Processor and Memory Components

HMM-305-B
CRAY J90se™ Series Systems
Last Modified: April 1998

Figures

Tables

Record of Revision

July 1996

Original printing.

May 1997

Added GigaRing™ channel numbering scheme for processor modules (refer to Table 4).

July 1997

Revised the online HTML and PDF versions to clarify a sentence in the "Memory" section on page 27.

April 1998

Revised to include the 512-Mword memory option.

Introduction

This document describes the processor and memory components of the CRAY J98se™, CRAY J916se™, and CRAY J932se™ computer systems (hereafter referred to as CRAY J90se™ series systems). A basic CRAY J90se series system includes one mainframe cabinet and one PC-10 I/O cabinet.

The mainframe cabinet may contain two, four, or eight memory modules and from one to eight processor modules. A memory module may be either half-populated or fully populated and can use 4-Mbit, 16-Mbit, or 64-Mbit dynamic random-access memory (DRAM) chips. Each processor module can support one GigaRing channel and up to 4 central processing units (CPUs), for a maximum of 32 CPUs per system. Refer to the *Hardware Overview* document for more information on system configurations, including memory options and sizes.

The CRAY J90se series processor and memory modules use application-specific integrated circuit (ASIC) technology. Each module is composed of an array of ASICs, which are based on very large-scale integration (VLSI) complementary metal oxide semiconductor (CMOS) technology.

The CRAY J90se processor module includes two major enhancements over CRAY J90 series processor modules: increased scalar performance and hardware support for the GigaRing architecture.

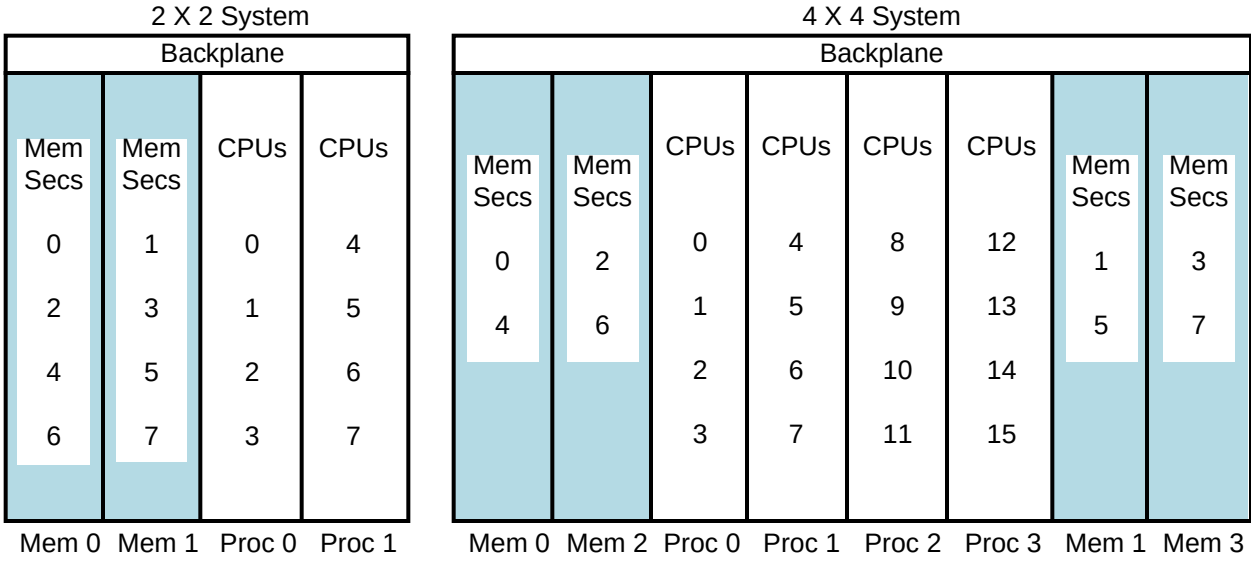
Backplane Configurations

The processor and memory modules connect to the CRAY J90se series system by means of a backplane that contains physical slots for the processor and memory modules. Three types of backplanes are available for CRAY J90se series systems: a 2 X 2 backplane for a CRAY J98se system, a 4 X 4 backplane for a CRAY J916se system, or an 8 X 8 midplane for a CRAY J932se system. These numbers refer to the maximum number of processor modules and memory modules that a system can include.

The 2 X 2 backplane configuration (refer to Figure 1) includes two memory modules and one or two processor modules (4 to 8 CPUs). The 4 X 4 backplane configuration (refer to Figure 1) includes four memory modules and from one to four processor modules (4 to 16 CPUs). The 8 X 8 midplane configuration (refer to Figure 2) includes eight memory modules and from one to eight processor modules (4 to 32 CPUs).

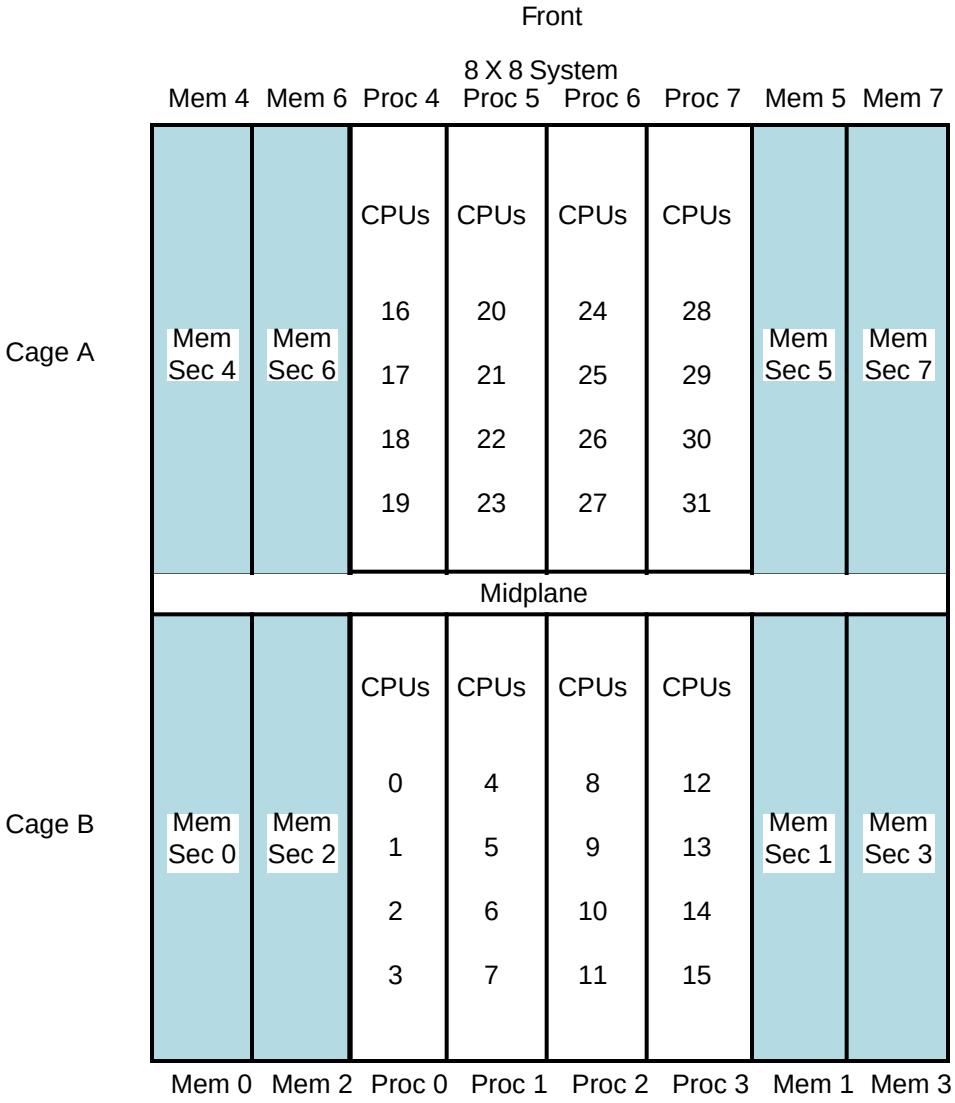
The memory module slots in all systems are always fully populated; however, some processor slots could be vacant.

