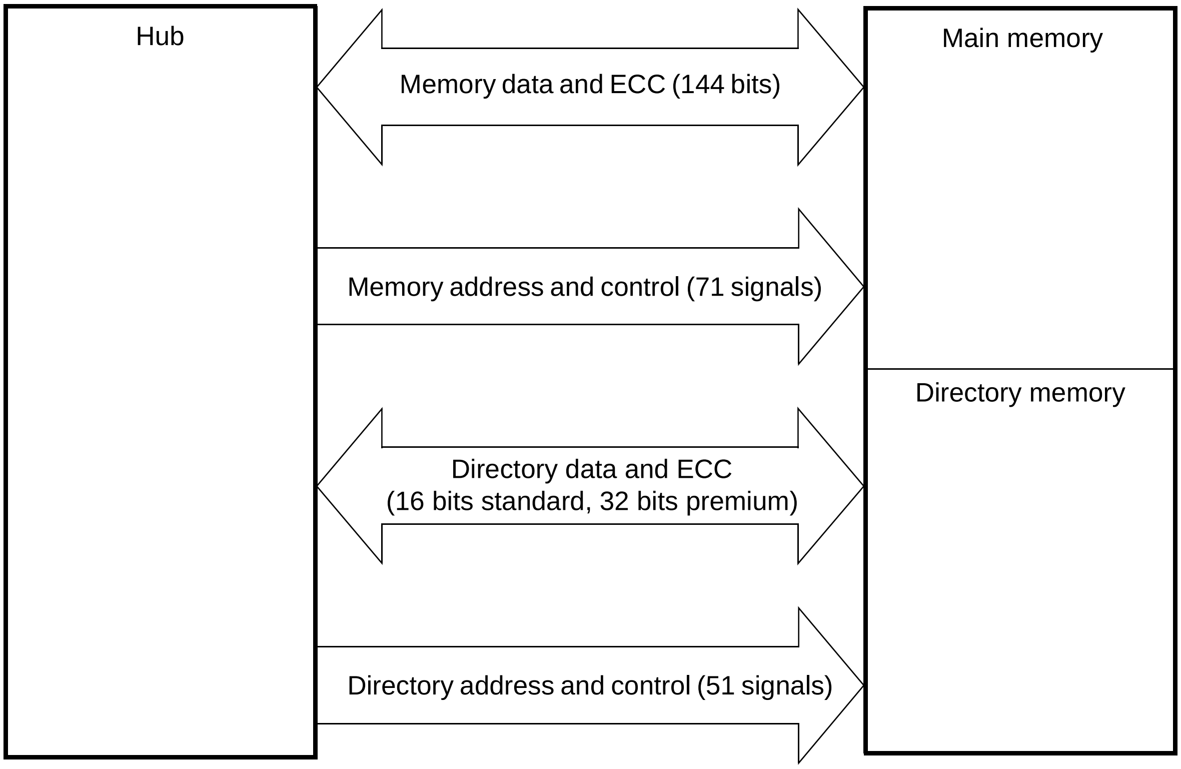

Hub
Main memory
Memory data and ECC (144 bits)
Memory address and control (71 signals)
Directory memory
Directory data and ECC
(16 bits standard, 32 bits premium)
Directory address and control (51 signals)

Figure 2-4 DIMM Paths

Memory is organized so that each compute node has a single shared address space of 8 Gbytes (refer to Figure 2-5). A node index number identifies these single shared address spaces. For example, the compute node that is assigned the first 8 Gbytes of memory (0 through 8 Gbytes minus [–] 1) is referred to as node index 0.

To determine the node index number of each physical node, execute the IRIX *icrash -e mem* command or execute the *hinv -v* command from the PROM monitor's command monitor prompt.

Note: When executed from an IRIX prompt, the *hinv -v* command does not provide node index information.

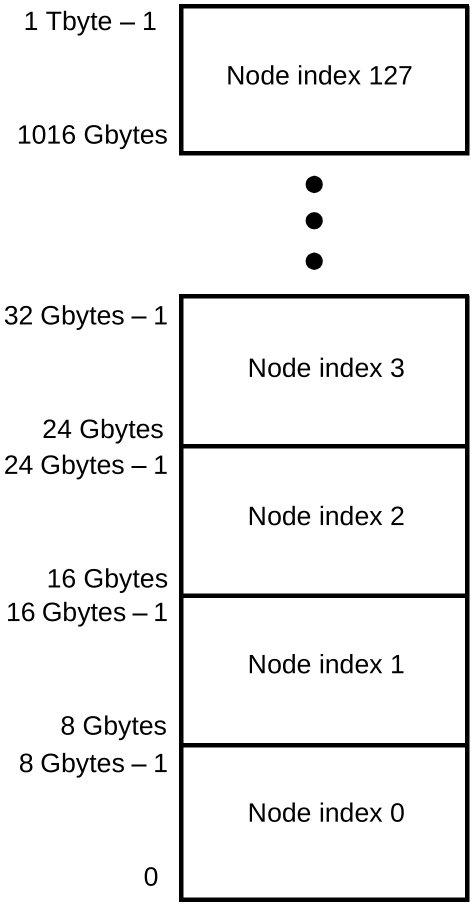


Figure 2-5 Address Space and Node Index Numbers

The physical memory address consists of two fields: a 7-bit node index and a 33-bit node offset (refer to Figure 2-6). The node index indicates the compute node that contains the local memory to be referenced. The node offset includes the bank number and the memory address within the bank.

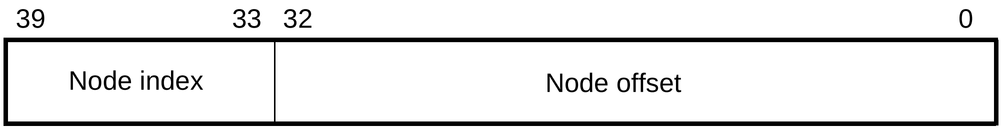


Figure 2-6 Physical Memory Address

2.1.4 Hub

The hub enables communication among the processors, memory, routers, and I/O devices (refer to Figure 2-7). It controls all activity within the compute node; for example, error correction and cache coherency. The hub also supports page migration.

The hub consists of:

- A central crossbar (XB)
- Two processor interfaces (PI_0 and PI_1)
- A memory/directory interface (MD)
- A network interface (NI)
- An I/O interface (II)
- A local block (LB)

Note: Error messages and other logged information may use the acronyms to identify the components of the hub.

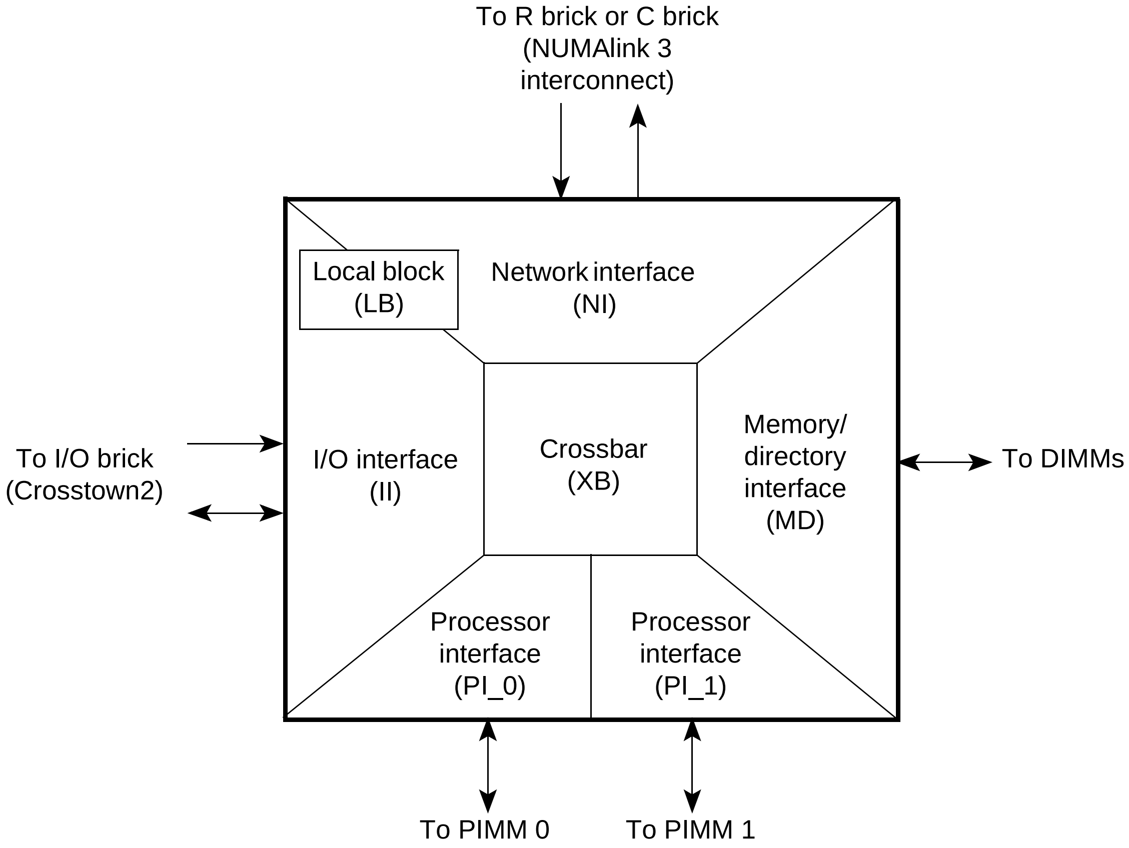


Figure 2-7 Hub Block Diagram

2.1.4.1 Crossbar Unit

The crossbar unit provides connectivity between the hub interfaces. It is an 8-input, 6-output crossbar that communicates directly with each interface (refer to Figure 2-8). The crossbar unit also performs arbitration and some error handling.

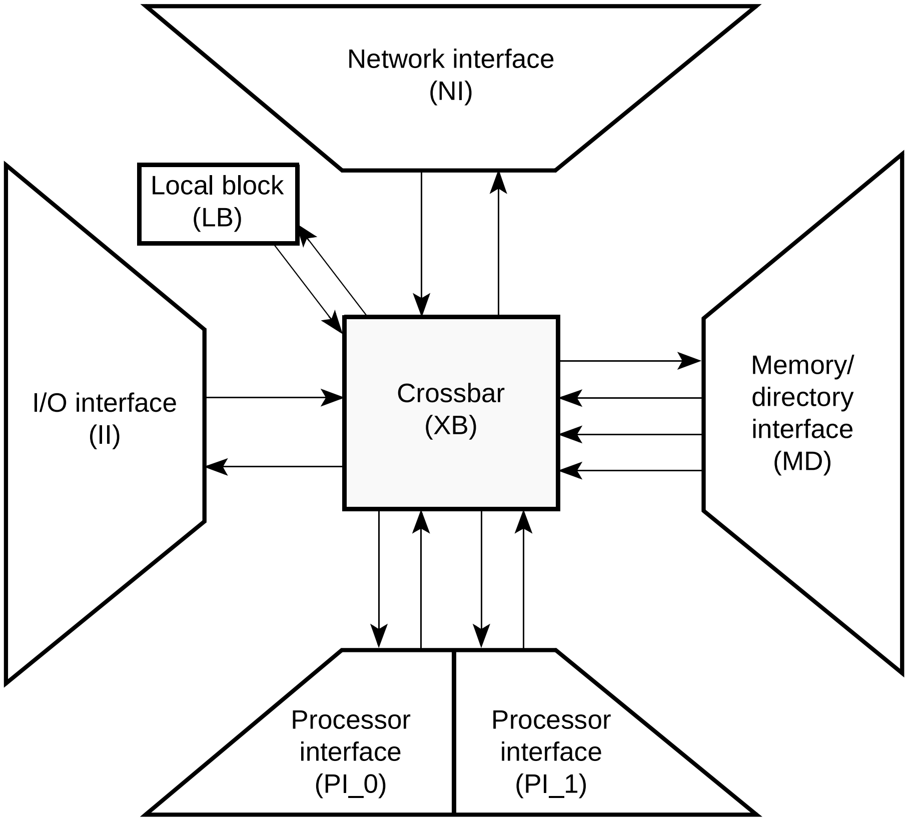


Figure 2-8 Crossbar Unit Block Diagram

2.1.4.2 Processor Interface

Each processor interface communicates directly with one or two processors that are located on a PIMM; a processor interface controls the flow of a bus that transfers address, data, and commands between the processor interface and the PIMM (refer to Figure 2-9). The processor interface also performs arbitration and some error handling.

Note: The processors do not communicate directly with each other; instead, they communicate via the processor interface and crossbar.

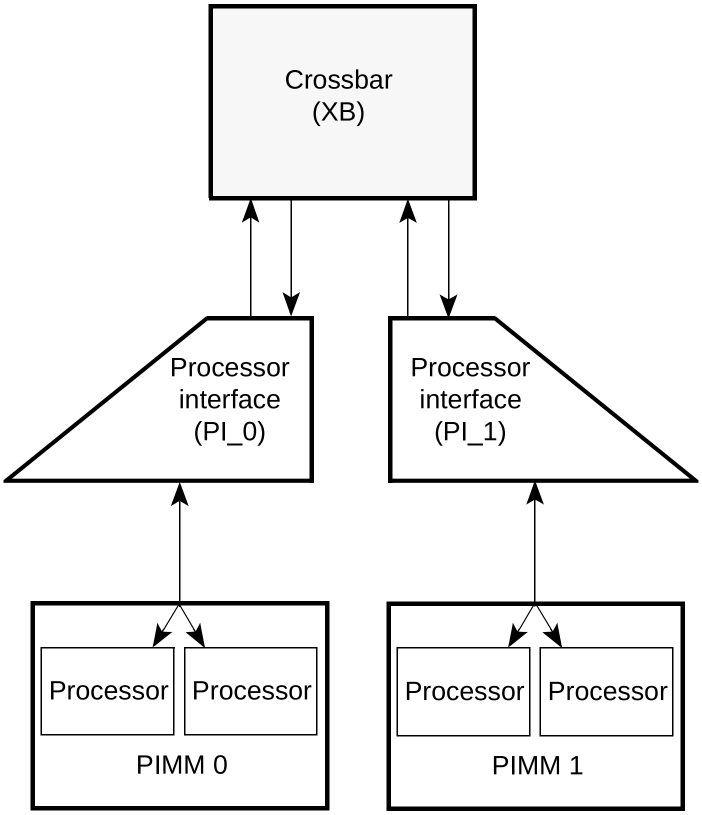


Figure 2-9 Processor Interface Block Diagram

2.1.4.3 Memory/Directory Interface

The memory / directory interface controls all memory access (refer to Figure 2-10). It consists of three blocks:

- Issue block - Receives new message requests, arbitrates the requests, and supplies the address and control signals to the DIMMs
- Memory block - Controls the flow of data during read and write operations:
  - Read operation- receives data from the memory chips, detects and corrects errors, and sends the data to the crossbar unit
  - Write operation - receives data from the crossbar unit, generates ECC, and writes the data to the DIMMs
- Directory block - Maintains cache coherence: reads the directory data, creates headers for outgoing messages, computes new directory data, generates ECC, and writes the new directory data to the DIMMs

The following list describes some of the other functions of the memory / directory interface:

- Supports page migration
- Refreshes the DDR-SDRAM on each DIMM approximately every 8 microseconds
- Supports a built-in self-test (BIST) that tests all of memory (data, ECC, and directory)

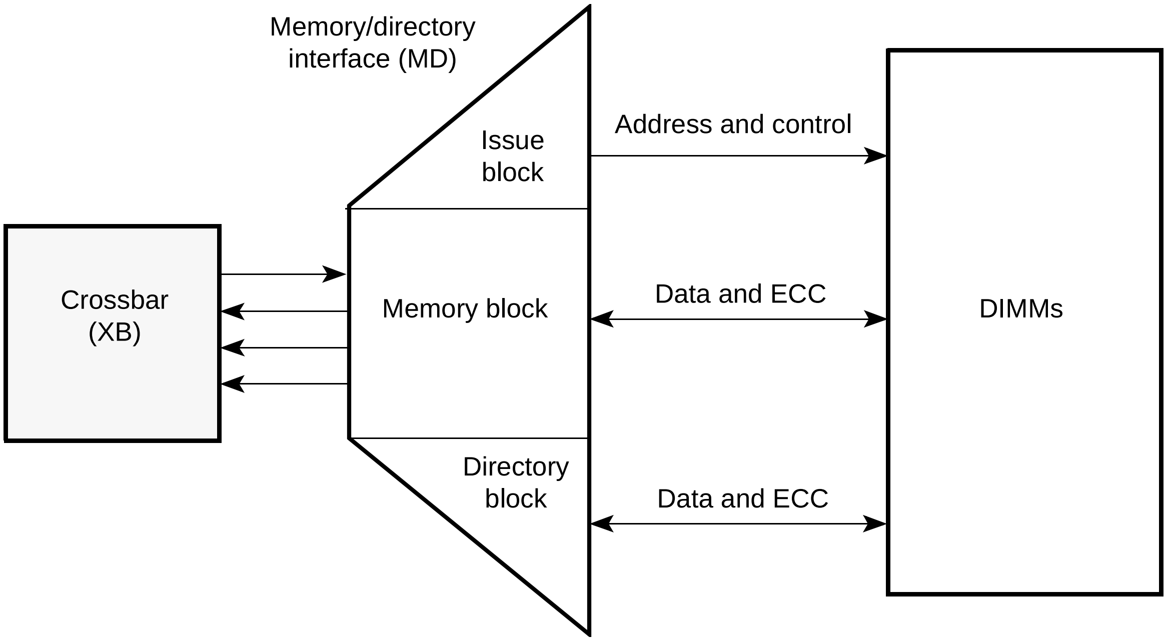


Figure 2-10 Memory / Directory Block Diagram

2.1.4.4 Network Interface

The network interface is the interface between the crossbar unit and the NUMAlink 3 interconnect (refer to Figure 2-11). It can connect to an R brick or to the network interface of another C brick. The network interface consists of:

- A source synchronous driver (SSD) - Receives 80 bits of data and control from the LLP logic at a frequency of 200 MHz. The SSD divides the 80 bits into 20-bit transfers, which the SSD sends to an R brick (or C brick) at a frequency of 800 MHz.
- A source synchronous receiver (SSR) - Receives 20 bits of data and control from the NUMAlink 3 interconnect at a frequency of 800 MHz. The SSR assembles four 20-bit transfers into an 80-bit transfer, which the SSR sends to the LLP logic at a frequency of 200 MHz.
- Link level protocol (LLP) logic - Protects messages that are sent over the NUMAlink 3 interconnect:
    - For outgoing messages, the LLP logic uses a 16-bit cyclic redundancy check (CRC) code referred to as CCITT to generate 16 check bits that protect 128 data bits. The LLP logic places these bits in the micropacket and sends the micropacket to the SSD.
    - For incoming messages, the LLP logic detects errors; for example, it detects single-, double-, and odd-bit signalling errors, burst errors, and lost or duplicate messages. The LLP logic attempts to recover from an error by resending the message. It continues to resend the message until it is error free or until it reaches a retry limit. For more information about LLP, refer to Chapter 4 (Data Integrity) of this document.
- Message receive logic - Translates incoming messages from the NUMAlink 3 interconnect to crossbar protocol and sends the messages to the crossbar unit.
- Message send logic - Translates outgoing messages from crossbar protocol to NUMAlink 3 protocol, retrieves routing information from a routing table, incorporates the routing information into the message headers, and sends the messages to the LLP logic.