Figure 1. 2 x 2 and 4 x 4 Module Slot Locations (Top View)



NOTES: Proc 1 slot may be vacant in 2 X 2 configurations.
Proc 1, Proc 2, and Proc 3 slots may be vacant in 4 X 4 configurations.
All memory module slots will always be filled.
The clock module is located across from the mainframe modules on the backplane.

Figure 2. 8 X 8 Module Slot Locations (Top View)



NOTES: Processor slots 1 through 7 may be vacant in an 8 X 8 configuration.
All memory module slots will always be filled.
The clock module is located beside the MEM0 module.

Processor Module

Each processor module (refer to Figure 3) contains 4 CPUs with one vector unit and one scalar unit per CPU. Each CPU also has a computation section that consists of operating registers, functional units, and a control section.

A processor module is approximately 16 in. X 20 in. and contains the logic for 4 CPUs. Each CPU has one high-speed port to each section of memory that supports 800-Mbyte/s write and read operations. The CI ASIC provides the I/O interface to the GigaRing interface.

A processor module has 20 layers: 8 signal layers with buried vias, 10 power/ground layers, and 2 surface layers. A processor module is composed of six printed circuit boards: a CPU board, a client interface board, a GigaRing interface board, a logic power module (LPM), and two small power modules that supply 2.6 Vdc to the PC ASIC. An integral aluminum framework that surrounds each processor module provides mechanical support, protects the components, and directs the airflow. Each CPU printed circuit board (PCB) is approximately 12 in. X 16 in. X 0.16 in. and contains ASICs in a 4 X 6 array. Refer to Figure 4 for a CPU block diagram. Refer to Figure 5 for a processor module block diagram.

Figure 3. CRAY J90se Processor Module

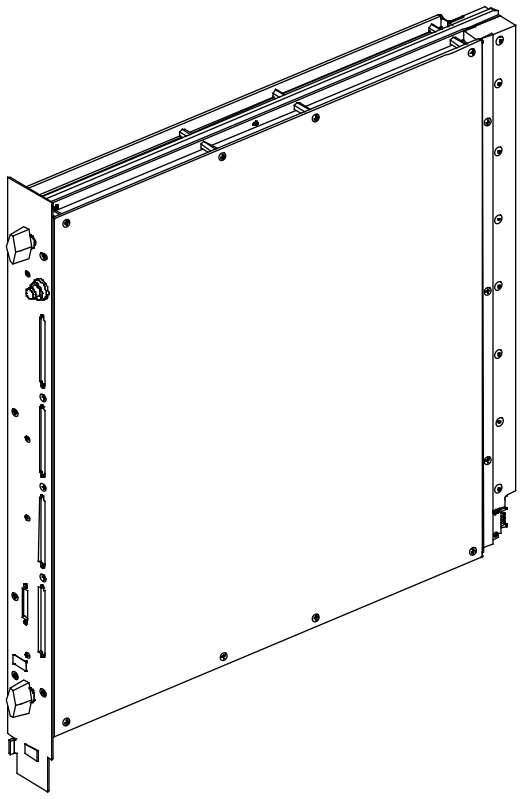
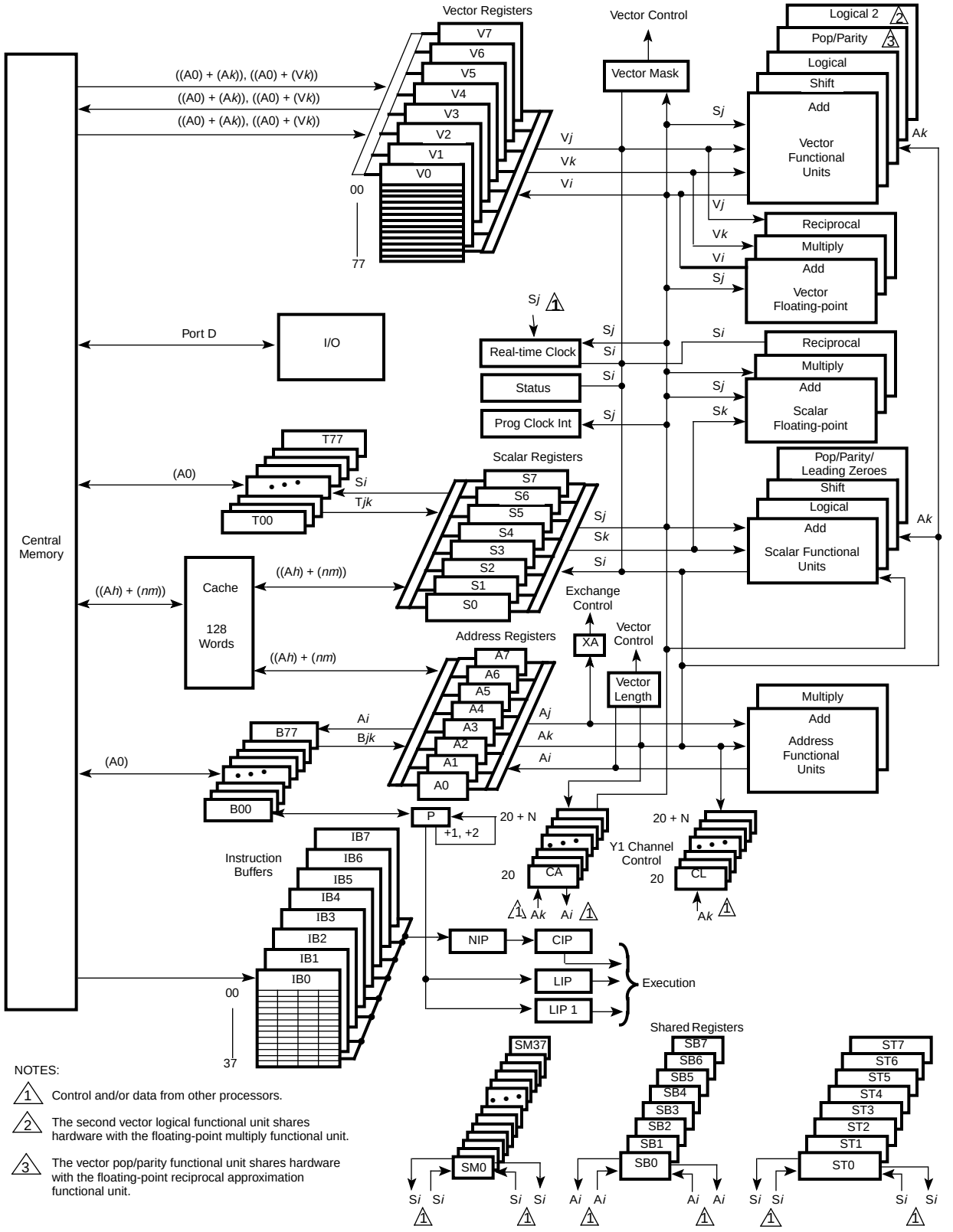


Figure 4. CPU Block Diagram



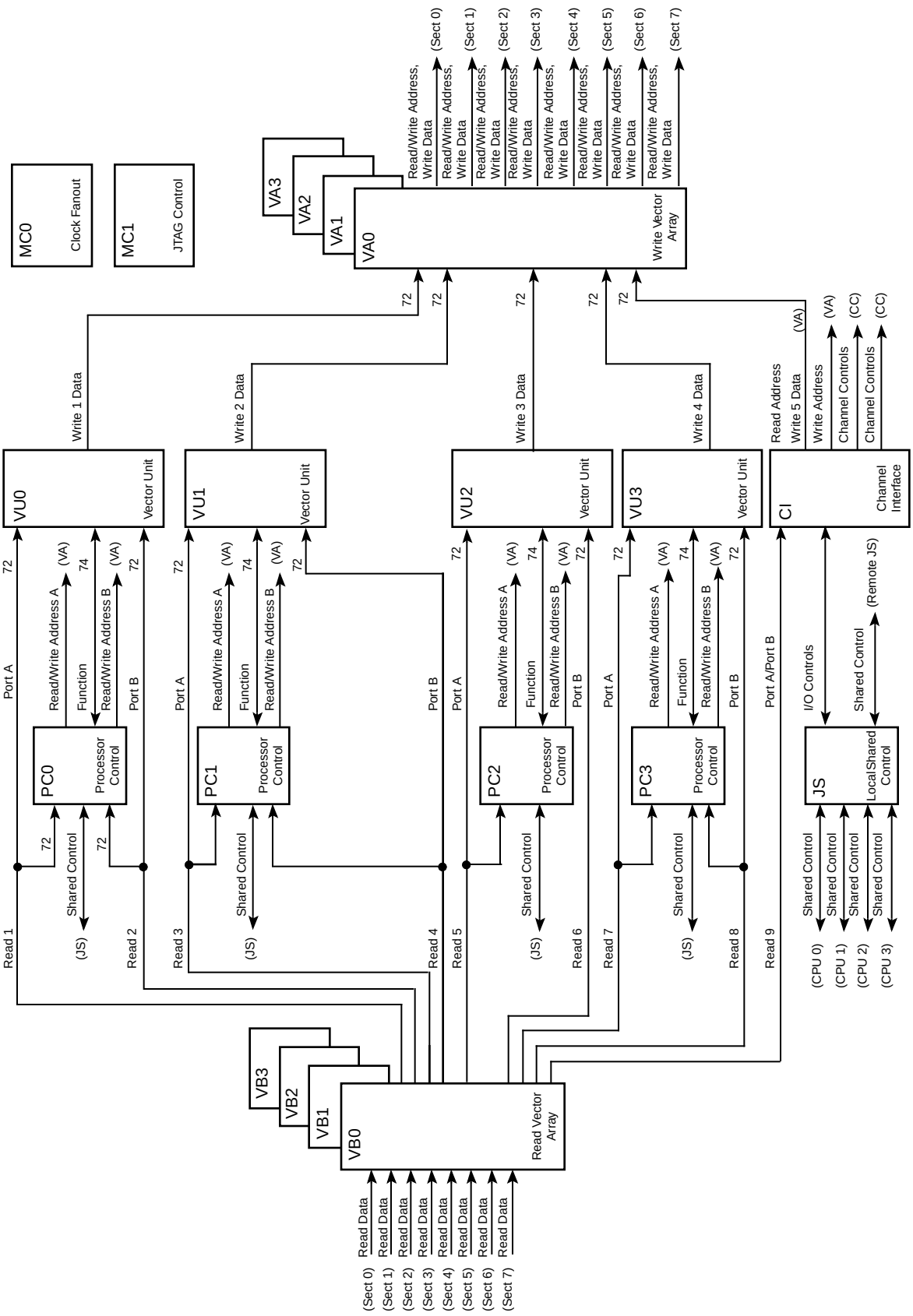


Figure 5. Processor Module Block Diagram (Represents 4 CPUs)

Processor ASIC Descriptions

Each CPU board contains the following eight types of ASICs, for a total of 20 ASICs on each CPU board.

VU

There are four vector unit ASICs (VU), each of which contains one full set (64 elements) of vector registers and associated vector and floating-point arithmetic units.

VA

Four vector array memory write interface ASICs (VA) control memory access and arbitrate memory conflicts. The VA ASICs send scalar, vector, and I/O requests to memory.

VB

Four vector array read data ASICs (VB) distribute memory read data to the requesting unit on the processor module.

JS

Each CPU board has one shared resources ASIC (JS), which includes the logic for interprocessor communication, shared registers, and interrupt control. The JS ASIC sends I/O memory errors to one of four local processors. Each JS ASIC contains a copy of the global real-time clock (RTC) and the status of each CPU. When the RTC is written, all global copies are updated at the same time. Each individual JS is then responsible for updating the copies of the RTC that are local to each PC ASIC.

PC+

Four processor control ASICs (PC+) contain the A and S registers, local RTC, instruction buffers, B/T registers, performance monitor, issue control, and scalar functional units, including scalar floating-point units. Each PC+ ASIC also contains a 128-word set-associative cache. The PC+ ASIC operates at twice the speed of the PC ASIC in CRAY J90 series systems. A special clocking scheme and additional interface logic synchronize the data that enters and leaves the PC+ ASIC.

The PC+ ASIC operates at 2.6 Vdc as compared to 3.3 Vdc and 5.0 Vdc for the rest of the processor module ASICs. A separate 2.6-Vdc power source for the PC+ ASIC is provided by two PC boards that reduce the 3.3 Vdc from the logic power module to 2.6 Vdc.

CI

One channel interface ASIC (CI) supports the functions of the GigaRing channel. The CI ASIC does not support true GigaRing protocol and therefore cannot operate at the full bandwidth that the GigaRing channel supports. The CI ASIC can support simultaneous read and write data transfers of approximately 200 Mbytes/s. Refer to the “GigaRing Client Interface” section for more information about I/O channel operations.

MC0´

One maintenance and clock fanout ASIC (MC0´) controls clock fanout to all modules in the system. Refer to the “System Clock” section at the end of this document for more information about clock functions.

MC1´

One maintenance and clock JTAG control ASIC (MC1´) provides Joint Test Action Group (JTAG), boundary scan, stop clock, maintenance channel, and reset maintenance functions.

Interprocessor Communications

The JS ASIC contains the shared resources logic for the CRAY J90se series systems. Each processor module has one JS ASIC. The following list describes the significant features of the JS ASIC:

- Each JS ASIC on each processor module contains a complete copy of all shared resources.
- Read operations of shared resources are done on the local JS ASIC.
- Write operations of shared resources are passed on to all JS ASICs.
- Shared registers are split into separate units to enable multiple concurrent access.

Each JS ASIC has a pair of buses that connects to each of the four PC ASICs on the processor module as well as a pair of buses that connects to the CI ASIC on that module. Each JS ASIC also has a dedicated bus to each of the other JS ASICs on other processor modules.

Shared Registers

When the PC+ ASCI generates a shared register read command, the local JS ASIC on the processor module receives the request and directs the command to its global logic. The global logic includes the shared registers, deadlock logic, set interprocessor interrupt (SIPI) control, I/O interrupt control, CPU status, and global real-time clock (RTC). The shared register read operation is held until all outstanding reads and writes from the same PC ASIC have completed; the data is then returned to the originating CPU. Because each JS ASIC has an identical copy of the shared resources, there is no need to pass the read request to the other JS ASICs.

When a shared register write command is received by the JS ASIC, the command waits for all outstanding read requests to complete. Then it must wait its turn to be sent on the dedicated bus that interconnects all JS ASICs. It must wait for access because other CPUs may also request access to the same resource. Once the request is granted, the write command is sent to all the JS ASICs in the system. No notification of completion is returned to the initiating CPU.

Real-time Clock

Each JS ASIC receives a copy of the global RTC. When the RTC is written, all global copies are updated at the same time. Each individual JS ASIC updates the copies of the RTC for each local PC+ ASIC. When the JS ASIC receives the global RTC load command, all other JS activity is halted. After the halt, the JS ASIC simultaneously transfers the new RTC value to all the PC ASICs.

Processor Status

The global logic of each JS ASIC contains the status of each CPU. The status includes the monitor mode flag bit, selected for external interrupts (SEI) flag bit, and the cluster number. Whenever any of these statuses change, the CPU must send the new value to the JS ASIC.

Test and Set Control

The test and set control logic handles test and set semaphore instructions for the processors. When a processor issues a test and set instruction, it sends a test and set command to the JS, which then passes it to the global logic using the dedicated bus that interconnects all JS ASICs. The global test and set logic contains the following information about each processor:

- Whether or not a processor is doing a test and set instruction
- The cluster number
- The semaphore register number

When the test and set logic receives the test and set command, it checks to determine whether the semaphore register is set. If the register is not set, the logic sets it and returns a completion status to the originating processor. If it is already set, the logic enters a wait-on-semaphore state and notifies the originating processor.

Whenever a clear semaphore (SM) or load SM instruction is executed, the status logic in the JS ASIC for each CPU determines whether the semaphore it is waiting on is cleared. If the semaphore register is cleared, the CPU makes a request to set it. One of the requesting CPUs is granted permission. The SM is set and the CPU is notified that the instruction has completed. Simultaneously, the processor that originated the test and set instruction is holding issue. It holds issue until it receives a response from the JS ASIC. If the JS ASIC returns a completion command to the CPU, then the test and set instruction issues and execution continues. If the JS ASIC returns a deadlock command, the program (P) register is backed up and the processor exchanges with the

deadlock flag set. A deadlock chain passes the waiting on semaphore (WS) bit and cluster number (CLN) for each CPU to processor status logic in each JS ASIC.

Interprocessor Interrupt

The interprocessor interrupt logic is part of the JS ASIC global logic. It routes interprocessor interrupts to the correct CPU. When a processor issues a SIPI instruction, it sends the command to the local JS ASIC, which then passes it to other JS ASICs. The interconnecting bus interface logic routes the SIPI to the correct processor.