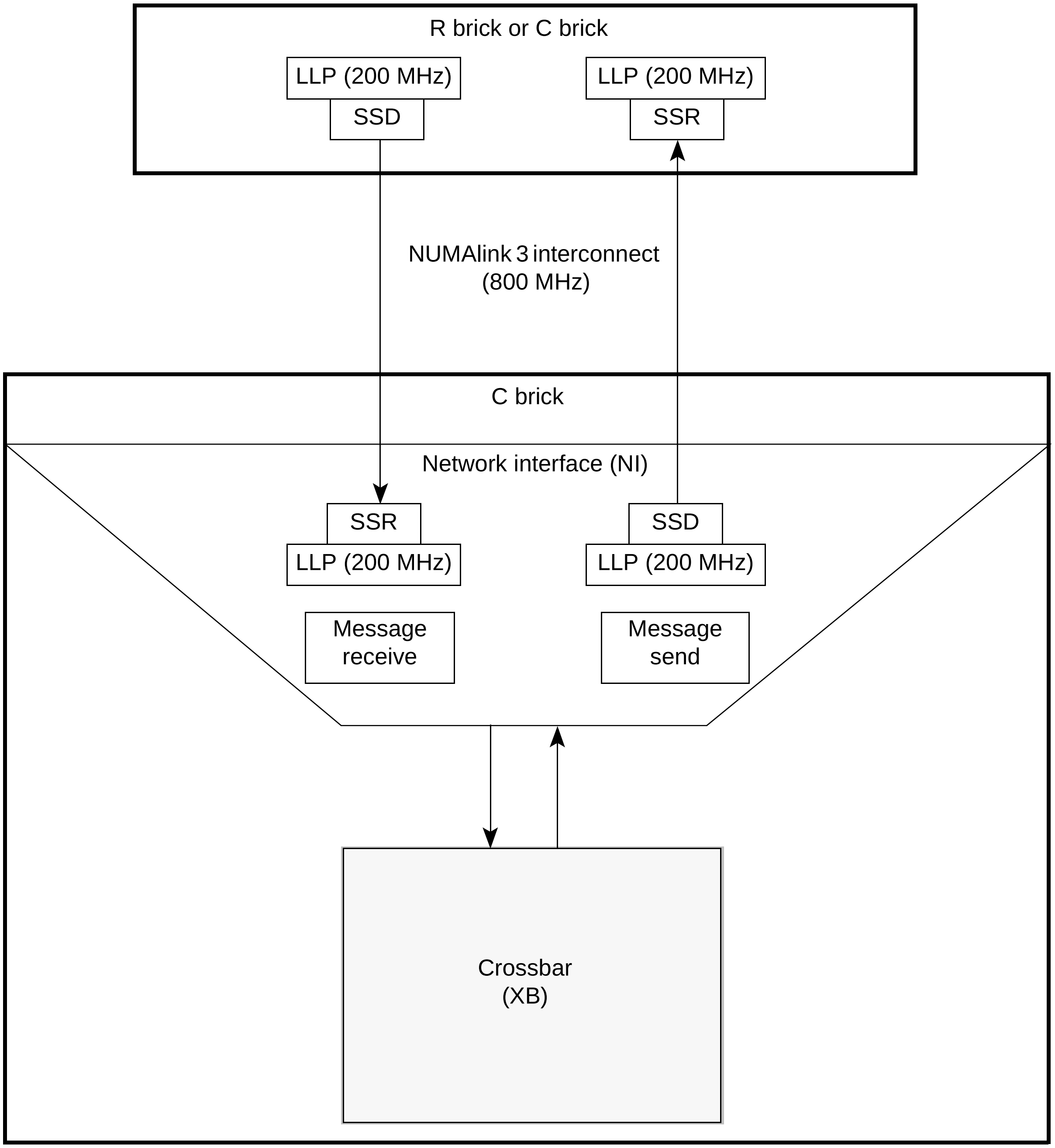


Figure 2-11 Network Interface Block Diagram

2.1.4.5 I/O Interface

The I/O interface allows the I/O devices to read and write memory (direct memory access [DMA] operations) and allows the processors within the system to control the I/O devices (PIO operations).

For DMA operations, the I/O device initiates a request by sending a Crosstalk2 request message to the I/O interface. The I/O interface translates the message into the crossbar format and sends the message to the crossbar unit. The crossbar unit sends the message to the memory/directory interface. For PIO operations, a processor initiates a request by sending a request message to the processor interface. The processor interface sends the message to the crossbar unit; the crossbar unit sends the message to the I/O interface. The I/O interface determines whether the request is a read or a write operation.

If the request is a read operation, the I/O interface saves the node number and processor number, translates the message into Crosstalk2 protocol, and sends the request to the I/O device. When the I/O device returns the read data, the I/O interface retrieves the node and processor numbers, creates a response message, and sends the response message to the processor that originated the request. If the request is a write operation, the I/O interface translates the message into Crosstalk2 protocol and sends the request to the I/O device. When the I/O interface receives a write response from the I/O device, the I/O interface checks the response for errors and other status information and discards the response message.

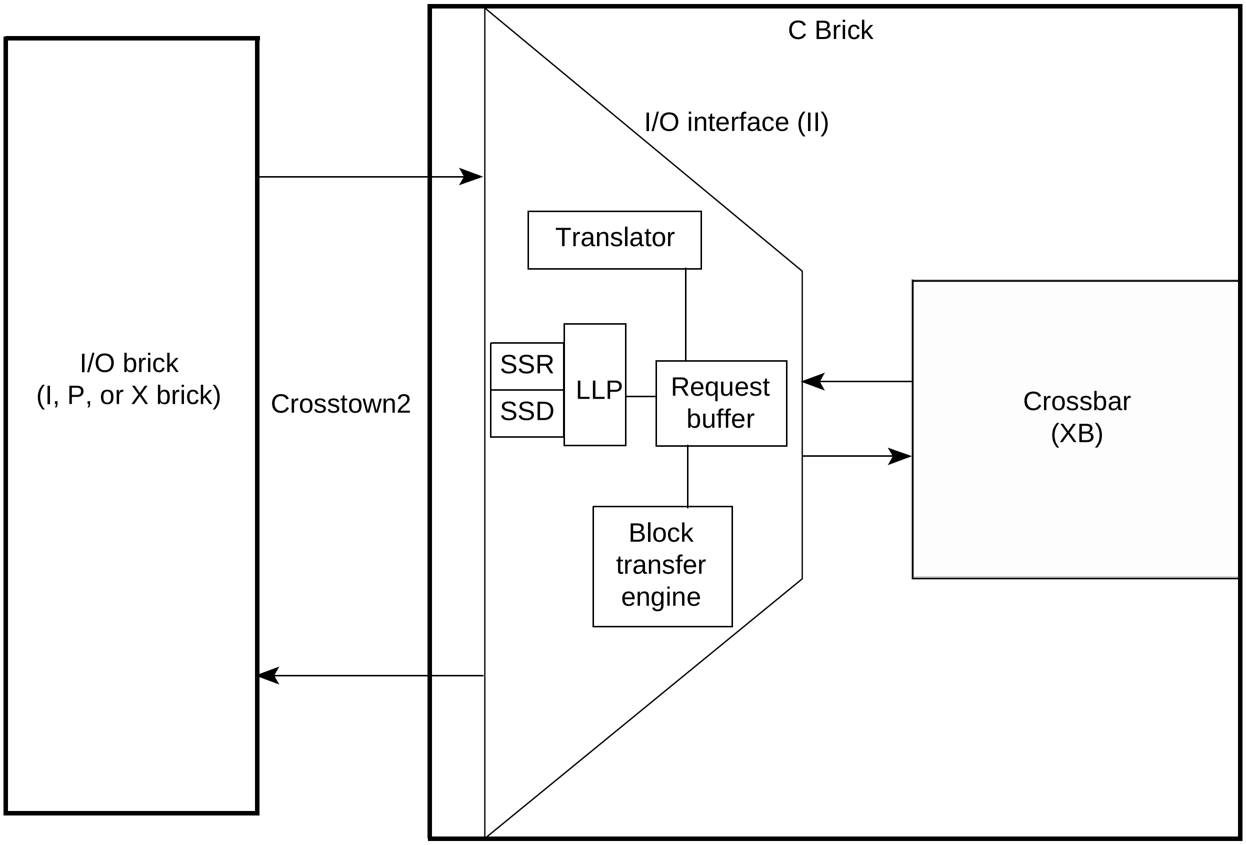


Figure 2-12 I/O Interface Block Diagram

2.1.4.6 Local Block

The local block services processor I/O (PIO) requests that are local to the hub; for example, it services PIO write and read requests for registers that are located in the network interface, the crossbar unit, and the memory/directory interface (refer to Figure 2-13). The local block also contains some registers that software can access via PIO reads and writes (normal and vector).

The local block also performs the following functions:

- Generates a real-time clock (RTC) signal
- Sends replies for processor I/O read and write requests
- Initiates outgoing vector PIO read and write operations

  Note: Vector operations are used during system initialization.

- Services local invalidate requests

  The local block receives an invalidate request when a local processor or I/O interface wants exclusive ownership of a cache block of data that is shared by other processors in the system. After the local block receives the request, it creates an invalidate message that it sends to the processors that hold copies of the cache block. The processors then invalidate their copy of the data.

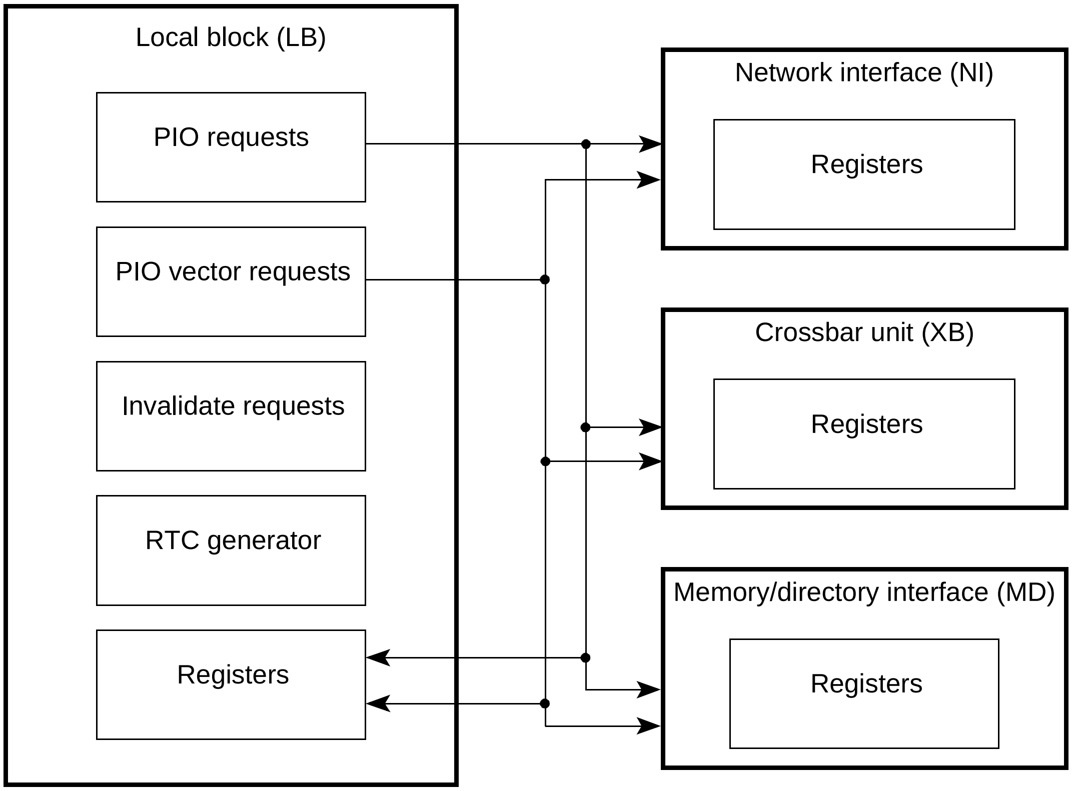
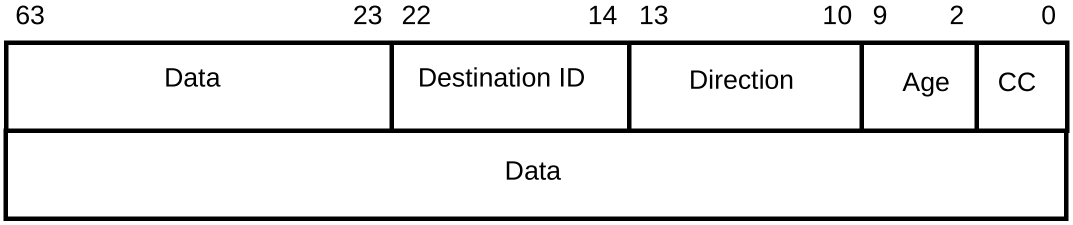


Figure 2-13 Local Block

2.2 NUMAlink 3 Interconnect

The NUMAlink 3 interconnect routes messages between the compute nodes.

Note: A message is one or more 128-bit micropackets. Each micropacket is accompanied with an 8-bit sideband field. The sideband field contains information that the NUMAlink 3 interconnect uses to transfer the message to its destination. For example, to indicate the last micropacket of a message, a tail bit of the sideband information is set to 1. Each message also consists of a header micropacket. Bits 0 through 22 of the header micropacket specify destination and routing information. The remaining bits are used for data (refer to Figure 2-14).



o Destination ID (identifier) specifies one of the 512 possible network destinations.
o Direction field specifies the port that the message will use to exit the next router.
o Age field indicates the age of the message.
o Congestion control (CC) specifies the virtual channel.

Figure 2-14 Header Micropacket

There is one static NUMAlink 3 path between any two compute nodes in the system. These NUMAlink 3 paths are defined by distributed routing tables. When a NUMAlink 3 path fails, software can bypass the failing component by changing the contents of the routing tables.

The NUMAlink 3 interconnect also uses link level protocol (LLP) to provide error-free transmission of messages. The LLP uses the CCITT CRC code to detect errors. It corrects errors by retransmitting the message. For more information about LLP, refer to Chapter 4 of this document.

Note: CRC is the acronym for cyclic redundancy check. CCITT is the French acronym for Comite Consultatif Internationale de Telegraphie et Telephonie (International Consulting Committee on Telegraphs and Telephones).

The NUMAlink 3 interconnect consists of cables, routers, MetaRouters, and repeat routers. A router transfers messages between compute nodes. A MetaRouter transfers messages between routers (for systems that have more than 128 processors). A repeat router transfers messages between MetaRouters and routers (for systems that have more than 256 processors) and between MetaRouters (for systems that have more than 384 processors).

The routers, MetaRouters, and repeat routers are linked together by cables in various configurations or topologies. When the system does not have an R brick, the NUMAlink 3 interconnect consists of a cable that links two compute nodes together. For more information about configurations or topologies, refer to the *System Configurations* document, publication number 108-0266-001.

The routers, MetaRouters, and repeat routers use identical hardware referred to as the R brick. The location of the R brick within the system determines whether it is a router, MetaRouter, or a repeat router.

The R brick consists of eight differential NUMAlink 3 ports and a router application specific integrated circuit (ASIC) that arbitrates for port access and provides connection between the ports via a crossbar unit (refer to Figure 2-15).