I/O Interrupt

When an I/O interrupt occurs, the CI ASIC sends the interrupt command to the JS ASIC along with the interrupting channel number. The global logic receives the command and channel number and sends them on the dedicated bus that interconnects all JS ASICs. After the processor to be interrupted has been determined, the local JS ASIC sends the interrupt to the PC+ ASIC.

I/O Memory Errors

In the CRAY J90se series systems, the I/O channels do not share memory ports with specific processors. Each I/O channel is associated with the 4 CPUs that share the processor module. Through the scan configuration, one of the processors is selected to handle I/O memory errors. When a memory error occurs on I/O, the CI ASIC passes the error information to the local JS ASIC. The JS ASIC then sends the error to the CPU that has been configured to handle memory I/O errors.

Scalar Cache

A scalar cache memory enables portions of the main memory address space to be mapped into a small high-speed memory. Only scalar (A and S) references are cached. The cache is split into a number of lines or groups of data words, which represent contiguous blocks of main memory locations. Each cache line has an associated tag that identifies which main memory address the line represents.

The cache may be direct-mapped (in which case a given memory address may be mapped to only one position in the cache) or associatively mapped (in which the memory address may be mapped to any location in the cache). A combination of these mapping techniques is most practical and is called *set-associative*. The CRAY J90se series system uses this set-associative scheme.

The cache is a 128-word, 2-way, set-associative cache, which contains 128 1-word lines in 64 sets of 2 lines each. The line replacement algorithm is least recently used (LRU) on a per-set basis. The only cache instruction is a 0016*j*1 instruction that invalidates cache in processor A*j*. The entire cache is invalidated on an exchange or a cache flush operation.

The following list summarizes the general operation of cache:

- Scalar reads that hit a valid cache word do make memory requests, but these are redundant and are later aborted.
- Only scalar misses (read or write) allocate cache lines.
- Memory returns for scalar reads update the cache and pass the return data to the CPU.
- Scalar writes store through the cache.
- Vector writes invalidate matching cache lines.
- An exchange or flush invalidates the entire cache.

When a scalar write to cache operation occurs that results in a hit (or allocate), the referenced word is updated. When a scalar read to cache operation occurs that results in a hit, the referenced word is read and sent immediately to the CPU. A scalar read or write operation that missed the cache and cannot allocate a new cache line makes a normal memory request, which does not affect the cache.

Processor Control

The basic timing control for processor and memory modules is provided by the master clock/scan PCB. The processor cabinet contains the master clock/scan PCB. To run a program within the system, you must load the program into the selected CPU and then issue the program functions. This process includes using an exchange sequence, a fetch sequence, and an issue sequence. The exchange sequence brings the program parameters into the appropriate registers within the selected CPU. The fetch sequence, which immediately follows the exchange sequence, transfers a block of instructions from memory into the CPU's instruction buffers. After the instructions are loaded into the instruction buffers, the issue sequence invokes the instructions one at a time.

The program address (P) register indicates the instruction to be issued. The P register points to the location in the instruction buffer that contains the desired instruction; this causes that instruction to be decoded and then issued. When the selected instruction is issued, the P register increments and causes a new instruction to begin the decode process. This function continues until the P register cannot locate a desired instruction in the instruction buffers. When the P register cannot locate an instruction, another fetch sequence is initiated, which loads another block of instructions into the instruction buffers.

Exchange Mechanism

Each CPU uses an exchange mechanism to load programs or to switch from one program to another. This exchange mechanism uses a block of program parameters called an *exchange package* (refer to Figure 6), which contains the required addresses and limits imposed on the program. Table 1 lists the read mode bit definitions for the exchange package that Figure 6 illustrates.

Another component of the exchange mechanism is the *exchange sequence*, which is a basic CPU operation that loads a new exchange package and saves a copy of the current one.

Exchange Package

The contents of the exchange package (refer to Figure 6) include eight address (A) registers, eight scalar (S) registers, and the contents of specific parameter registers. Refer to Table 2 for descriptions of the acronyms and bits that represent these parameters. Table 3 describes the exchange package port and read mode value translations. The exchange package is a block of memory that contains the basic parameters for a particular program. This block of memory

is 16 words long and provides continuity when a program stops and restarts from one section of the program to the next, or when a program terminates and a new program is introduced.

Two new exchange package bits designate the processor type, which is either a CRAY J90 processor module or a CRAY J90se processor module. Exchange bits in word 7 (bits 30 and 31) are saved in bit positions 62 and 63 of the full word that is stored to memory. In the CRAY J90 series systems, both bits are 0's. In CRAY J90se series systems, bit 63 is 0 and bit 62 is 1.

The UNICOS® operating system compensates for the differences in performance between CRAY J90 processor modules and CRAY J90se processor modules. Also, some library routines have been adapted to allow for the differences in processor performance.

Exchange Sequence

The exchange sequence provides a process that deactivates the currently executing program and places its current operating parameters in memory for later retrieval. The next step in the process retrieves the operating parameters of the new program from a specified memory location and places them into the CPU exchange registers.

Figure 6. Exchange Package

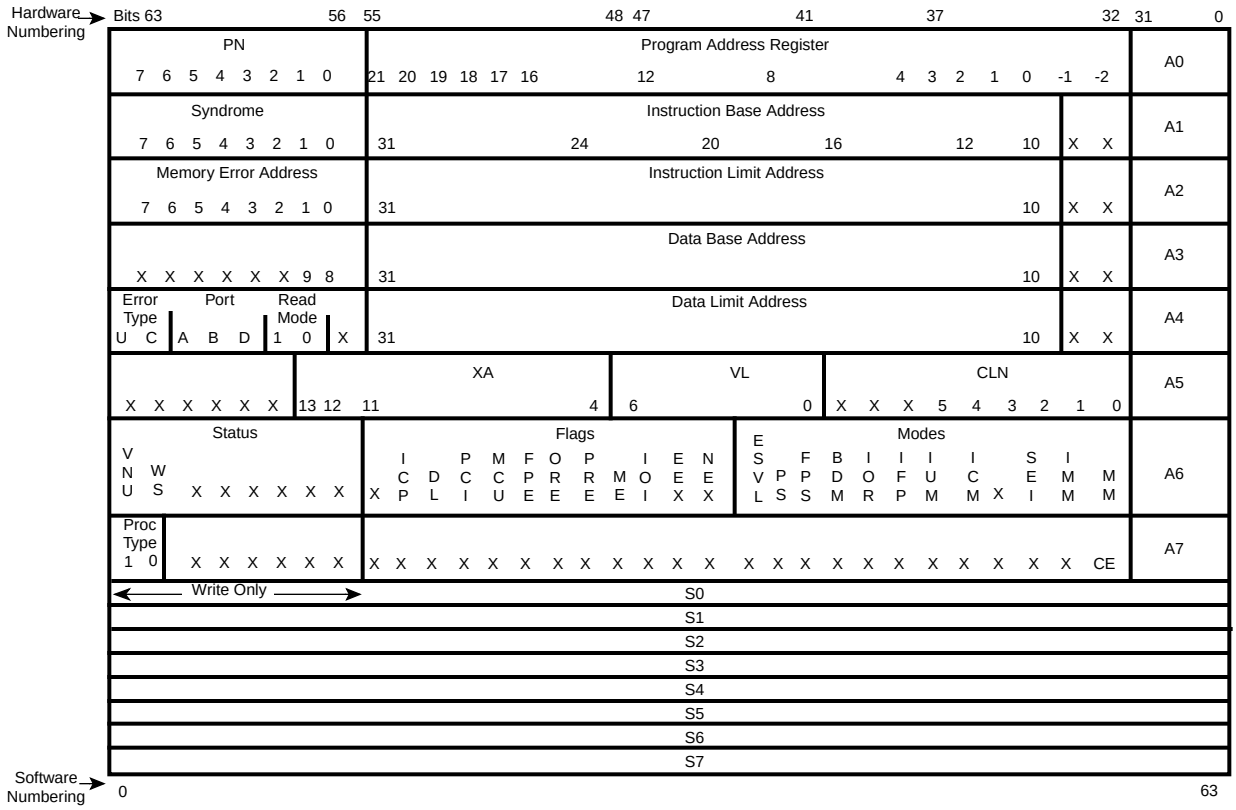


Table 1. Read Mode Bit Definitions

Bit 1	Bit 2	Port A	Port B	Port D Channel
0	0	EX	Fetch A	n+1
0	1	B	T	n+3
1	0	Vector	Vector	n+5
1	1	A/S	Fetch B	n+7

n = Processor module number X10 + 20

Table 2. Exchange Package Bit Assignments

		Bits		
Field	Word	Software	Hardware	Description
PN	0	3 – 7	60– 56	Processor number
P	0	8 – 31	55 – 32	Program address register
S	1	0 – 7	63 – 56	Syndrome bits
IBA	1	8 – 29	55 – 34	Instruction base address register
MEA	2	0 – 7	63 – 56	Memory error address
ILA	2	8 – 29	55 – 34	Instruction limit address register
MEA	3	6 – 7	57 – 56	Memory error address (continued)
DBA	3	8 – 29	55 – 34	Data base address register
E	4	0 – 1	63 – 62	Read error type
PORT	4	2 – 4	61 – 59	Port
RM	4	5 – 6	58 – 57	Read mode
DLA	4	8 – 29	55 – 34	Data limit address register
XA	5	6 – 15	57 – 48	Exchange address register
VL	5	16 – 22	47 – 41	Vector length register
CLN	5	26 – 31	37 – 32	Cluster number
VNU	6	0	63	Vector not used
WS	6	1	62	Waiting for semaphore
FLAGS	6	9 – 19	54 – 44	Flag register
MODES	6	20 – 31	43 – 32	Mode register
CE	7	31	32	Cache enable
A	0 – 7	32 – 63	32 – 0	Eight A registers
S	8 – 15	0 – 63	63 – 0	Eight S registers
PROC TYPE	7	0 – 1	62 – 63	Processor type

Table 3. Exchange Package Port and Read Mode Translations

Port Value	Mode Value	Type of Transfer when Error Occurred	Explanation
4 = A	0	EX	Error occurred while reading the exchange package
4 = A	1	B	Error occurred during a read to the B registers
4 = A	2	V	Error occurred during a vector read from memory
4 = A	3	A, S	Error occurred during a memory read to the A or S registers
2 = B	0	Fetch A	Error occurred during an instruction fetch operation on port A
2 = B	1	T	Error occurred during a block transfer to the T registers
2 = B	2	V	Error occurred during a vector read from memory
2 = B	3	Fetch B	Error occurred during an instruction fetch operation on port B
1 = D	0	Y1 or HIPPI	SECDED error occurred during a memory read for channel (n) output
1 = D	1	Y1 or HIPPI	SECDED error occurred during a memory read for channel (n + 3) output
1 = D	2	Y1 or HIPPI	SECDED error occurred during a memory read for channel (n + 5) output
1 = D	3	Y1 or HIPPI	SECDED error occurred during a memory read for channel (n + 7) output

n = Processor number

Processor and Memory Communication

Each processor module has a path to each of the 8 memory sections. To resolve conflicts, each of the 4 CPUs on a processor module contains a copy of all memory and shared register reservations.

In each CPU, the operating registers, instruction buffers, and exchange package have access to central memory through memory ports. Each CPU has two ports, port A and port B, to enable up to two simultaneous memory references from each CPU (two memory read operations or one read operation and one write operation). Port D handles I/O and instruction fetch operations.

Power PCB

Each processor module has its own logic power module (LPM), which is located on the back side of the processor module. The LPM receives 48-Vdc power and converts it to the closely regulated power that the processor ASICs require.

The PC+ ASIC operates at 2.6 Vdc as compared to 3.3 Vdc and 5.0 Vdc for the rest of the processor module ASICs. A separate 2.6-Vdc power source for the PC+ ASIC is provided by two PC boards that reduce the 3.3 Vdc from the logic power module to 2.6 Vdc.

The onboard power board is also used on memory modules. On a memory module, a converter supplies 5-Vdc power for the DRAM chips and 3.3-Vdc power for the ASICs. Both power sources are provided in n + 1 units for improved reliability. The LPM also provides several local control functions such as voltage margining, local inhibit, and air-stream overtemperature sensing.

GigaRing Client Interface

Channel Overview

CRAY J90se series systems use a GigaRing client interface for all I/O operations. The client interface connects the processor board to the GigaRing node chip. Refer to Figure 7. The client interface supports four GigaRing channel operations: message in, message out, direct memory access (DMA) in, and DMA out. The DMA operations can be either master DMA or slave DMA. There are three communication buses between the client interface and the processor module: DMA bus, CPU commands bus (CCSBUS), and console bus (CONBUS). The CONBUS transmits serial scan latch data for master clear operations and boundary scan testing.

Figure 7. CRAY J90se Series Client Interface

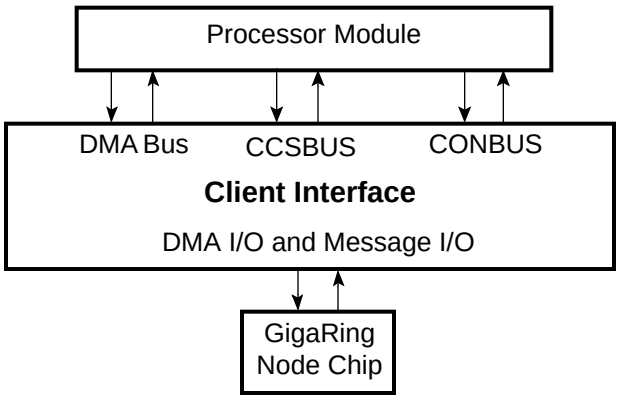


Table 4 lists the GigaRing channel numbers for each processor module. One octal GigaRing channel number corresponds to each processor module. The available channel numbers for each system configuration are listed in the UNICOS `param` file. Use these channel numbers when you add GigaRing channels to the system.

Table 4. GigaRing Channel Numbering

Channel Number	Processor Module
024	Processor Module 0
034	Processor Module 1
044	Processor Module 2
054	Processor Module 3
064	Processor Module 4
074	Processor Module 5

Table 4. GigaRing Channel Numbering (continued)

Channel Number	Processor Module
104	Processor Module 6
114	Processor Module 7

Software Overview

A transfer information block (TIB) controls all channel operations within the CRAY J90se series GigaRing client interface. The TIB contains information that is necessary for the GigaRing client interface to handle messages and DMA transfers. There are three TIBs: message in, message out, and DMA transfer. A TIB can be up to sixteen 32-bit words in size and contains all the information needed to start and complete a data transfer. The TIB is not updated when a transfer is completed; instead, the transfer operation creates a task completion block (TCB) to reflect what happened during the transfer. The TCB contains the ending status of the transfer along with any error information. TCBs can also be sixteen 32-bit words.

Error Detection

A parity circuit protects the client interface's memory and first-in-first-out (FIFO) circuitry. Messages from the CI chip have parity generated by the client interface. The GigaRing chip receives the parity and checks for errors. If there is an error, the packet is lost and a GigaRing chip parity error flag bit is set.

Each DMA parity error is handled differently, depending on the type of request. Write request DMA parity errors are communicated back to the GigaRing target. Read response DMA parity errors are communicated back to the DMA engine that is doing the transfer. The DMA engine shuts down the transfer and sets the error bit in the associated TCB. All other DMA packets with parity errors are detected by the GigaRing node chip, which sets the parity error flag.