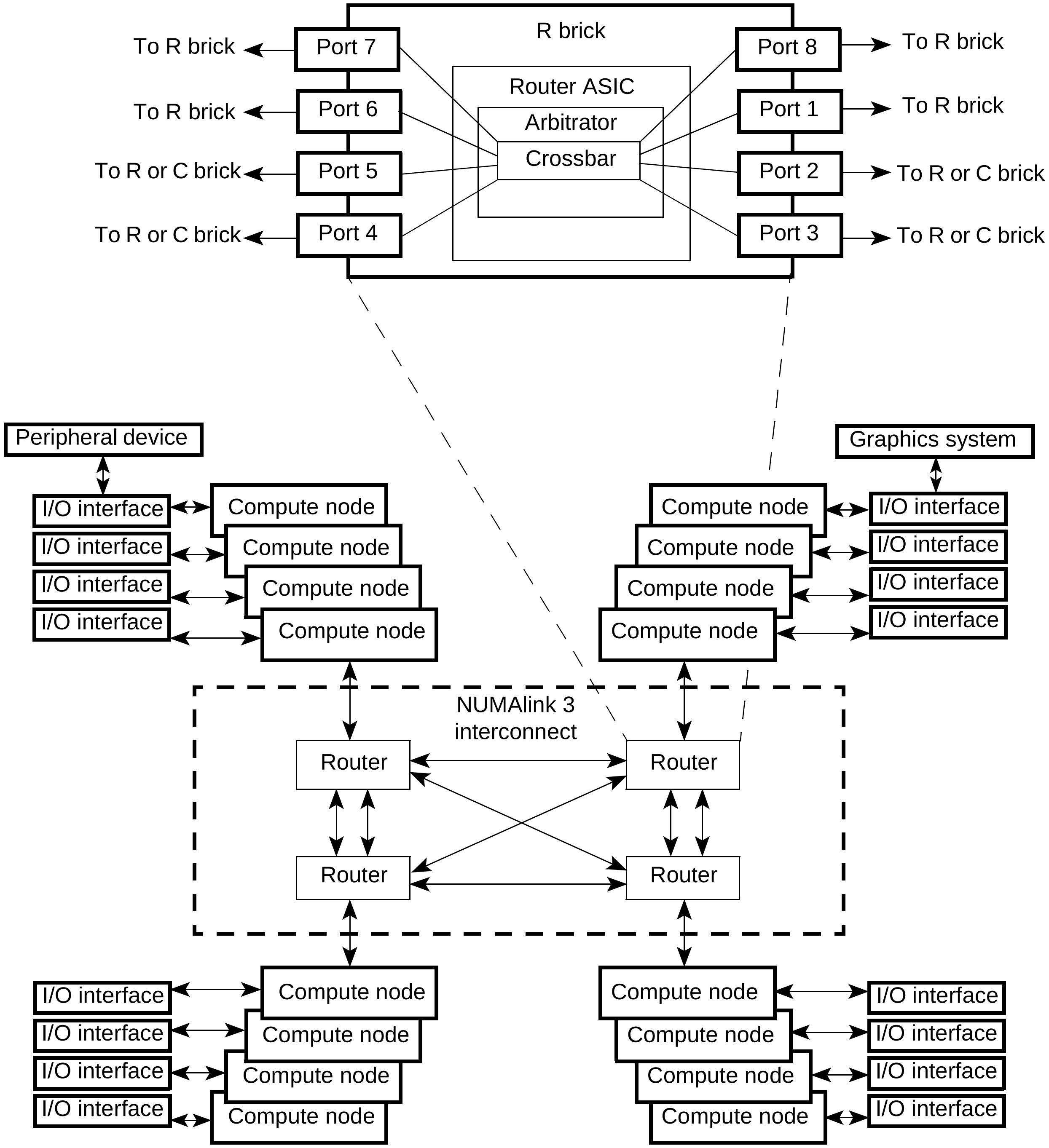


Figure 2-15 SGI 3000 Series Servers Block Diagram with Detailed Router (R Brick)

Note: The SGI 3800 servers require all eight ports of the R brick. The SGI Origin 3400 servers require only six ports; therefore, the R bricks of the SGI 3400 servers have two ports disabled. Disabling two ports prevents illegal system upgrades or mergers that violate Federal and International export laws, invalidate SGI contractual agreements, and / or decrease SGI revenue.

Each port has four virtual channels (virtual channels 0 through 3) that prevent deadlock conditions within the interconnect. A deadlock condition occurs when one or more messages are locked in a state from which they cannot proceed.

For example, requests require responses. To prevent deadlock conditions between the requests and responses, requests use virtual channels 0 and 1 and responses use virtual channels 2 and 3 (refer to Figure 2-16). The odd-numbered virtual channels are used to prevent deadlock conditions among the requests or responses.

Note: A request or response packet is assigned to a virtual channel and must complete the transfer before the virtual channel can be used by another packet.

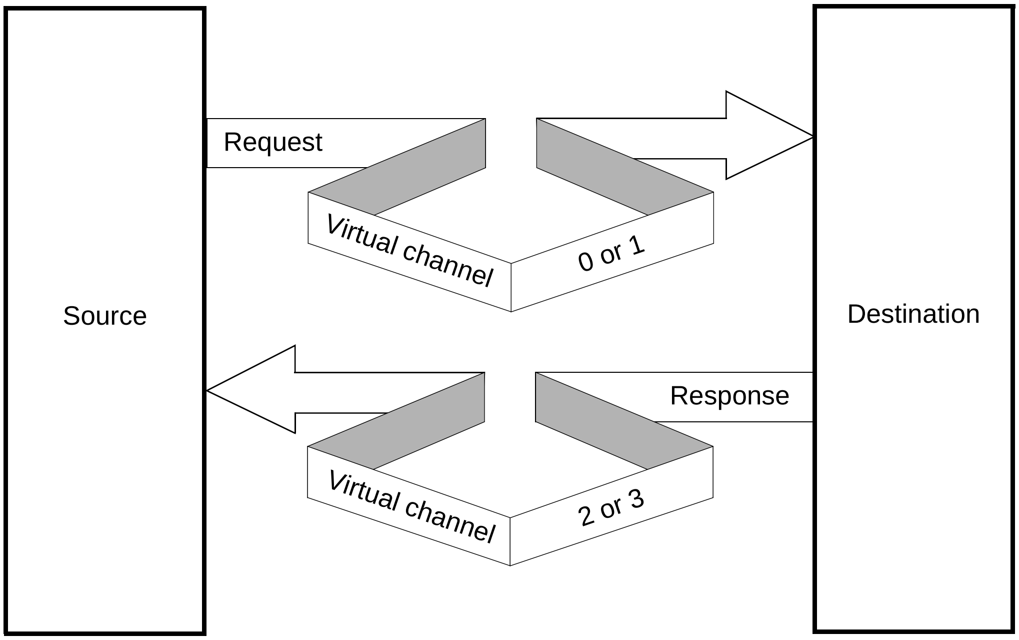


Figure 2-16 Request and Response Virtual Channels

When a message enters a port via a virtual channel, the router ASIC uses the direction field of the message header to determine which port the message should use to exit the R brick. Once the router ASIC determines the exit port, the router ASIC arbitrates for port access.

The age field of the message header contains a value of 0 through 240 to indicate the priority of the message. For example, when the age field contains a 0, the message is new; therefore, it has low priority. When the age field contains 240, the message has high priority. Once the message enters the R brick, it ages (contents of the age field increment) at a programmable rate until it exits the R brick. The router ASIC places the new age value in the age field of the message header before it sends the message to the next R brick.

When the message has priority to use the port, the message passes through a crossbar (refer to Figure 2-17) and exits the R brick via the designated port.

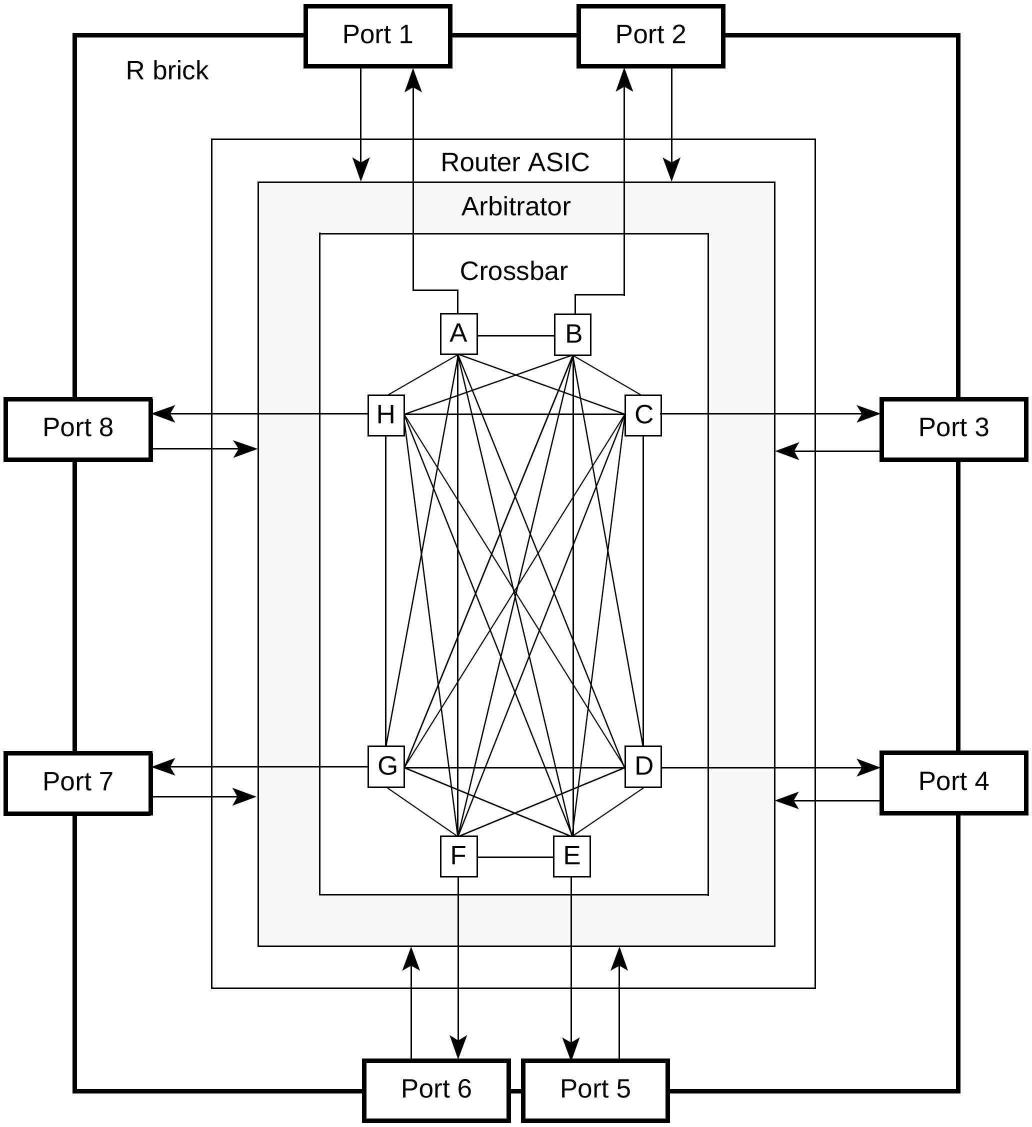


Figure 2-17 Router Block Diagram

When a message enters a port via a virtual channel, the router ASIC also uses the destination identifier of the message header to retrieve routing information from the R-brick routing table (refer to Figure 2-18). For example, each R brick has a register that holds the R-brick identification number and a routing table that consists of two subtables: a local table and a remote table. The four least significant bits of the destination identifier are the address for the local table; the upper five bits of the destination identifier are the address for the remote table. When the upper five bits of the destination identifier match the contents of the R-brick identification register, the router ASIC retrieves information from the local table to steer the message to the correct destination.

When the upper five bits of the destination identifier do not match the contents of the R-brick identification register, the router ASIC uses the remote table to retrieve routing information. This routing information indicates the port that the message will use to exit the next R brick in the NUMAlink 3 path. The router ASIC places this exit port information in the direction field of the message header.

Note: The SGI 3000 series servers also support vector routing, which is used for network configuration and administration. For this type of routing, routing tables are not used; instead, vector-routed messages contain the routing information that defines the complete path between the source and destination.

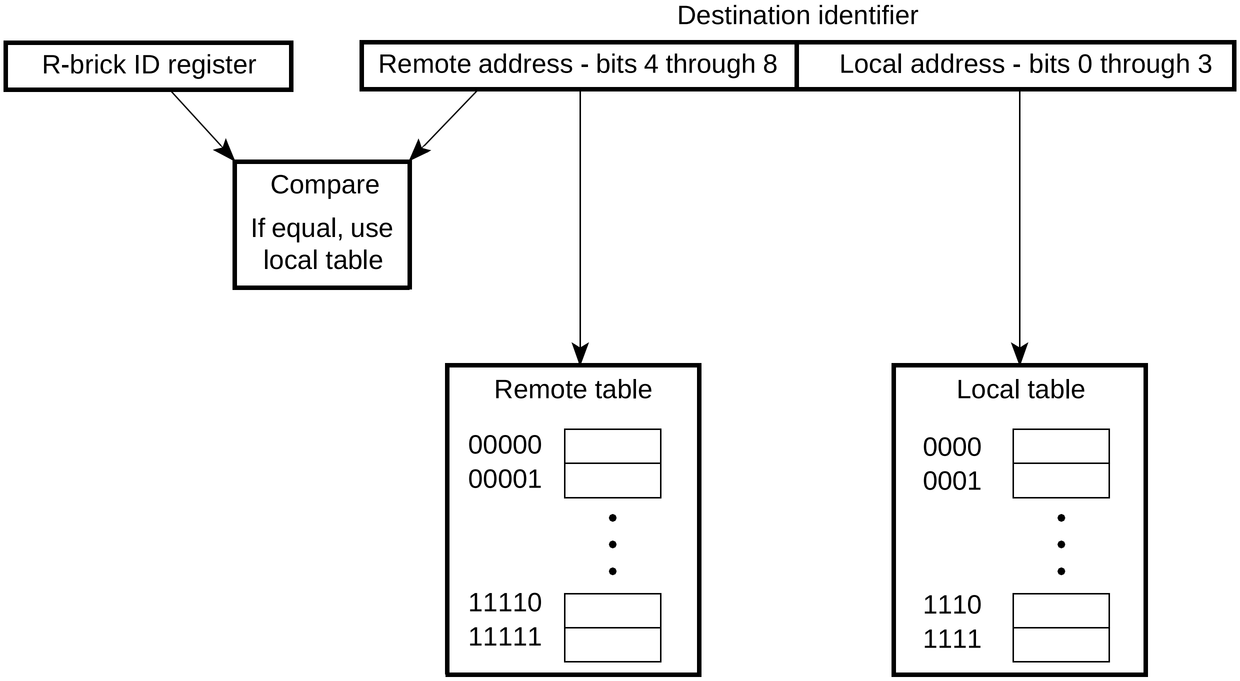


Figure 2-18 Local and Remote Routing Tables

2.2.1 Routers

When the R brick functions as a router, the R brick uses the eight ports to transfer messages between compute nodes: four ports connect to compute nodes (C bricks) and four ports connect to other routers (R bricks); refer to Figure 2-19.

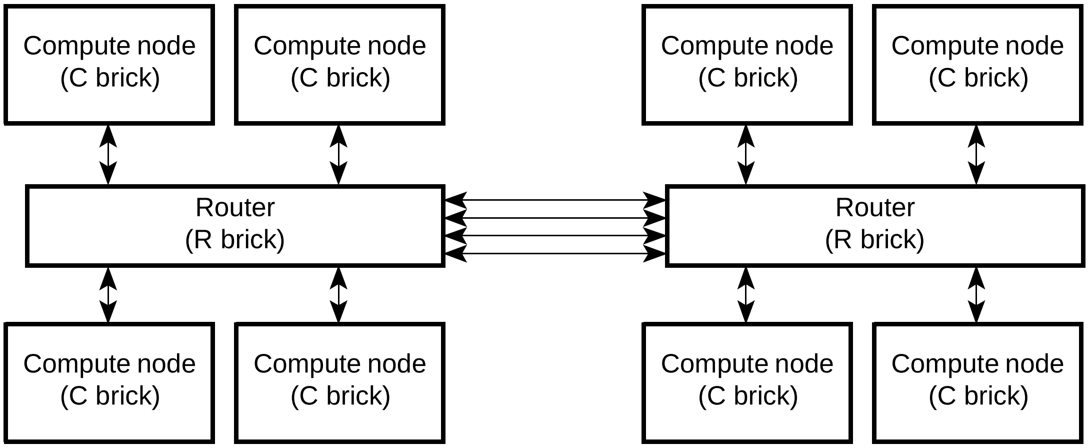


Figure 2-19 NUMAlink 3 Ports

2.2.2 MetaRouters

Systems that have more than 128 processors have additional R bricks that are referred to as MetaRouters. MetaRouters connect a group of 128 processors to another group of 128 processors or less. For example, in Figure 2-20 the Metarouters connect the 128 processors of Racks 001 through 004 to the 128 processors of Racks 011 through 014.

Note: Each router connects to four compute nodes; in this example, each compute node contains four processors.

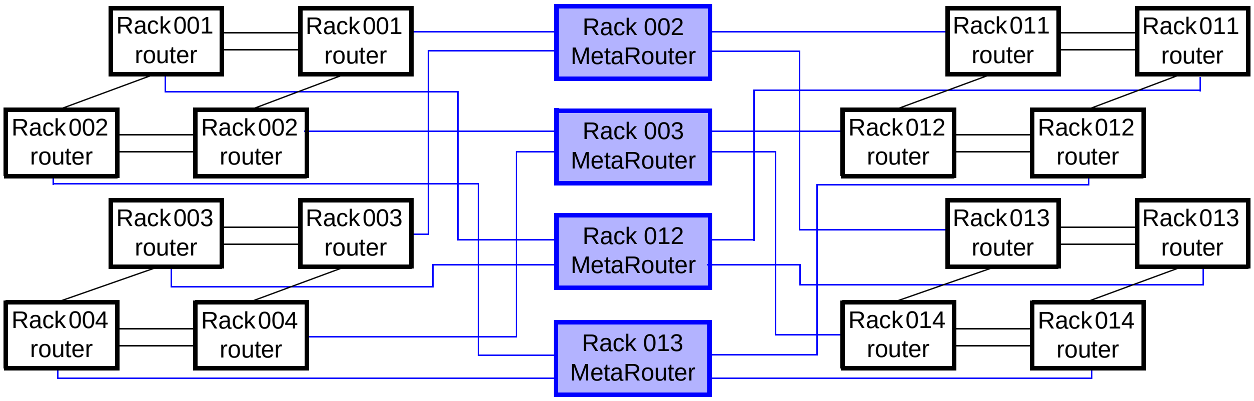


Figure 2-20 MetaRouter (256-processor System)

2.2.3 Repeat Routers

For systems that have more than 256 processors, R bricks also provide a connection between MetaRouters and between MetaRouters and routers. For this function, the R brick is referred to as a repeat router. For example, in Figure 2-21 the processors of Racks 001 through 004 and Racks 011 through 014 create a group of 256 processors. In order to connect this group of processors to another group of 256 processors, MetaRouters and repeat routers are used.

Note: When an R brick is a MetaRouter or a repeat router, all eight ports can connect to other R bricks.