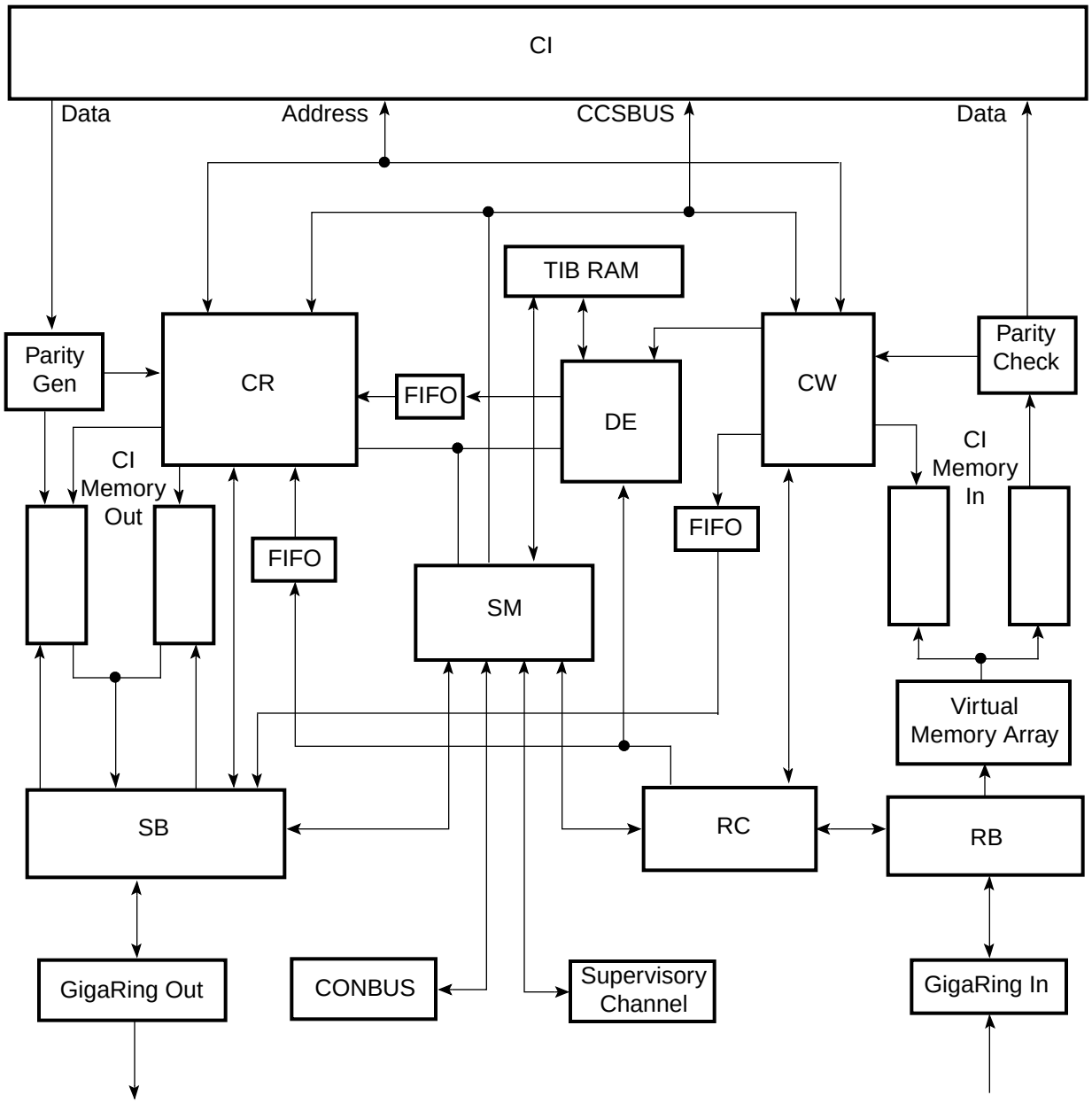
Hardware Overview

Each processor module in the system can have its own GigaRing client channel interface. The client interface communicates with the CI ASIC on the processor module. Two printed circuit boards comprise the client interface. The GigaRing client interface hardware is described in Table 5 and illustrated in Figure 8.

Table 5. Client Interface Hardware Descriptions

Function	Description
Virtual Channel Memory	To optimize channel performance, the GigaRing node chip delivers data on four virtual channels in any order. Data on these four channels may have several different destinations. A virtual channel memory array receives the data from the RB chip and holds the virtual channel data while a packet is being received.
CI Memory In	A split memory array is used to hold data going to the CI ASIC. The headers of write requests, read responses, and all of the block done requests are written into one memory array. The bodies of write requests read responses, and messages are written into the other memory array.
CI Memory Out	A split memory array is used to hold data coming from the CI ASIC. The headers of write requests, read responses, and block init responses are written into one memory array. The bodies of write requests, read responses, and messages are written into the other memory array.
CW - CI Write Control	The CW chip receives write information from the RC chip and controls all write operations to the CI ASIC. The CW also contains TIB pointers.
RB - GigaRing Receive Buffer Control	The RB is the receiving interface to the GigaRing node chip. The RB breaks down incoming virtual channel data into packets and places them into a virtual channel memory array; it also checks for errors.
RC - Virtual Channel to CI Memory In Control	The RC chip receives data destination information from the RB chip after the entire packet has been received from the GigaRing node chip and placed in the virtual channel memory. The RC chip will move the data to the CI memory in array, or to the CR request FIFO, or to the DMA engines.
DE - DMA Read and Write Engines	The DE chips are the DMA read and write engines. The DE interprets the TIBs that contain all the information needed to start and complete a transfer. The DMA engines are controlled by a stoker and maintenance control chip (SM).
SM - Stoker and Maintenance Control	The SM chip contains the TIB pointer to the master DMA TIBs. The SM will attempt to post the TIBs to a data transfer engine as engines become available.
CR - CI Read Control	The CR chip receives request information from the RC chip as well as requests from the DE chip. The CR controls all read operations to the CI ASIC and loads the CI memory out array with packets and headers.
SB - GigaRing Send Buffer Control	The SB chip is the send interface to the GigaRing node chip. The SB is informed when packets have been written into the CI memory out array, DMA engine FIFO, or CW response FIFO. The SB controls data flow through the four GigaRing virtual channels.

Figure 8. Client Interface Block Diagram



Memory

Central memory consists of two, four, or eight memory modules that provide from 64 to 4,096 Mwords of memory. Central memory has a peak memory bandwidth of 12.8 Gbytes/s for a 2 X 2 backplane, 25.6 Gbytes/s for a 4 X 4 backplane, and 51.2 Gbytes/s for an 8 X 8 midplane. Each memory word consists of 72 bits: 64 data bits and 8 check bits for error detection.

The memory components are distributed across all memory modules; the number and type of components depend on the system configuration. Refer to Figure 13 for a memory module block diagram. DRAM chips provide storage for data and correction bits. The DRAM chips have a 70-ns access time.

Refer to Figure 9 and Figure 10 for illustrations of half-populated and fully populated memory modules in a 4 X 4 system. In a 2 X 2 or 4 X 4 backplane configuration, central memory is divided into 8 sections. Each memory section in a 4 X 4 system contains 4 subsections, and each memory section in a 2 X 2 system contains 2 subsections. Each subsection contains 16 pseudobanks when fully populated and 8 banks when half populated.

In a 2 X 2 system, a fully populated memory contains 256 (decimal) banks, and a half-populated memory contains 128 (decimal) banks. In a 4 X 4 system, a fully populated memory contains a total of 512 (decimal) banks; a half-populated memory contains 256 (decimal) banks. Each central memory bank can be accessed once every 14 CPs.

Refer to Figure 11 and Figure 12 for illustrations of half-populated and fully populated memory modules in an 8 X 8 system. In an 8 X 8 midplane configuration, central memory is also divided into 8 sections. Each memory section contains 8 subsections. Each subsection contains 16 pseudobanks when fully populated and 8 banks when half populated. A fully populated memory contains 1,024 (decimal) banks, and a half-populated memory contains 512 (decimal) banks.