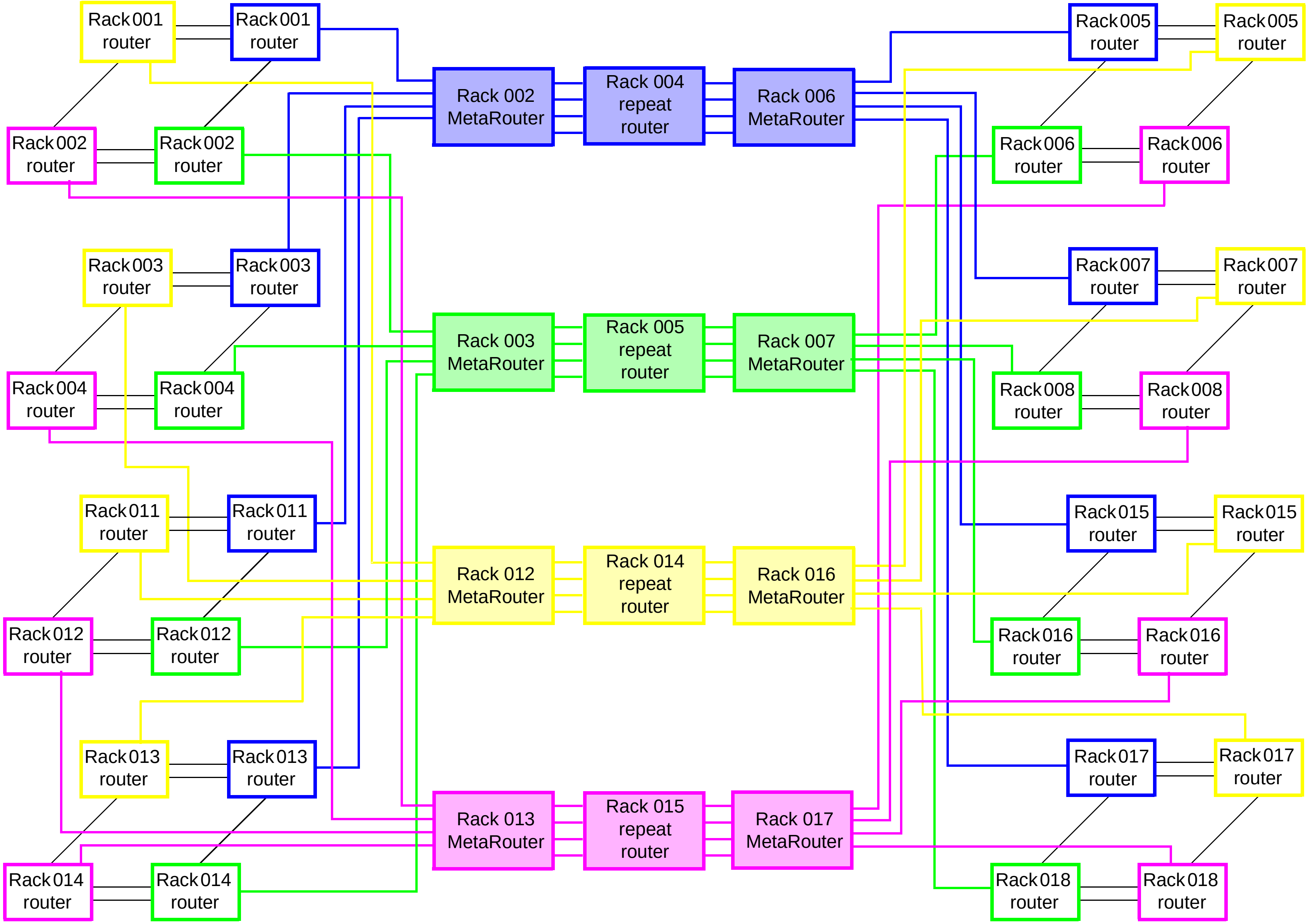


Figure 2-21 MetaRouters and Repeat Routers (512-processor system)

2.3 I/O Devices

The SGI 3000 series servers support a peripheral component interface (PCI) based I/O system and the legacy Crosstalk I/O (XIO) system of the Origin 2000 and Octane systems. The PCI-based I/O system, which is an industry standard for connecting peripherals to a processor, is the primary I/O system for the SGI 3000 series servers.

The key component of the PCI-based and XIO-based I/O system is the Xbridge ASIC. The Xbridge ASIC receives packets from a Crosstown2 port, converts the packets from Crosstalk protocol to PCI protocol, and passes this information to a PCI device. The Xbridge ASIC can also pass packets from a Crosstown2 port to an XIO device (no conversion necessary).

The Xbridge ASIC consists of eight ports: two Crosstown2 ports, four Crosstalk ports, and two PCI ports (refer to Figure 2-22). Ports A and B are the Crosstown2 ports, ports 8, 9, C, and D are Crosstalk ports, and ports E and F are the PCI ports.

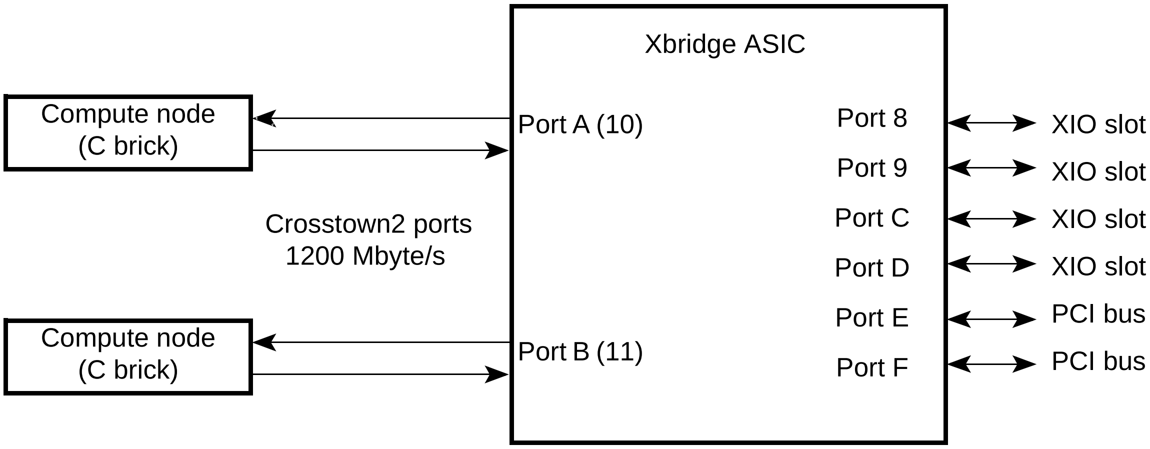


Figure 2-22 Xbridge Block Diagram

The Xbridge ASIC also ensures that the packets are free of errors by using link level protocol. When the Xbridge ASIC detects an error, it retransmits the packet, sets a flag in the Link *X* Status register, and if an interrupt for this error condition is enabled, interrupts the host processor. The Xbridge ASIC continues to retransmit a failing packet until it is error free or until it exceeds the maximum retry limit.

Note: Each port has a Link Status register; *X* equals port 8 through F. For example, port E has a Link E Status register.

For more information about link level protocol, refer to Chapter 4 (Data Integrity) of this document.

2.3.1 PCI-based I/O

The following I/O bricks support PCI-based I/O:

- The I brick provides the base I/O functions for the SGI 3000 series servers, which includes 1 or 2 system disks for boot functions. It also houses up to 5 PCI cards.
- The P brick houses up to 12 PCI cards.

2.3.1.1 I Brick

The I brick contains the IO7 devices that provide access to the network via a 10/100BaseT Ethernet port, and to external peripherals via one IEEE-1394 port and two universal serial bus (USB) ports.

Note: The IEEE-1394 port transports digital data for audio, digital compact discs, and video. A USB port connects peripherals such as monitors, keyboards, mouse, telephones, and modems to the system.

The I brick also has two PCI buses: one bus seats three 33-MHz PCI cards and the other bus seats two 33-MHz or 66-MHz PCI cards. Both buses can support 32- and 64-bit data and addressing at the same time.

An Xbridge ASIC controls the two PCI buses and is the interface between the peripheral devices and the compute nodes (refer to Figure 2-23).

The I-brick L1 controller controls how the I-brick power board applies power to the PCI cards. The power board applies power to the PCI cards starting with the lowest numbered slot. It continues to apply power to the PCI slots until all of the power has been consumed. The L1 controller uses two pins in each PCI slot to total the power consumption of the PCI cards.

Note: The power board supplies an average of 17.5 W (5.3 A, 3.3 V) of power to each PCI slot; however, a PCI card may consume up to 25 W of power.

The L1 controller prints a message to the console if there is not enough power for all of the PCI cards.

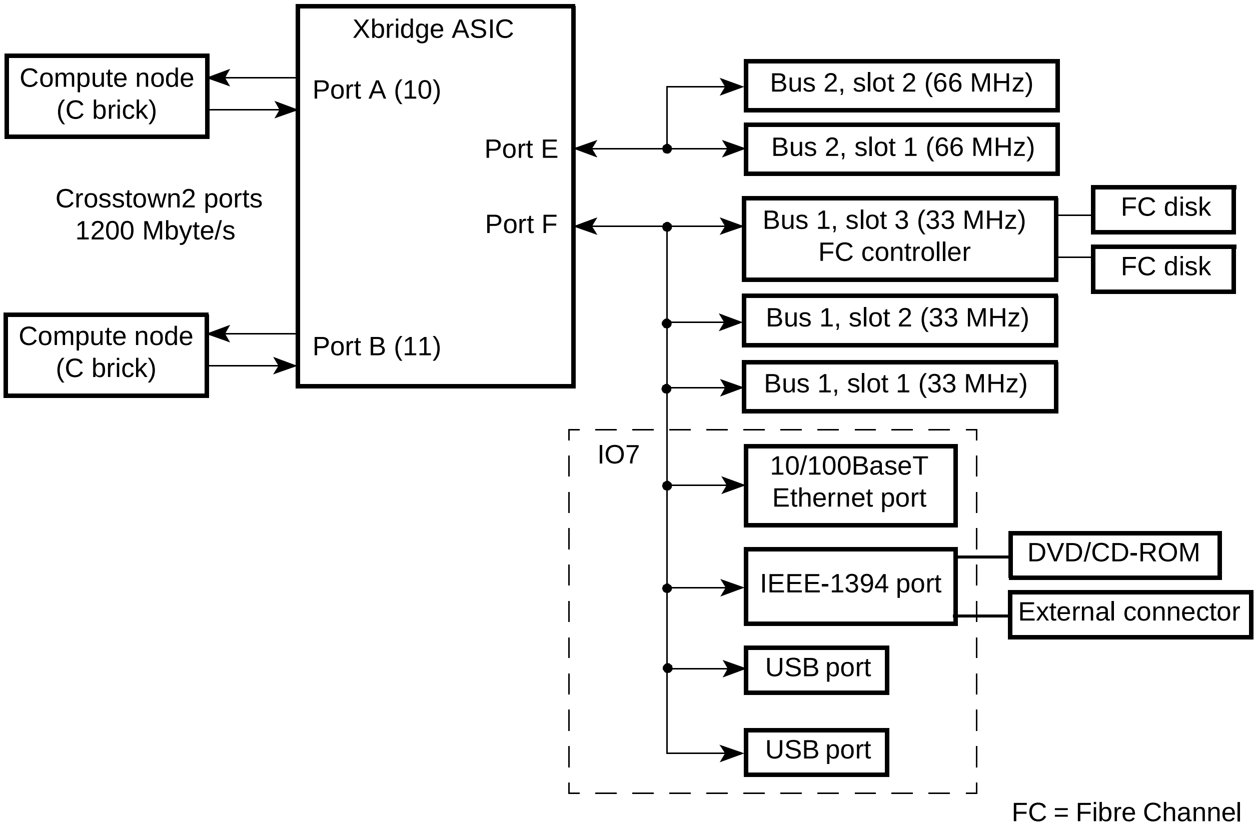


Figure 2-23 I-brick Block Diagram

2.3.1.2 P Brick

The P brick can seat up to twelve PCI cards; for example:

- Gigabit Ethernet adapter
- Fibre Channel (fiber-optic cable)
- Fibre Channel (copper cable)
- Ultra SCSI (small computer system interface) two port, high voltage differential
- Ultra SCSI two port, low voltage differential
- Ultra SCSI two port, single ended adapter
- ATM (asynchronous transfer mode) OC3 (optional connection 3) adapter
- ATM OC12 (optional connection 12)
- FDDI (fiber distributed device interface) dual-attach adapter
- RAD (rapid application development) audio
- Serial card (two port)

The P brick has six PCI buses; each bus supports either two 33-MHz PCI cards or two 66-MHz PCI cards. The buses can support 32- and 64-bit addressing at the same time.

The P brick has three Xbridge ASICs that control the PCI buses (refer to Figure 2-24): Xbridge ASIC U0 is the interface between the Crosstown2 ports and the PCI slots of Bus 3 and Bus 4. Xbridge ASIC U0 is also the Crosstown2 port interface for the Xbridge ASICs U1 and U2. Xbridge ASIC U1 controls Bus 1 and Bus 2 and Xbridge ASIC U2 controls Bus 5 and Bus 6.

The P-brick L1 controller controls how the P-brick power board applies power to the PCI cards. The power board applies power to the PCI cards starting with the lowest numbered slot. It continues to apply power to the PCI slots until all of the power has been consumed. The L1 controller uses two pins in each PCI slot to total the power consumption of the PCI cards.

Note: The power board supplies an average of 17.5 W (5.3 A, 3.3 V) of power to each PCI slot; however, a PCI card may consume up to 25 W of power.

The L1 controller prints a message to the console if there is not enough power for all of the PCI cards.

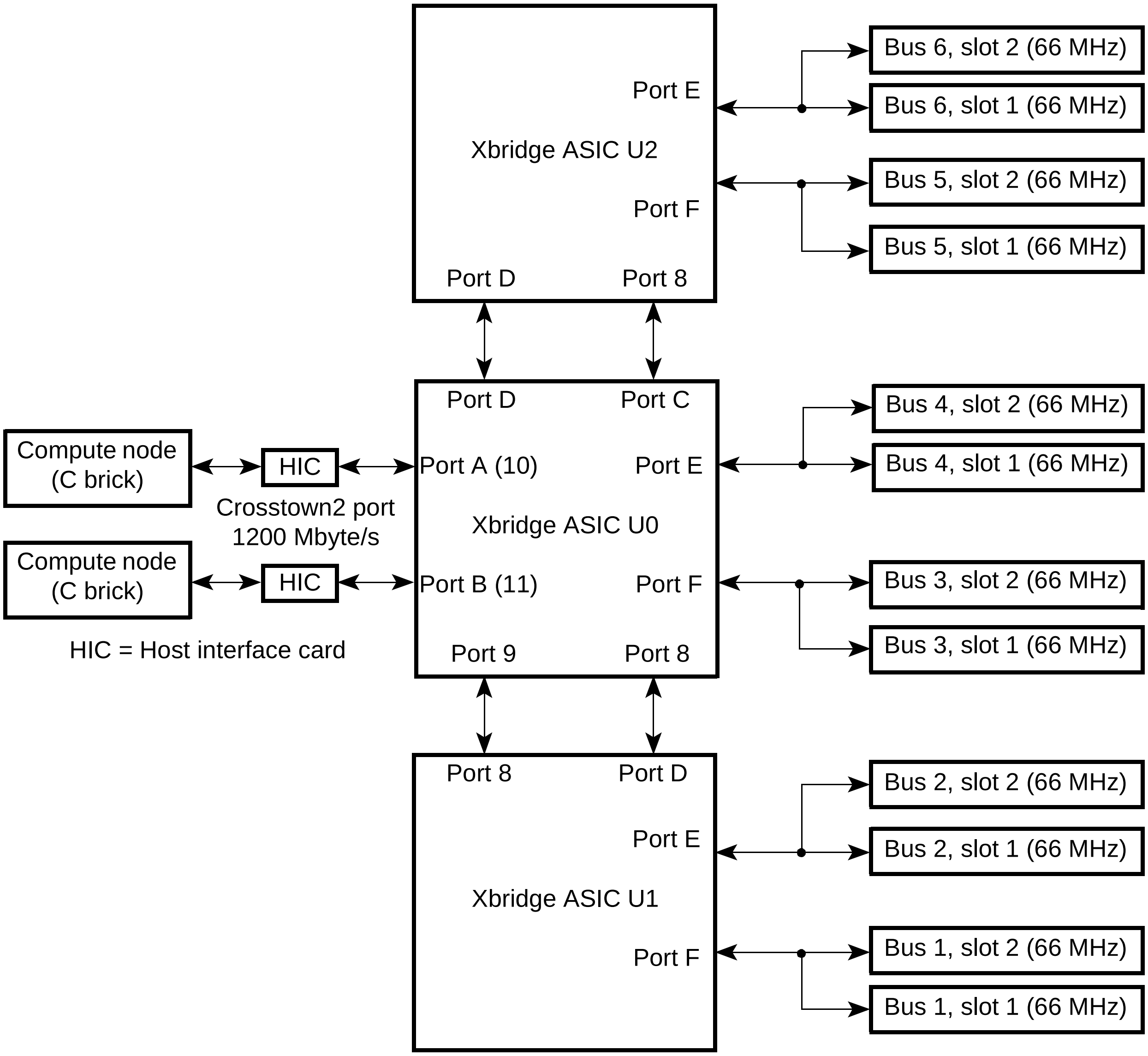


Figure 2-24 P-brick Block Diagram

2.3.2 Legacy Crosstalk I/O (XIO)

The X brick supports legacy Crosstalk I/O; it provides four half-height XIO slots that are compatible with the XIO slots of the Origin 2000 and Octane systems. Some of the XIO cards include:

- Single-port GSN (gigabyte system network) adapter (copper)
- Single-port serial HIPPI (high performance parallel interface)
- Digital video
- FIDDI (fiber distributed device interface)
- High-definition video
- XIO to VME adapters

An Xbridge ASIC controls the four XIO slots and is the interface between the Crosstown2 ports and the XIO slots (refer to Figure 2-25).

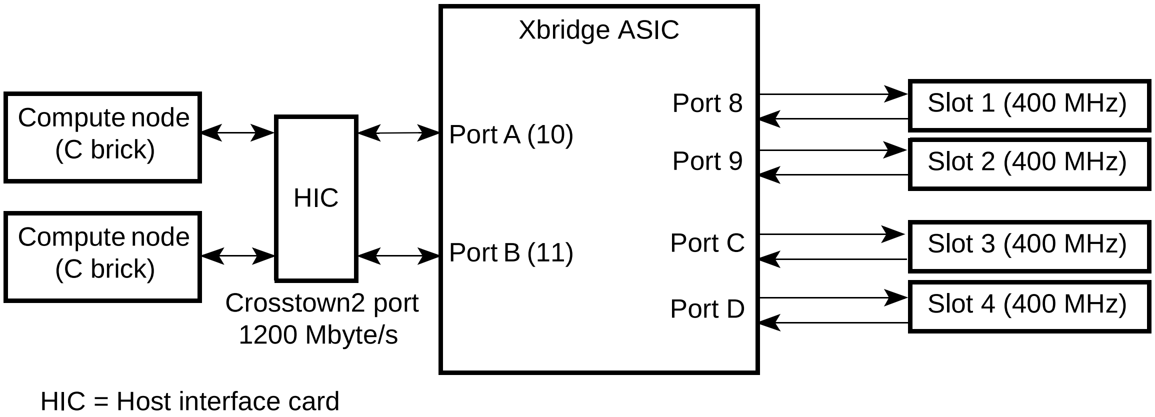


Figure 2-25 X-brick Block Diagram

2.4 Disk Devices

The disk devices are housed in D bricks. The D bricks support two Fibre Channel loops (refer to Figure 2-26), contain up to twelve Fibre Channel disk drives, and are purchased from an OEM vendor as tested units.

The D brick is available with or without RAID control logic (RAID module). A D brick with a RAID module contains 5 to 12 disk drives and is referred to as a base enclosure. A D brick without a RAID module contains 2 to 12 disk drives and is referred to as a JBOD (just a bunch of disks) enclosure in a JBOD configuration or an expansion enclosure in a RAID configuration.

In a RAID configuration, the expansion enclosure must be cabled to a base enclosure. The base enclosure can address up to 32 disk drives; therefore, two expansion enclosures can be cabled to one base enclosure (refer to Figure 2-27).

When a D brick is housed in an SGI 3000 series server rack, the D brick only supports a JBOD configuration (refer again to Figure 2-27).

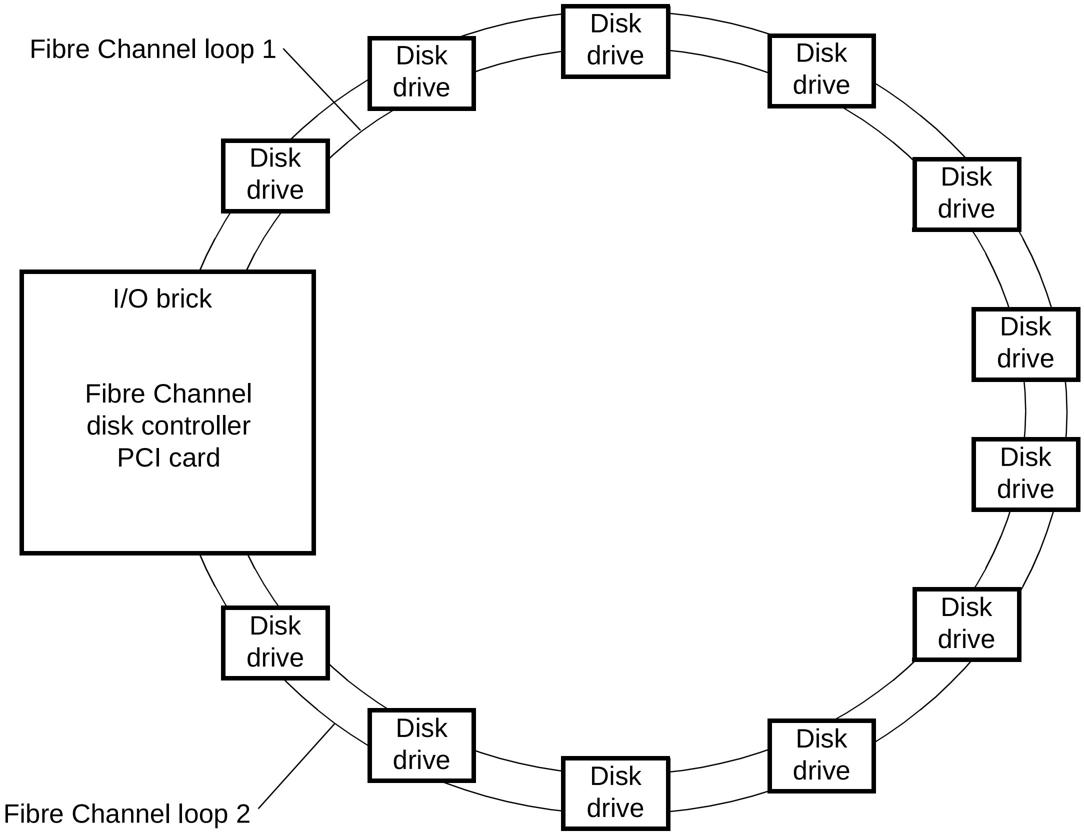


Figure 2-26 Fibre Channel Loops

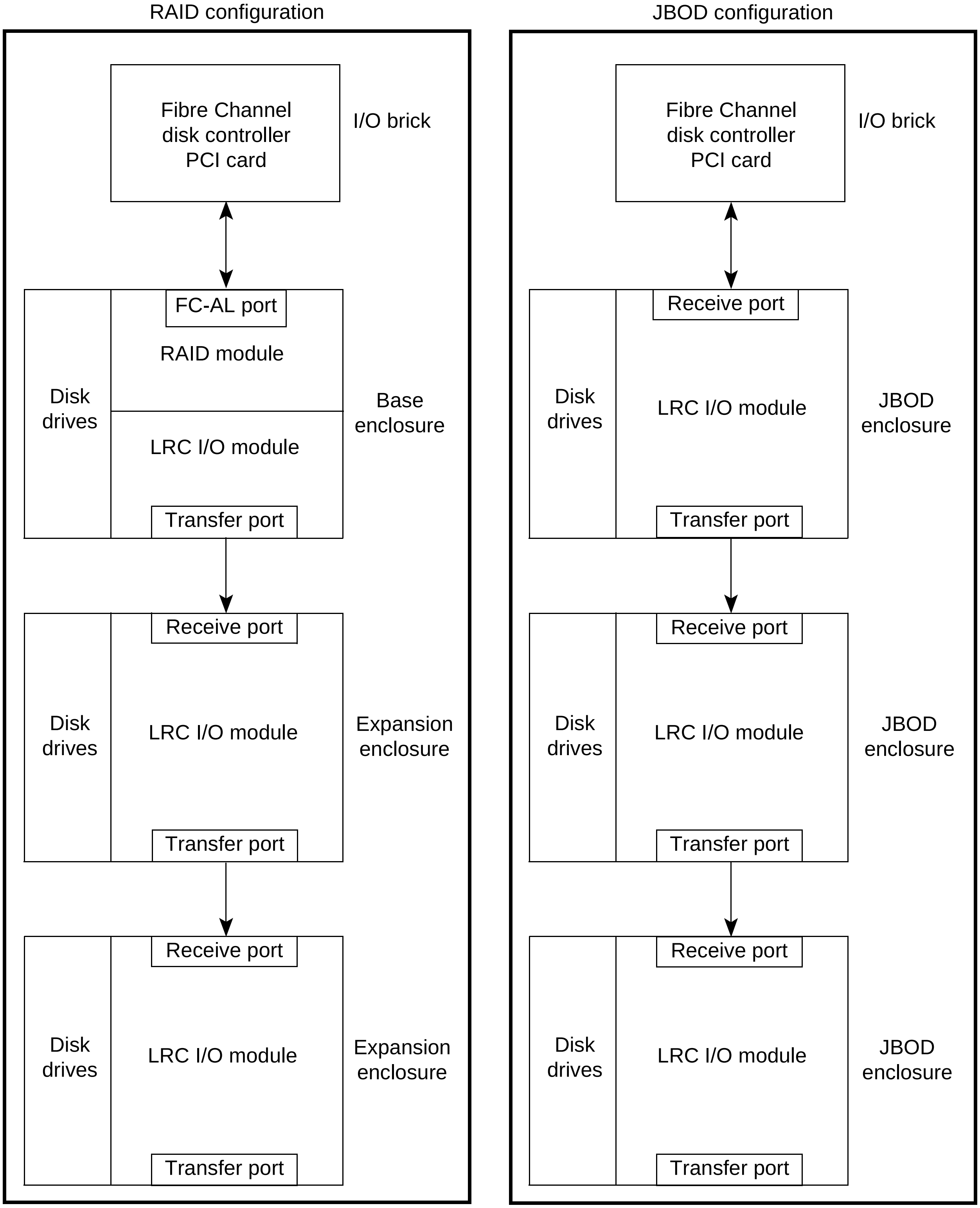


Figure 2-27 RAID and JBOD Configurations

2.5 Power Devices

The power devices of the SGI 3000 series servers consist of:

- Power distribution units (PDUs)

  A PDU receives power from a power receptacle and distributes this power to the power bays.

- Power bays that contain distributed power supplies

  Each power bay can contain between two and six distributed power supplies. The power bay supplies AC voltage to these power supplies and it also monitors and controls the power supplies.

  Each distributed power supply inputs the single-phase AC power and outputs 950 W at 48 Vdc. The outputs are bused together to provide 4,750 W of available power in an N+1 redundant configuration.

  Note: A minimum of two supplies must be present to provide the N+1 redundant configuration. The number of additional power supplies depends on the rack type. For example, the power bay of the short rack contains three power supplies; the power bays of the compute and I/O racks contain four power supplies.

  The power bay has eight output connectors that supply power to the bricks; for example, in Figure 2-28 a power cable connects one power bay connector to one C brick. Via this power connector and cable, the power bay can transfer 48 Vdc, 12-V standby voltage, and signals to the C brick.

- Voltage regulator modules (VRMs) and DC-to-DC converters

  The VRMs and DC-to-DC converters receive 48 Vdc from the power bay and convert it to the voltage levels that the brick components require.

The following text explains how the power devices supply power to the components within a brick:

When the power bay receives power from the PDU and the 12-V Enable switch for the brick is on, the power bay powers on the L1 controller by supplying a 12-V standby voltage to the brick.

Note: The power bay does not supply the 48 Vdc to the brick until the L1 controller signals it to do so.

The L1 controller signals the power bay to supply the 48 Vdc to its brick after a user activates the 48-V Enable for the brick. A user can activate the 48-V Enable for a brick via a hardware momentary button that is located below the L1 display or by entering a console command (*pwr u*).

When the brick receives the 48 Vdc, the L1 controller controls the application of the 48 Vdc to the VRMs and DC-to-DC converters within the brick. The VRMs and DC-to-DC converters convert the 48 Vdc to the voltage levels that the brick components require.

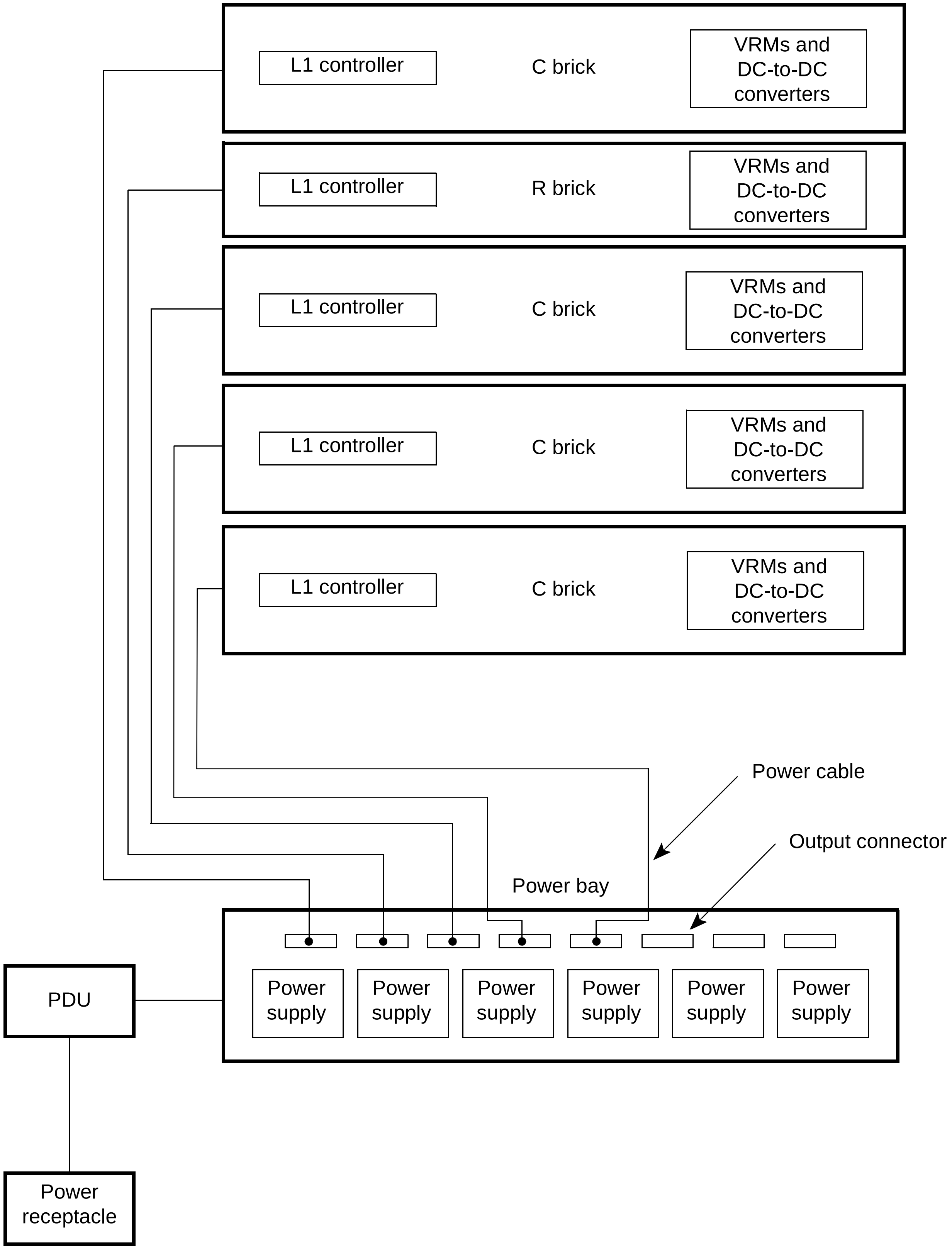


Figure 2-28 Power Block Diagram

2.6 Onyx3 Graphics System

The SGI Origin 3000 series servers can connect to a Silicon Graphics Onyx3 graphics system, which is also referred to as a G brick. The G brick supports two graphics pipes: one pipe consists of one Ktown2 connection, one GE16-4 board, one or two RM10 boards, and one DG5 board; the second pipe consists of one Ktown2 connection, one GE16-4 board, up to four RM10 boards, and one DG5 board. Each pipe requires connection to two processors within the SGI 3000 series servers.

The G brick also supports the InfiniteReality, InfiniteReality2, and InfiniteReality3 board sets; however, the InfiniteReality3 board set is the default board set for the Onyx3 graphics system (refer to Figure 2-29). The InfiniteReality3 board set consists of the following boards:

- Ktown2

  The Ktown2 board has two Crosstown2 connections; one connection for each pipe.

- GE16-4 (geometry engine)

  The GE16-4 board contains four processors that process OpenGL commands and vertex data that the GE16-4 board receives from the host processors. The GE16-4 board performs geometric transformations (for example, matrix calculations, scaling, and rotating), performs image processing, and subdivides complex polygons into triangles. Data output from the GE16-4 board passes through a Triangle bus to the raster manager subsystem.

- RM10 (raster manager or memory board)

  The RM10 board contains the main memory of the graphics system and 256 Mbytes of texture memory. The RM10 board also performs the following functions:

  - Scans and converts triangle data into 2x2 pixel quads
  - Performs texture mapping if specified
  - Organizes data into vertical spans and transfers it to the image processors
  - Fills the triangles and stores screen pixel data in the image processor array (framebuffer subsystem)

- DG5 (display generator)

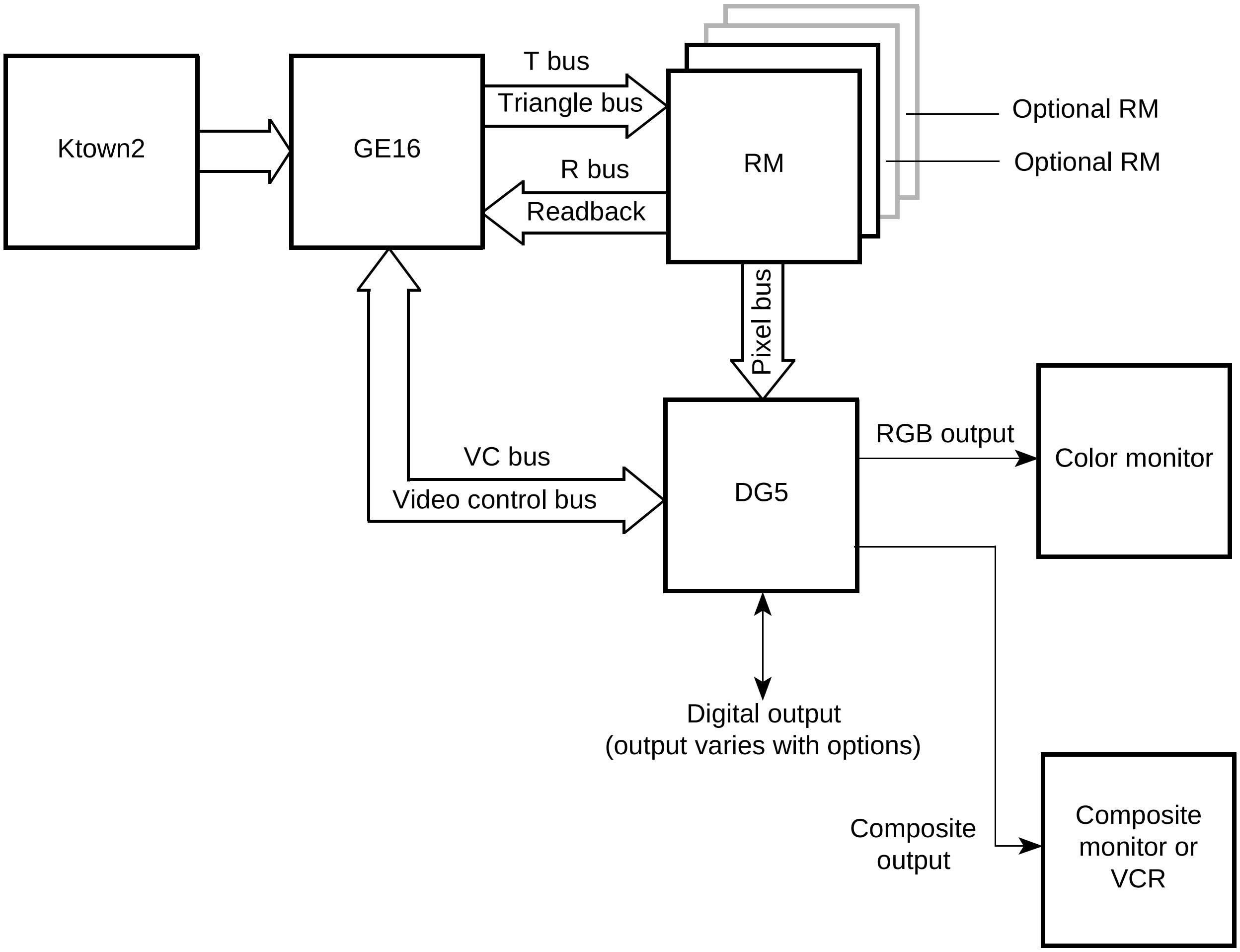
  The DG5 board inputs digital pixel data from the framebuffer subsystem via the video frontplane and converts the data into analog RGB or composite output that can be sent to a monitor or video tape recorder (VTR) device.

  There are several versions of the DG5 board:

  - DG5-2 has two graphics monitor connections.
  - DG5-8 has eight graphics monitor connections.
  - DG5-GVO (graphic-to-video option) provides the same features as the DG5-2 board and additional support for the CCIR601 digital video standard.
  - DG5-DPLEX provides the same features as the DG5-2 board and additional support for multiplex multipipe rendering.

The G brick also contains an L1 controller that performs various functions for the G brick, a power supply, and a midplane that connects the board set to the power supply and the L1 controller.

Note: For more information about the L1 controller, refer to the System Controllers subsection.



T bus transfers triangle data from GE16 to RMs for color fill.
R bus transfers readback data from RMs to GE16.
VC bus transfers video control information between the GE16 and DG5.
Pixel bus transfers pixels from the RMs to the DG5 for display.

Figure 2-29 G-brick Block Diagram

2.7 System Controllers

The SGI 3000 series servers contain three types of controllers that monitor and control the system: level 1 (L1), level 2 (L2), and level 3 (L3). The L1 controller is a brick-level controller that resides in the C, R, I, P, X, and G bricks. The L2 controller is a rack-level controller that resides in the compute rack(s). The L3 controller is a system-level controller; it is a Linux software package that resides on a Silicon Graphics 230 visual workstation.

The L2 controller is optional in the SGI 3200 server; however, it is required in the SGI 3400 and SGI 3800 servers. The L3 controller is optional in all SGI 3000 series servers.

When the system has an L2 controller, the L1 controllers are slave devices. When the system does not have an L2 controller, one of the L1 controllers is the master controller.

The controllers communicate with each other in the following manner:

- An L1 controller of an I/O brick communicates with an L1 controller of a C brick (refer to Figure 2-30).
- In an SGI 3200 server that has two C bricks (no R brick), the slave L1 controller of a C brick communicates with the master L1 controller of the other C brick (not shown in Figure 2-30).
- An L1 controller of a G brick communicates with the L2 controller via a USB connection.
- An L2 controller communicates with other L2 controllers or an L3 controller via an Ethernet hub.

Note: The following two bulleted items apply to systems that have one or more R bricks.

- An L1 controller of a C brick communicates with the L2 controller via the USB hub of an R brick.
- An L1 controller of an R brick communicates with the L2 controller via the USB hub of the R brick.

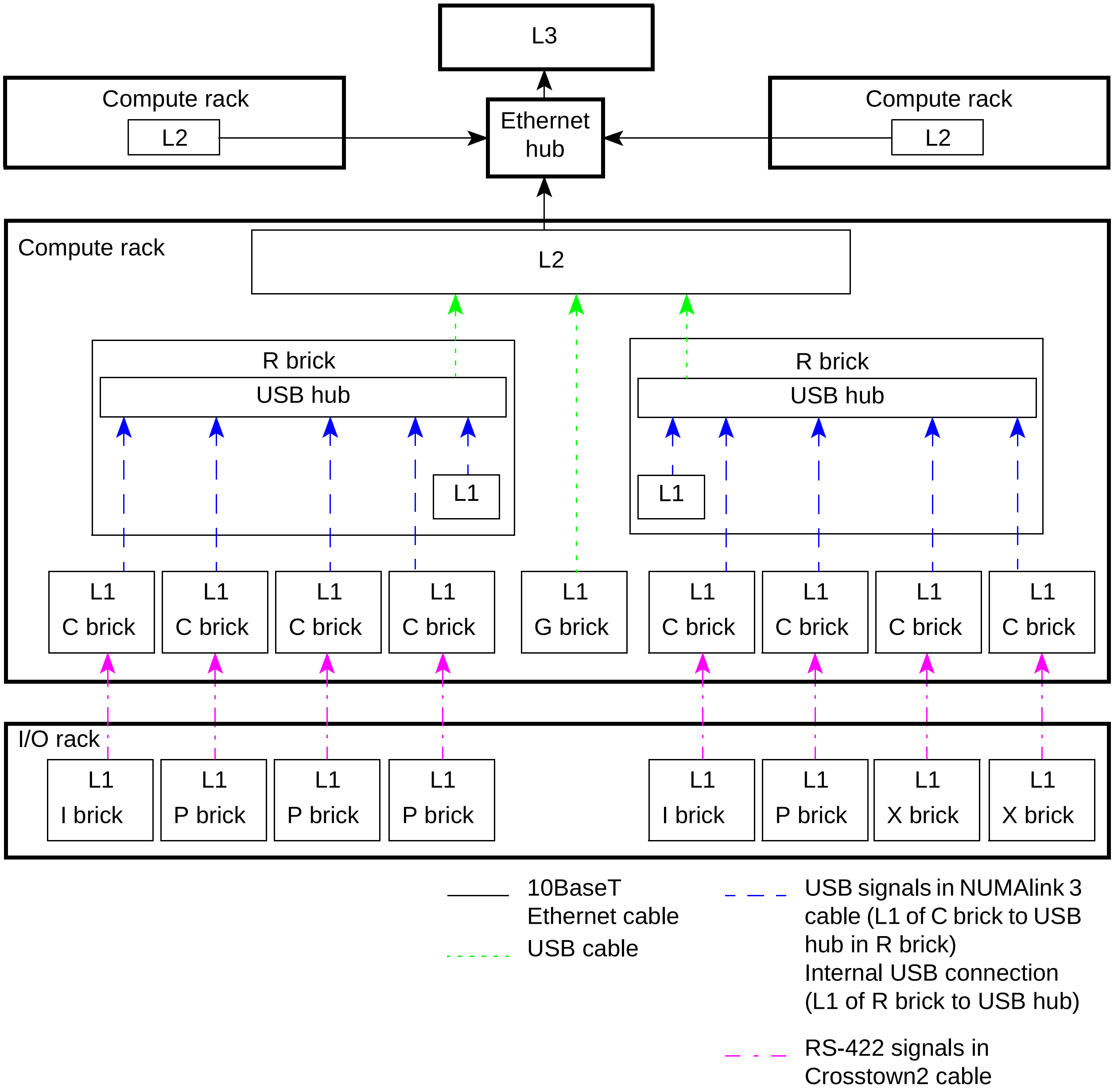


Figure 2-30 Controller Network

2.7.1 L1 Controller

The L1 controller performs many functions; some of the functions are common to the C, R, I, P, X, and G bricks and some are specific to a brick type. Table 2-2 lists some of the functions that the L1 controller performs. For information about all of the L1 controller functions, refer to the *System Controllers* document, publication number 108-0241-001.

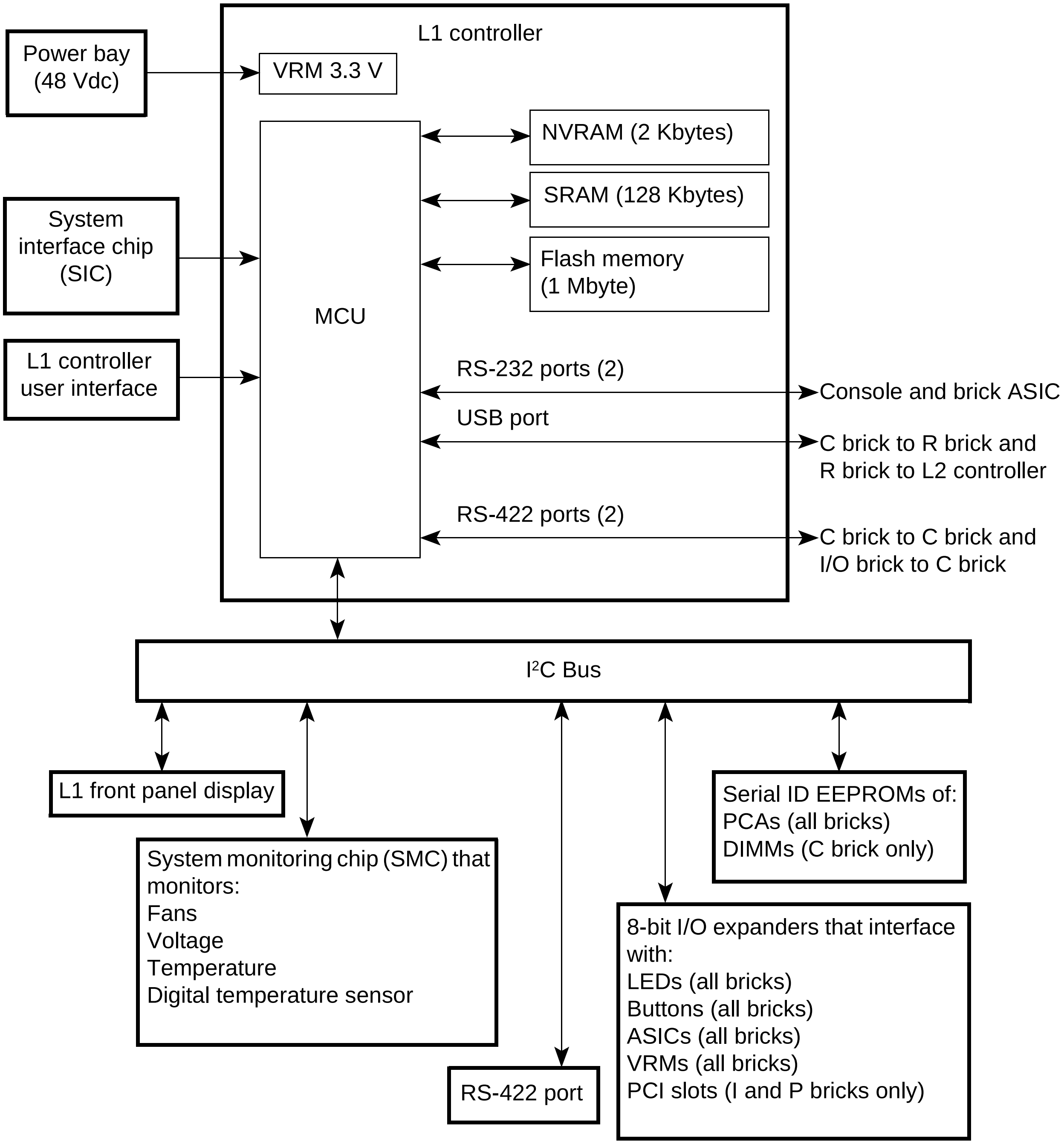
Table 2-2 L1 Controller Functions

Function	C Brick	R Brick	I Brick	P Brick	X Brick	G Brick
Controls voltage regulator modules (VRMs)	X	X	X	X	X	
Monitors voltage and reports failures	X	X	X	X	X	X
Controls voltage margining within the brick	X	X	X	X	X	X
Controls and monitors fan speed	X		X	X	X	X
Monitors and reports operating temperature and status of input power	X	X	X	X	X	X
Monitors and controls LEDs	X	X	X	X	X	X
Reads system identification (ID) PROMs	X	X	X	X	X	X
Monitors the Power On / Off switch, the Reset switch, and the Nonmaskable Interrupt (NMI) switch	X					
Monitors the Power On / Off switch		X	X	X	X	X
Provides a USB hub chip that has 6 master ports: one port connects internally to the R-brick L1 controller, four ports connect to the L1 controllers of four C bricks (via the NUMAlink 3 cable), and a master port connects to the L2 controller		X				
Reports the population of the PCI cards and the power levels of the PCI slots			X	X		
Calculates the power requirements of the PCI cards that are installed, compares this value to the available power, and determines which PCI slots will power up			X	X		
Powers up the PCI slots and their associated LEDs			X	X		
Reports the power levels of the XIO slots					X	
Controls the termination voltage margins of the XIO cards					X	

Figure 2-31 illustrates the basic block diagram of the L1 controller. It consists of:

- A front panel display
- A microcontroller unit (MCU)
- SRAM, NVRAM, and flash memory
- A voltage regulator module (VRM)
- An inter-integrated circuit bus (I^2C) bus
- RS-232, RS-422, and USB ports

Note: For some bricks, the L1 controller has more functionality than depicted by this illustration.



Power bay (48 Vdc)
L1 controller
VRM 3.3 V
NVRAM (2 Kbytes)
SRAM (128 Kbytes)
System interface chip (SIC)
Flash memory (1 Mbyte)
MCU
L1 controller user interface
RS-232 ports (2)
Console and brick ASIC
USB port
C brick to R brick and R brick to L2 controller
RS-422 ports (2)
C brick to C brick and I/O brick to C brick
I²C Bus
L1 front panel display
Serial ID EEPROMs of: PCAs (all bricks) DIMMs (C brick only)
System monitoring chip (SMC) that monitors: Fans Voltage Temperature Digital temperature sensor
8-bit I/O expanders that interface with: LEDs (all bricks) Buttons (all bricks) ASICs (all bricks) VRMs (all bricks) PCI slots (I and P bricks only)
RS-422 port

Figure 2-31 L1 Controller Block Diagram

2.7.1.1 Front Panel Display

The L1 controller has a front panel display that allows you to control and monitor the functional components of the system (bricks) (refer to Figure 2-32). The front panel display consists of:

- A 2-line, 12-character liquid crystal display (LCD) that provides:
  - Brick identification
  - System status
  - Warning of required service or failure
- Three momentary switches: On / Off, NMI, and Reset

  Note: The NMI and Reset switches are used only by the C brick.
- Three status LEDs: System On (green), Service Required (amber), and Failure (red)

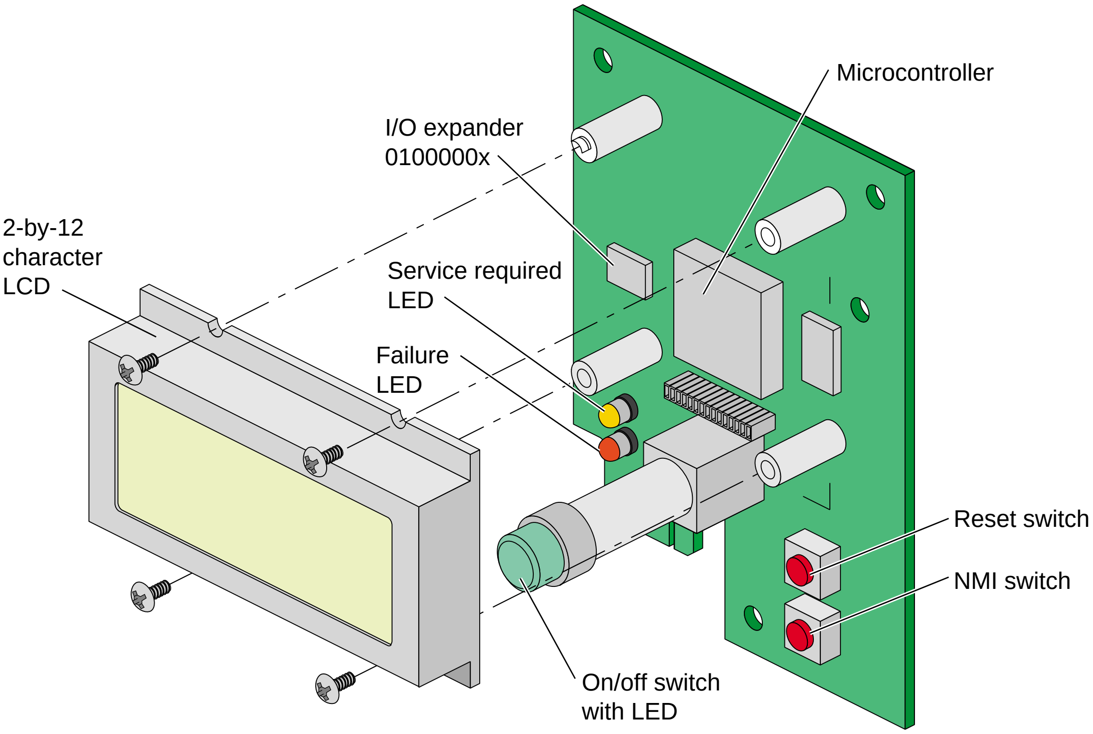


Figure 2-32 Front Panel Display

2.7.1.2 MCU

The MCU is a microprocessor that controls the flow of data and control signals among the components of the L1 controller. The MCU interfaces directly with a scan interface chip (SIC), which is the interface between the L1 controller and the JTAG boundary scan logic. The SIC has four scan ports.

For more information about the MCU, refer to the *System Controllers* document, publication number 108-0241-001.

2.7.1.3 Memory

The L1 controller has the following memory:

- 128 Kbytes of SRAM
- 1 Mbyte of flash memory that contains configuration information for the boot procedure
- 2 Kbytes of nonvolatile timekeeping random access memory (NVRAM) and a real-time clock

Note: The NVRAM is the only socket component on the L1 controller.

2.7.1.4 Voltage Regulator Module (VRM)

The L1 controller receives 12-V standby voltage from a rack power bay that a VRM regulates to 3.3 V. All components of the L1 controller require 3.3 V.

2.7.1.5 Inter-integrated Circuit (I^{2}C) Bus

The L1 controller interfaces with several components via an I^{2}C bus. The I^{2}C bus is a master-slave two-wire serial bus with a clock and a data line. Every transaction consists of an address followed by data. The I^{2}C bus has an interrupt line that the slave devices share. The MCU is the master of the I^{2}C bus.

Table 2-3 lists the slave devices of the I^{2}C bus.

Table 2-3 I^{2}C Bus Slave Devices

Slave Device	Description
Front panel display	The front panel display is a 2-line, 12-character alphanumeric display (located in the front of the brick) that provides brick identification, system status, and warning of required service or failure.
System monitor chip (SMC)	The SMC monitors environmental conditions. It has upper- and lower-limit registers that indicate acceptable environmental boundaries. When an environmental condition (temperature, fan speed, or voltage) exceeds a boundary, the SMC asserts an interrupt line that is connected to an I/O expander. The L1 controller polls this I/O expander. The SMC also performs the following actions: Monitors temperature Monitors five analog voltage inputs Monitors fan speed via two fan tachometer inputs Controls fan speed Compares all monitored values to upper- and lower-limit values Asserts an interrupt when a value exceeds the boundary value Polls all inputs every second
Serial ID EEPROMs	The serial ID EEPROMs contain identification (ID) information. For example, the DIMM EEPROM contains the part number, date code, and memory chip manufacturer. The PCA EEPROM specifies the board ID, revision, and JTAG scan topology information.
I/O expanders	The I/O expanders expand the MCU interface for I/O operations that are not time critical. For example, the L1 controller powers off a fan (powers off the 12-V VRM of the fan) via an I/O expander.

2.7.1.6 Ports

The L1 controller has five serial ports:

- Two RS-232 ports (console and diagnostic)
- Two RS-422 ports
- One USB port

The L1 controller of a C brick connects to the console (laptop or terminal computer) through one of the standard RS-232 ports; the console connects to the Console connector (DB-9 connector) of the C brick. The other RS-232 port connects to an ASIC within a brick (for example, hub ASIC of the C brick or router ASIC of the R brick).

The RS-422 ports are used to:

- Connect two C bricks in a system that does not have an R brick; the port routes console and diagnostic data and propagates resets, NMI, and power between the two C bricks. This connection occurs via the NUMAlink 3 cable.
- Connect a C brick to an I/O brick; this connection occurs via the Crosstown2 cable.
- Connect a brick to the SSI power management card; this connection occurs via the power cable.

The USB port connects a C brick to an R brick and an R brick to the L2 or L3 controller.

For example, the L1 controller in the C brick uses the RS-232 port to connect to the console, the USB port to connect to an R brick, and an RS-422 port to connect to an I/O brick (refer to Figure 2-33).

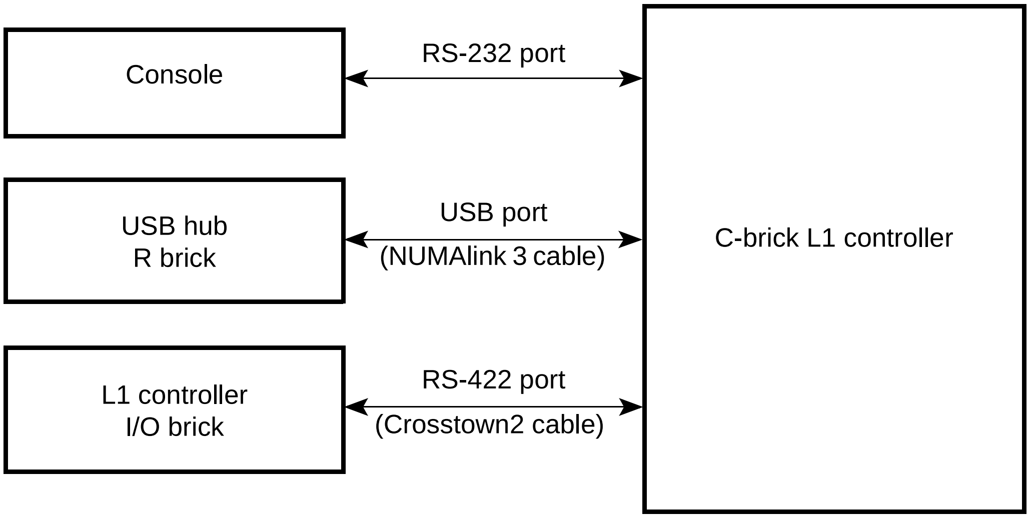


Figure 2-33 Serial Ports of the C-brick L1 Controller

2.7.2 L2 Controller

The L2 controller is a rack-level controller; it is a single-board computer that runs an embedded operating system out of flash memory.

A system requires an L2 controller when the system:

- Contains an R brick
- Will be maintained remotely
- Has a rack display

The L2 system performs the following functions:

- Maintains controller configuration and topology information between the L2 and L3 controllers
- Allows remote maintenance
- Controls resource sharing
- Controls L1 controllers
- Routes data between upstream devices and downstream devices (refer to Figure 2-34)

  Upstream devices (for example, rack display, console, and modem) provide control for the system, initiate commands for the downstream devices, and act on the messages that they receive from downstream devices.

  Downstream devices (for example, the USB hub of the R brick and L1 controllers of the bricks) perform the actions that are specified by the L2 controller commands, send responses to the L2 controller that indicate the status of the commands, and send error messages to the L2 controller.

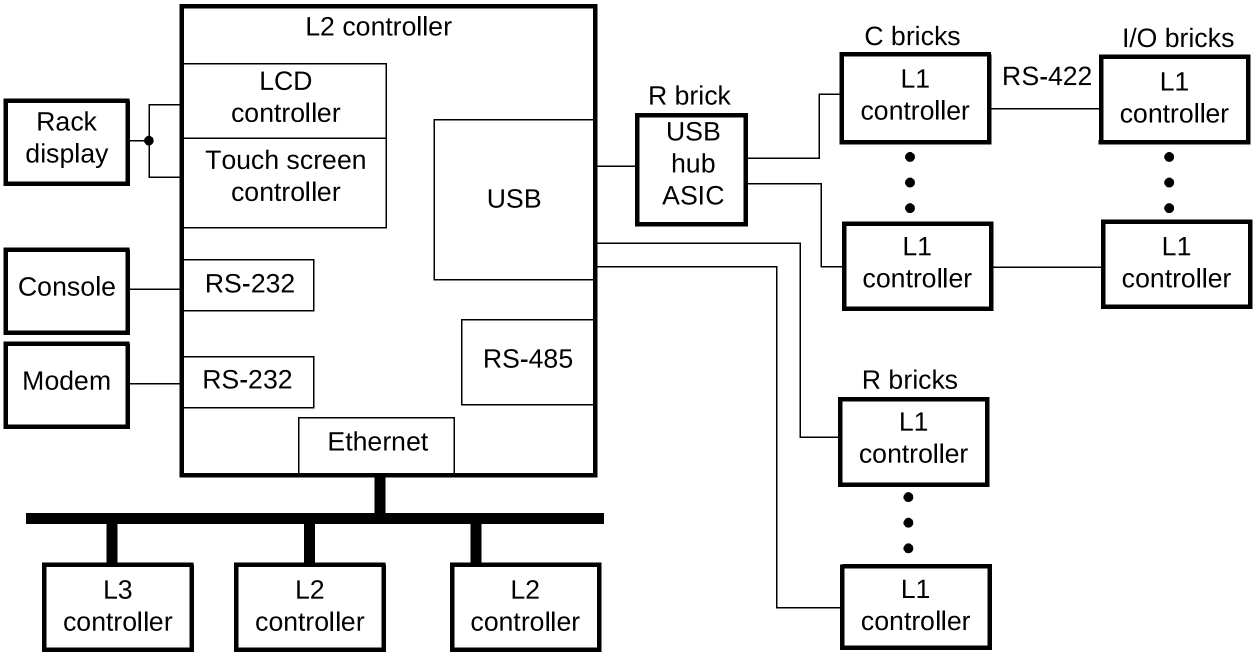


Figure 2-34 L2 Controller Block Diagram

A system can contain more than one L2 controller; for example, in an SGI 3800 server that contains 512 processors (eight compute racks), each compute rack has an L2 controller. When you issue a command to one of the L2 controllers, this command is propagated to all of the other L2 controllers in the system.

The L2 controller consists of rack display controllers, ports, and a software component.

2.7.2.1 Rack Display Controllers

The rack display consists of a touch-pad LCD-screen display. The L2 controller has a touch screen controller that translates what the user touches into commands. The L2 controller also has an LCD controller that displays the results of the commands.

2.7.2.2 Ports

Table 2-4 lists the ports of the L2 controller.

Table 2-4 L2 Controller Ports

Port	Connector Label	Description
Four standard downstream USB ports	L1 Ports 1 through 4	Normally, the USB ports connect the L2 controller to the USB hub of the R brick; however, when the system does not have an R brick, the L2 controller connects to an L1 controller of a C brick via a USB port.
10BaseT Ethernet port	Enet	The 10BaseT Ethernet port connects the L2 controller to an Ethernet hub.
Two RS-232 ports (DB-9)	Console and Modem	The RS-232 ports are the console and modem ports that allow the user to input text-based commands and to receive text-based results. The console and modem ports operate in one of the following modes: Normal mode - L2 controller interprets all commands from the console. Console mode - L2 controller forwards all commands to the system console, except for the commands that are prefixed with CTRL T; the L2 controller interprets these commands. L1 mode - L2 controller forwards all commands to the specified L1 controller, except for the commands that are prefixed with CTRL T; the L2 controller interprets these commands. L2 mode - L2 controller forwards all commands to the specified L2 controller. Remote access tool (RAT) mode - Legacy automated service tools use this mode.
One RS-485 port	ICMB	Not used.
Rack display port	LCD Display	The rack display port connects the L2 controller to the rack display.

2.7.2.3 Software Component

The L2 controller contains a software component that transfers data from a send client to the appropriate receive client (refer to Figure 2-35). The clients that the L2 controller communicates with are local to the L2 controller.

Note: Non-device tasks are also referred to as router clients.

The software allows the router clients to:

- Register with the router (indicates who the client is with a unique ID)
- Register to receive messages from other clients (local or remote)
- Receive commands and send corresponding responses
- Send commands and receive corresponding responses
- Receive messages that they are registered to receive

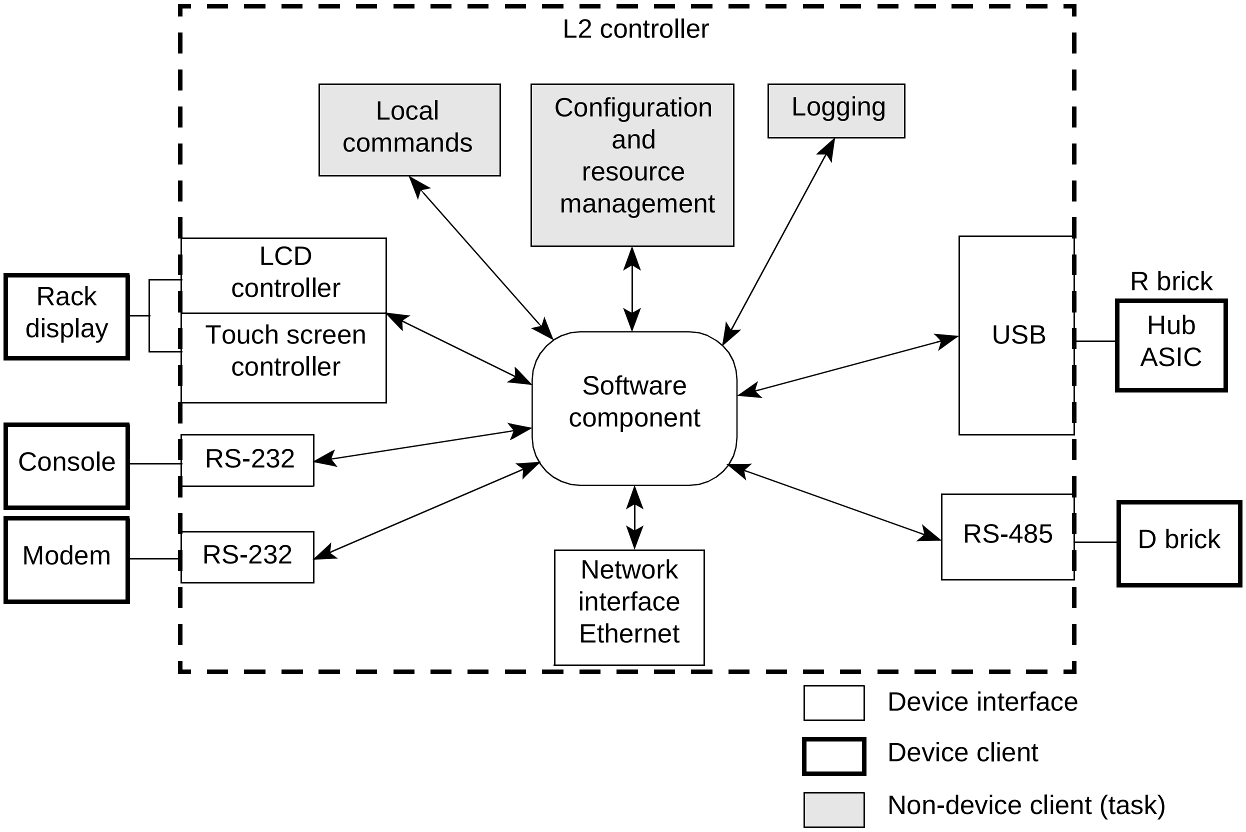


Figure 2-35 Software Component

The L2 controller logs the following information in separate files:

- Messages and command responses from the L1 controllers (includes the I/O bricks)
- Messages and output from the system console
- Debugging messages that the L2 controller produces
- Commands and responses from the rack display
- Messages and output that are sent to the console (attached to the L2 controller)
- Messages and output that are sent to the modem port (attached to the L2 controller)

2.7.3 L3 Controller

The L3 controller is a system-level controller. It is a Linux software package that resides on a Silicon Graphics 230 visual workstation. The workstation connects to one of two types of components: an Ethernet hub or a C brick.

The Silicon Graphics 230 visual workstation connects to the Ethernet hub through an Ethernet connector that is provided on the workstation. The Ethernet hub provides connections to the L2 controllers in the system.

The 230 connects to a C brick through a USB connector that is provided on the workstation. This configuration is used for small systems that do not have an R brick or an L2 controller.

The L3 controller provides the following maintenance, control, and administrative functions:

- Access to the following applications:
    - Scan control system (SCS) - Allows the user to control and monitor the onboard and offboard boundary scan test.
    - Environment control system (ECS) - Allows the user to monitor and control system power and cooling. The user can power on, power off, and reset the system. The user can monitor cabinet and brick temperatures, fan speeds, power status, and failure LED states.
    - Configuration control system (CCS) - Allows the user to manipulate the system configuration.
    - Maintenance control system (MCS) - Allows the user to execute sMDK (scalable Micro-diagnostic Kernel) diagnostics and view the results.
- Console for the L1 and L2 controllers
- Console for the system
- Remote support

Each L3 controller application that needs to send commands to or receive messages from the L2 controller must:

- Establish a connection to the network interface so that it can communicate with the software component of the L2 controller
- Send commands to and receive responses from the devices that connect to the L2 controller
- Register to receive messages from the devices that connect to the L2 controller

Chapter 3

Serial Numbers

Each SGI Origin 3000 series server has a system serial number (SSN) and each brick within the system has a brick serial number and public key.

3.1 System Serial Number

Each SGI 3000 series server has an eight-character SSN that consists of an “L” followed by seven digits (L0000001 through L9999998). The seven digits do not provide any information about the system hardware configuration.

Note: Serial numbers L0000000 and L9999999 are reserved for special purposes.

Manufacturing assigns the SSN to a system and places a label on the exterior of the system. Multiple-rack systems have a single SSN that is common to all racks.

Note: When a system in the field is upgraded to include an additional rack, the SSN label on the new rack will not match the SSN labels of the other racks within the system.

The SSN is stored in the NVRAM of each L1 controller. For the C brick, the NVRAM is located on the IP35 motherboard. For R, I, P, and X bricks, the NVRAM is located on the power board. For the G brick, the NVRAM is located on the controller board.

3.2 Brick Serial Number and Public Key

Bricks are identified by an alphanumeric serial number and public key. The brick serial number consists of three letters followed by three numbers (AAA000 through ZZZ999). The letters and numbers do not provide any information about the brick. The public key, which consists of three numeric fields, is created by a key generation software program. Manufacturing writes the public key into each brick.

The brick serial number is displayed on the board that houses the L1 controller logic; for example, the IP35 motherboard for the C brick, the power board for the R, I, P, and X bricks, and the controller board for the G brick. The public key is not displayed within the brick.

Both the brick serial number and the public key are stored in the brick EEPROM and the L1 controller NVRAM. For the C brick, the EEPROM and NVRAM are located on the IP35 motherboard. For R, I, P, and X bricks, the EEPROM and NVRAM are located on the power board. For the G brick, the EEPROM and NVRAM are located on the controller board.

3.3 System Serial Number Validation

In order for an R brick to power on, the BSN and SSN of the R brick must meet certain criteria. When the L1 controller is powered on, the L1 controller software performs the following status checks of the BSN and SSN:

- Verifies that the BSN in the NVRAM matches the BSN in the EEPROM. If the BSN in the NVRAM does not match BSN in the EEPROM, the brick does not power up.
- Verifies that the SSN of the brick is valid.
  - If the SSN in not in the form of L*nnnnnnn*, the brick does not power up.
  - If the SSN is L9999999 (a special SSN that disables a brick), the brick does not power up.
  - If the SSN is L0000000, the L1 software obtains the SSN from a neighboring brick, writes it to the L1 controller NVRAM, and allows the brick to power up.

    Note: The next time that the brick powers up, it will have the same SSN as the other bricks in the system.
- Verifies that the brick SSN matches the neighboring bricks. If the SSN does not match the neighboring bricks, the brick does not power up.

When an R brick needs to be added to a system or moved to a different system, SGI authorized personnel can clear the SSN that is stored in the L1 controller’s NVRAM. When the L1 controller is powered on in the new system, the L1 controller determines that the brick has a blank SSN and allows it to power on. The L1 controller obtains the SSN of the other bricks in the system and writes it to the NVRAM.

To clear the SSN, SGI authorized personnel use a temporary authenticator generation software program, which is located on a secure Web server that can be accessed via WebSAFE (use your SGI login and password). Using the brick serial number, this software generates an authenticator (four alphanumeric fields [total of 18 characters]) that is based on the brick public key and the current date and time.

The SGI authorized personnel input this authenticator into the L1 controller validation/serial number change software. The L1 controller software compares the authenticator to the brick public key. When the authenticator matches the brick public key and it has not expired, the L1 controller software clears the brick SSN. When the authenticator does not match the public key or if the authenticator has expired, the L1 controller software does not change the SSN.

Chapter 4

Data Integrity

The SGI Origin 3000 series servers support the following features that help maintain the integrity of data when it is transferred between components within the system (refer to Figure 4-1):

- Single-error correction, double-error detection (SECDED) error correction code (ECC)
- Link-level protocol (LLP)
- Parity

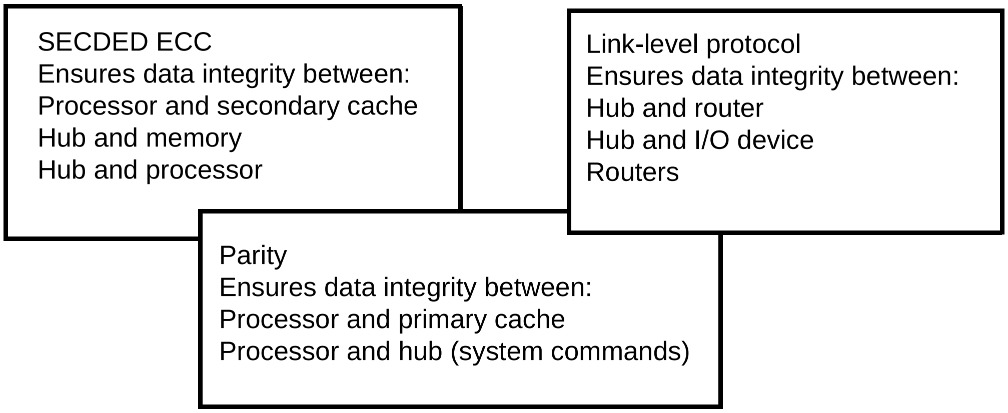

SECDED ECC
Ensures data integrity between:
Processor and secondary cache
Hub and memory
Hub and processor
Link-level protocol
Ensures data integrity between:
Hub and router
Hub and I/O device
Routers
Parity
Ensures data integrity between:
Processor and primary cache
Processor and hub (system commands)

Figure 4-1 Data Integrity Features

4.1 SECDED ECC

The SGI 3000 series servers use SECDED ECC to protect data when transferred to / from secondary cache, main memory, and directory memory (refer to Figure 4-2). The check bits are generated on writes and stored with the data in memory. When data is read from memory, new check bits are generated and compared (exclusively ORed) with the stored check bits. The result of the compare is called a syndrome. A syndrome of 0 indicates that the data is correct; no error occurred. A syndrome with an odd number of bits set to 1 indicates that a single-bit error occurred. A syndrome with an even number of bits set to 1 indicates that a double- or multiple-bit error occurred.

For main memory and secondary cache, 64 bits of data are protected with 8 check bits. For directory memory, the directory check bits are part of the data word. In 64-bit premium-directory mode, a 64-bit word has 57 data bits and 7 check bits. In 32-bit standard-directory mode, a 32-bit word has 26 data bits and 6 check bits.

The SGI 3000 series servers also use SECDED ECC to protect address and data when transferred between the hub and a processor.

Note: All SECDED errors cause interrupts, which are handled by the cache-exception interrupt handler.

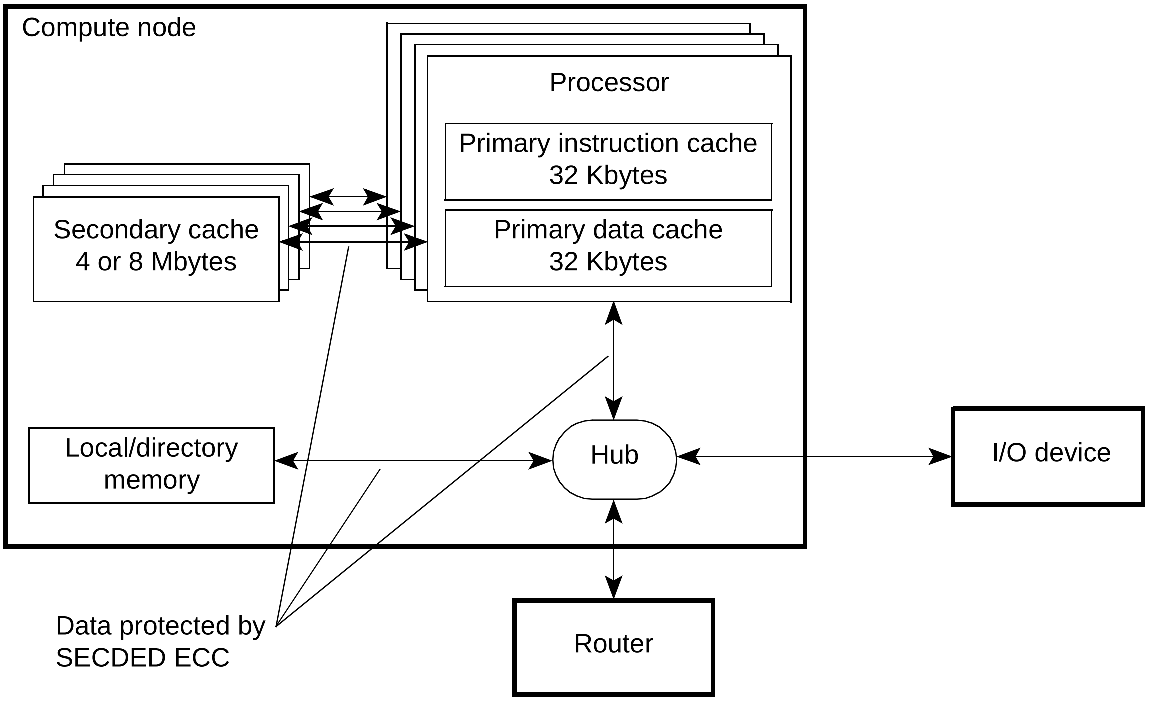


Figure 4-2 SECDED ECC

4.2 Link-level Protocol (LLP)

The SGI 3000 series servers use LLP to protect data when transferred between a hub and a router, a hub and an I/O device, and routers (refer to Figure 4-3). LLP is also the interface to source synchronous drivers (SSDs) and source synchronous receivers (SSRs) (refer to Figure 4-5).

The LLP provides:

- Detection of all single-, double-, and odd-bit signalling errors
- Detection of all burst errors up to 16 bits long
- Detection of lost or duplicate packets
- Recovery of all detected errors through retry

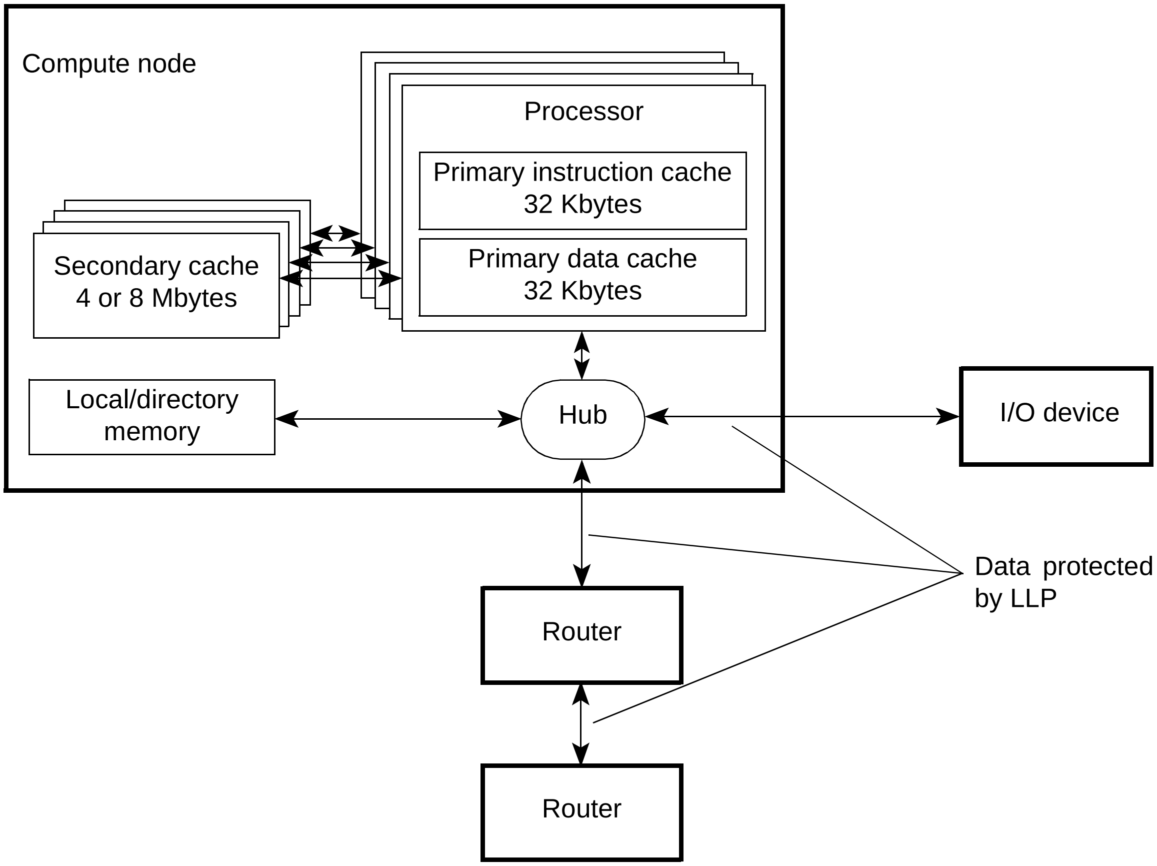


Figure 4-3 Link-level Protocol

For link-level protocol, data is transferred in micropackets; each micropacket contains 128 data bits, 16 error-detection check bits, 8 sequence number (SN) bits, and 8 sideband bits (refer to Figure 4-4). The transmit SN assigns a number to the micropacket. The receive SN provides acknowledge and flow control information from the local receiver to the remote transmitter. The sideband information contains packet framing and flow control information.

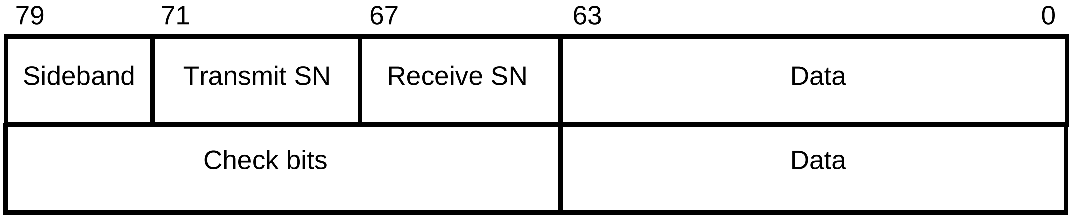


Figure 4-4 Link-level Protocol Packet Layout

The hub ASIC of the compute node, the router ASIC of the R brick, and the Xbridge ASICs of the I/O bricks use LLP to protect data (refer to Figure 4-5). Each of these components consists of:

- A source synchronous driver (SSD) - Receives 80 bits of data and control from the LLP send logic at a frequency of 200 MHz (NUMAlink 3) or 150 MHz (Crosstown2). The SSD divides the 80 bits into 20-bit transfers, which the SSD sends to the next brick at a frequency of 800 MHz (NUMAlink 3) or 600 MHz (Crosstown2).
- A source synchronous receiver (SSR) - Receives 20 bits of data and control at a frequency of 800 MHz (NUMAlink 3) or 600 MHz (Crosstown2). The SSR assembles four 20-bit transfers into an 80-bit transfer, which the SSR sends to the LLP receive logic at a frequency of 200 MHz (NUMAlink 3) or 150 MHz (Crosstown2).
- LLP receive logic - Checks the micropacket for errors. If it detects an error, it attempts to recover from the error by resending the micropacket via the LLP send logic. The LLP send logic continues to resend the micropacket until the LLP receive logic indicates that the micropacket is error free or until a retry limit is reached. If the retry limit is reached, the LLP send logic shuts down (ignores all future data) until it is reset.
- LLP send logic - Uses a 16-bit cyclic redundancy check (CRC) code referred to as CCITT to generate 16 check bits that protect 128 data bits. The LLP send logic places these bits in the micropacket and sends the micropacket to the SSD.

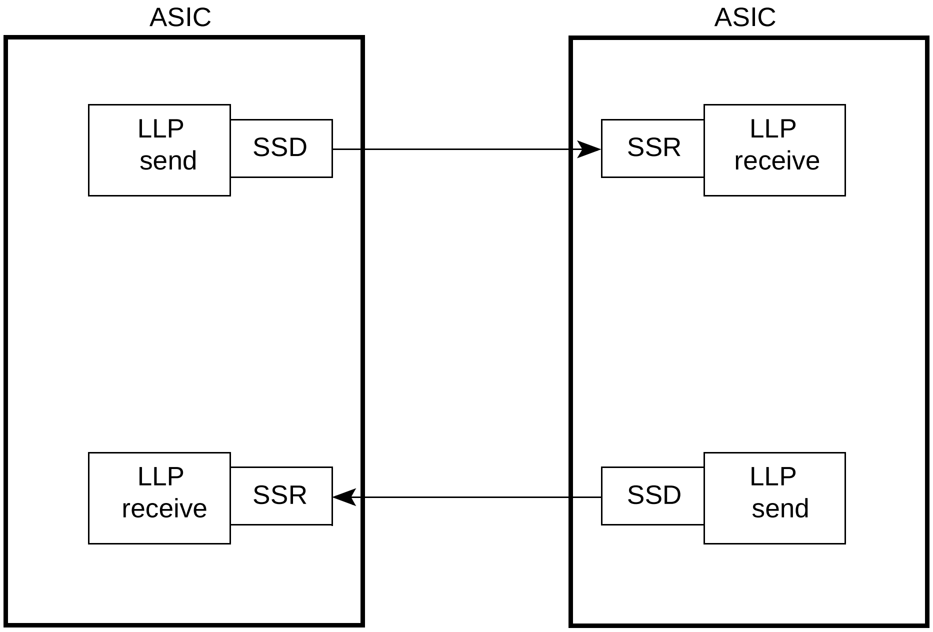


Figure 4-5 Link-level Protocol Logic

4.3 Parity

The SGI 3000 series servers use parity to protect data when it is transferred between a processor and primary cache and to protect system commands that are sent between the hub and a processor (refer to Figure 4-6).

For example, when the processor writes data to its primary cache, a parity bit is generated and is stored with the data in the primary cache. To create odd parity, the processor sets the parity bit to 1 when an even number of data bits are set to 1. To create even parity, the processor sets the parity bit to 1 when an odd number of data bits are set to 1.

All parity errors cause interrupts, which are handled by the cache-exception interrupt handler.

When a parity error occurs while a system command is sent between a processor and the hub during an address cycle, the processor interface ignores the command and address and sets the appropriate bit in the ERROR_INTERRUPT_PENDING (ERR_INT_PEND) register to a 1.

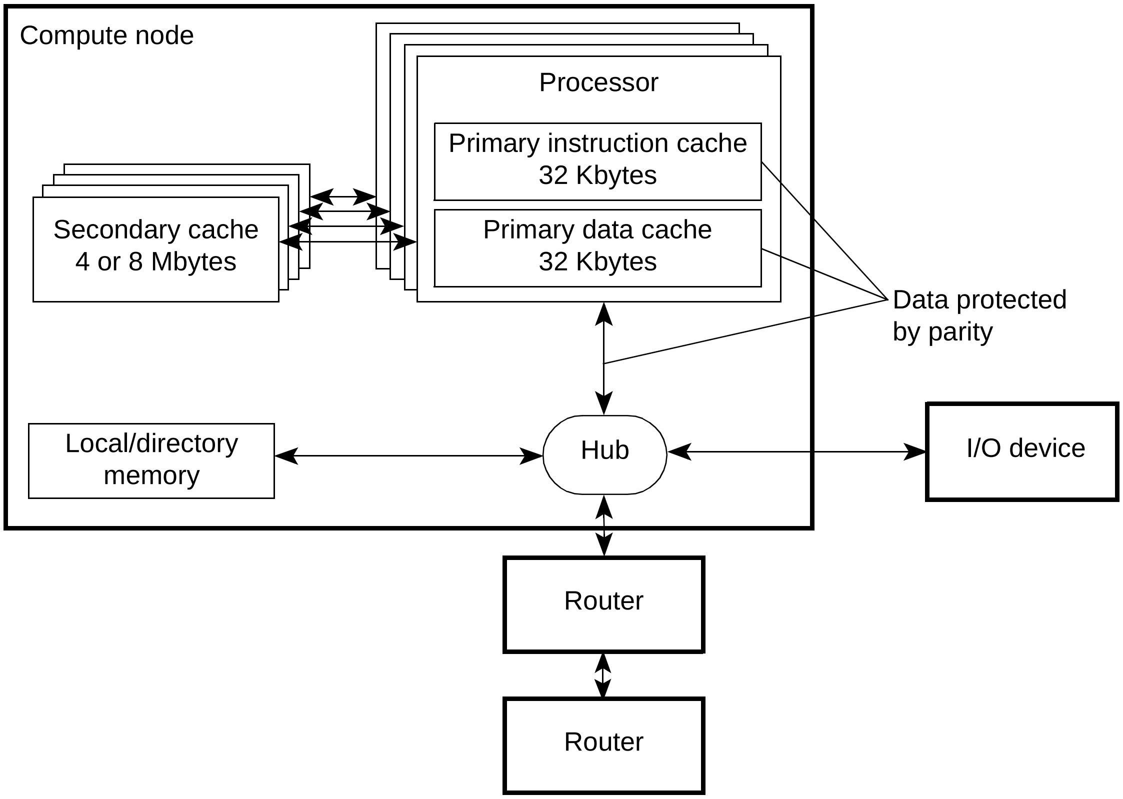


Figure 4-6 Parity

Reader Comment Form

Title: System Architecture
SGI™ Origin™ 3000 Server Series

Number: 108-0240-002

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

For what purpose did you primarily use this document?

_____Troubleshooting
_____Reference information
_____Other - please explain

_____Tutorial or introduction
_____Classroom use

____

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

_____Accuracy ____
_____Organization ____
_____Readability ____
_____Physical qualities (binding, printing, page layout) _________________
_____Amount of diagrams and photos ____________________________
_____Quality of diagrams and photos ____________________________

Completeness (Check one and explain your answer)

_____Too much information _____ Too little information _____Correct amount

____
____
____

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

____
____
____

NAME ______________________________
JOB TITLE ___________________________
E-MAIL ADDRESS ______________________
SITE/LOCATION _______________________
TELEPHONE __________________________
DATE ________________________________
[or attach your business card]

sgi™

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