Figure 9. Memory Module Layout - 4 x 4 Half Populated

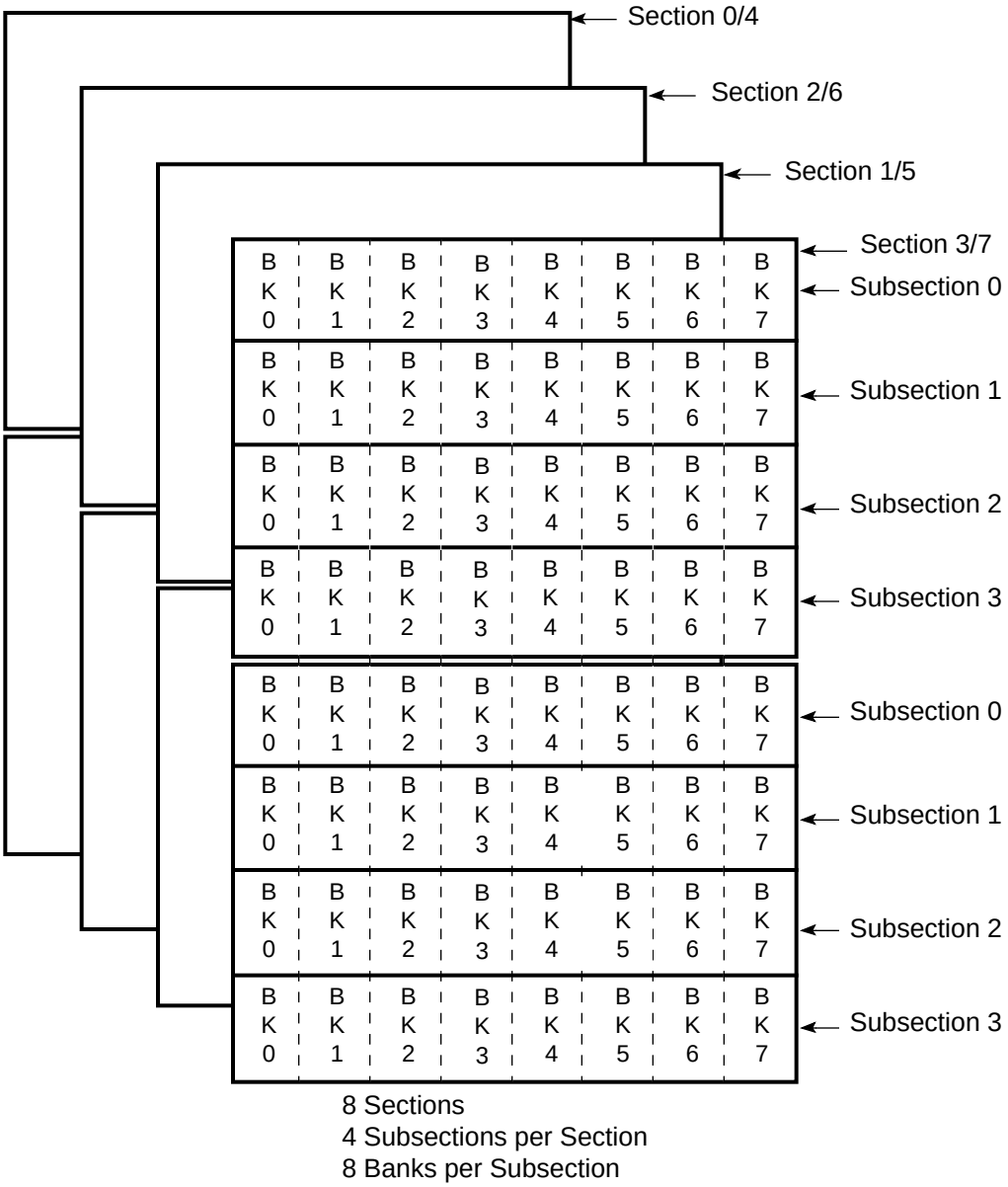


Figure 10. Memory Module Layout - 4 x 4 Fully Populated

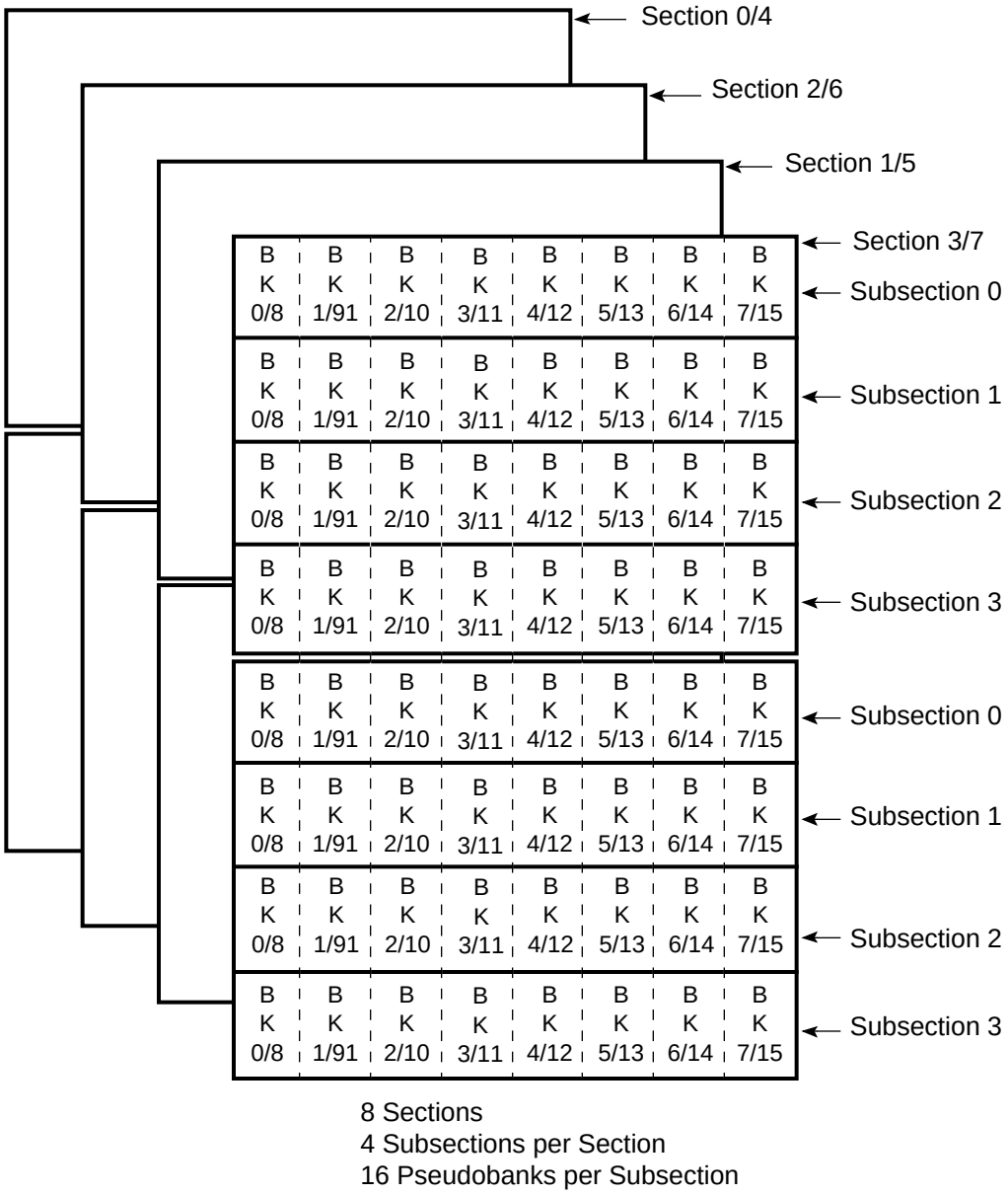


Figure 11. Memory Module Layout - 8 x 8 Half Populated

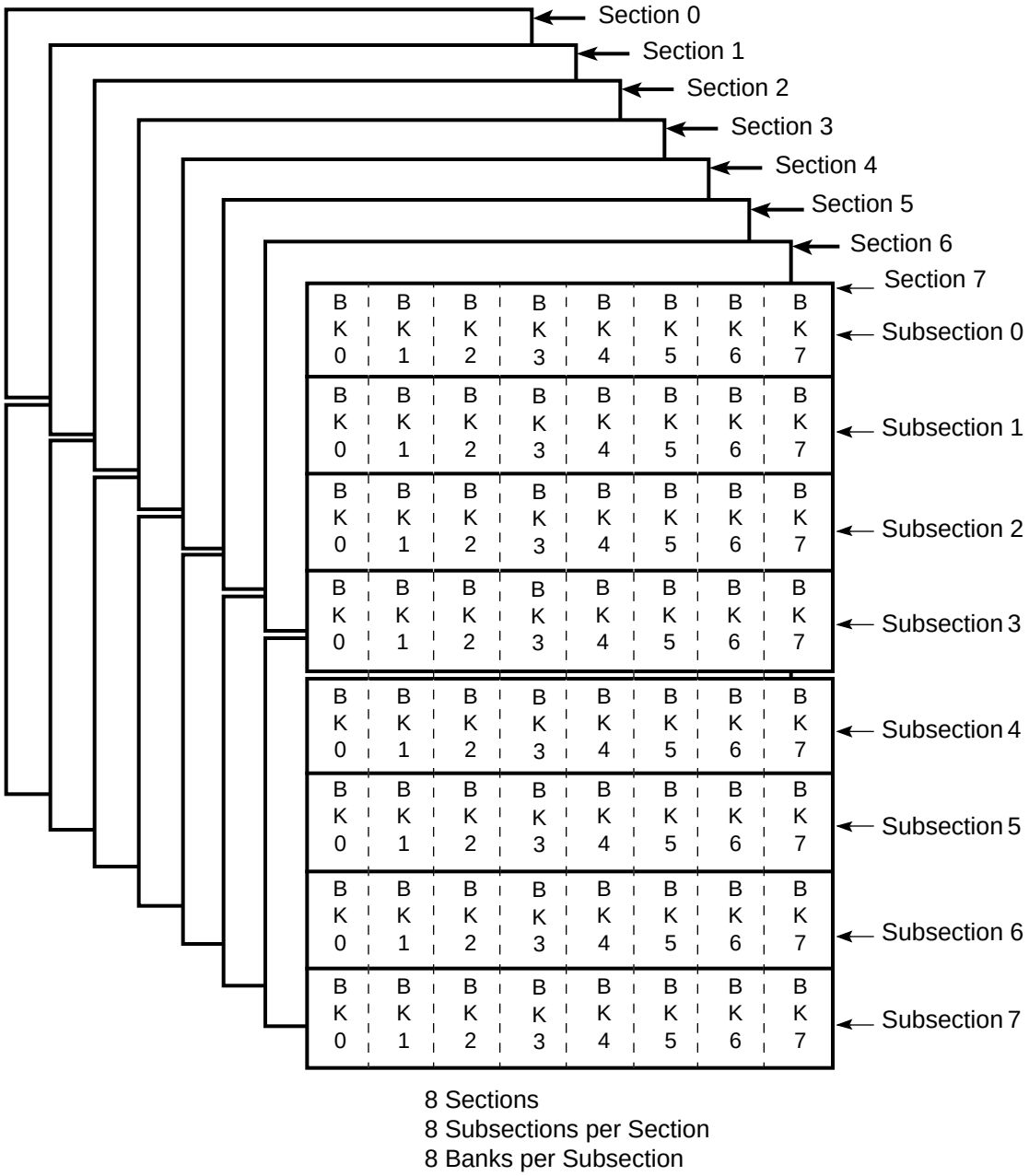
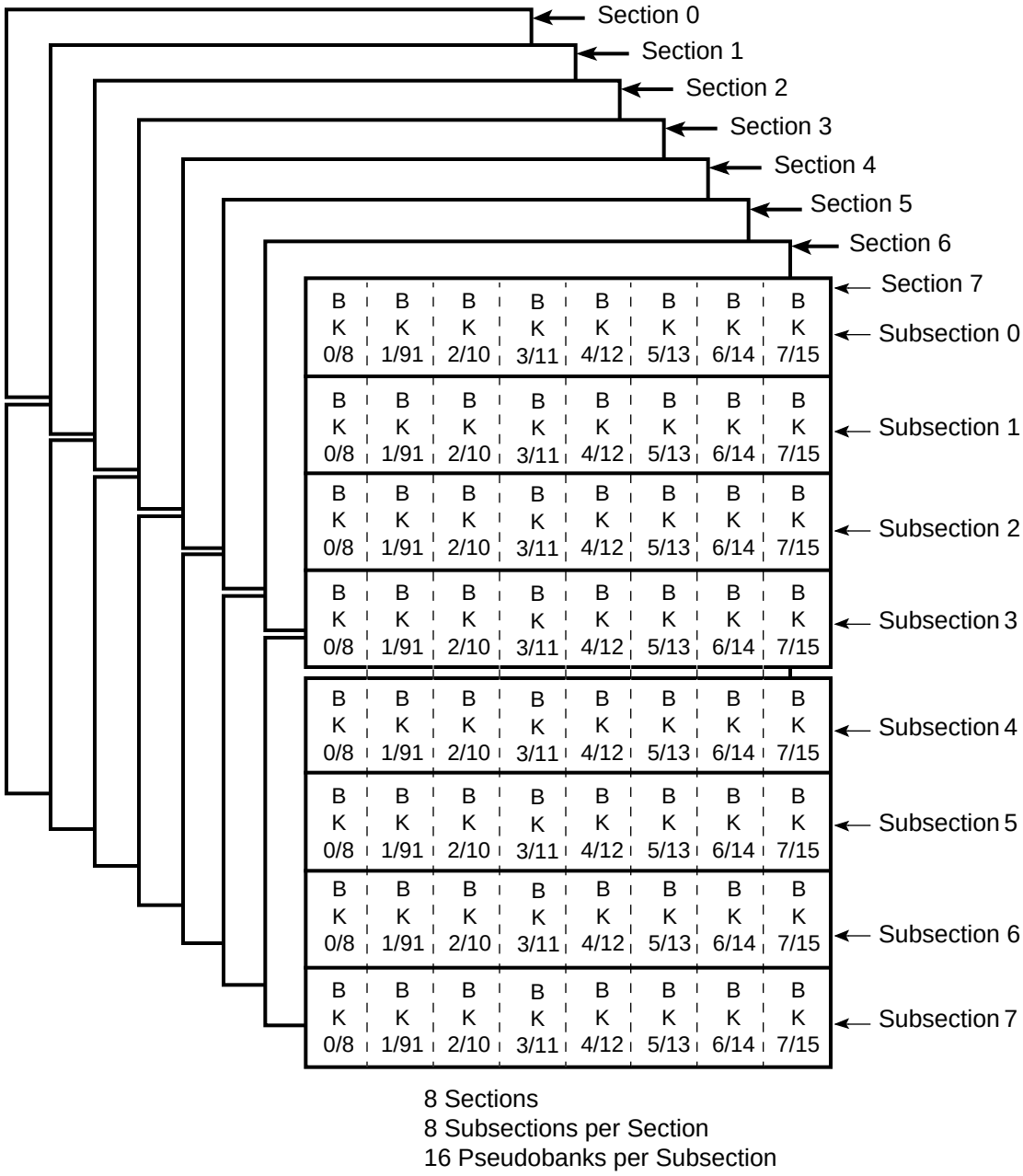


Figure 12. Memory Module Layout - 8 x 8 Fully Populated



Refer to Table 6 for the memory configurations that are available for 2 X 2, 4 X 4, and 8 X 8 systems.

Table 6. Memory Configurations

Backplane Type	Number of Processor Modules	Number of Memory Modules	Memory Chip Sizes 4-Mbit DRAMs		Memory Chip Sizes 16-Mbit DRAMs		Memory Sizes 64-Mbit DRAMs	
			MEM16	MEM32	MEM64	MEM128	MEM256	MEM512
2 X 2	1 or 2	2	32 MW 256 MB	64 MW 512 MB	128 MW 1,024 MB	256 MW 2,048 MB	512 MW 4,096 MB	1,024 MW 8,192 MB
4 X 4	1, 2, 3, or 4	4	64 MW 512 MB	128 MW 1,025 MB	256 MW 2,048 MB	512 MW 4,096 MB	1,024 MW 8,192 MB	2,048 MW 16,384 MB
8 X 8	4, 5, 6, or 7	8	128 MW 1,024 MB	256 MW 2,048 MB	512 MW 4,096 MB	1,024 MW 8,192 MB	2,048 MW 16,384 MB	4,096 MW 32,568 MB

Memory Module Construction

A memory module is approximately 16 in. X 19.5 in. X 0.16 in. and comprises 26 metal layers: 12 signal layers with buried vias, 12 power/ground layers, and 2 surface layers. Each memory module contains 64 dual in-line memory (DIM) modules. Each DIM module, which is a 1 in. X 6 in. daughter card, contains ten DRAM chips in a fully populated system. A half-populated memory system contains five DRAM chips. A 132-pin header mounts the DIM modules on-edge to the memory module. The DIM modules are soldered to the memory module and are not field replaceable. Refer to Figure 14 for an illustration of the DIM module.

The aluminum framework that surrounds the memory module provides mechanical support during engagement and alignment of the modules in the backplane. This framework also provides support for all the components on the module. Aluminum covers on the front and back of the module direct airflow.

Each memory module has its own power supply in the form of a logic power module (LPM) that is located on the back side of the printed circuit board. The LPM receives 48-Vdc power and converts it to the closely regulated 5.0-Vdc power for the memory DRAMs and 3.3-Vdc power for the ASICs. The power module provides n + 1 capabilities for improved reliability. The power module also provides several local control functions such as voltage margining, local inhibit, and air-stream overtemperature sensing. Refer to the *Power, Cooling, and Control* document for more information on power.

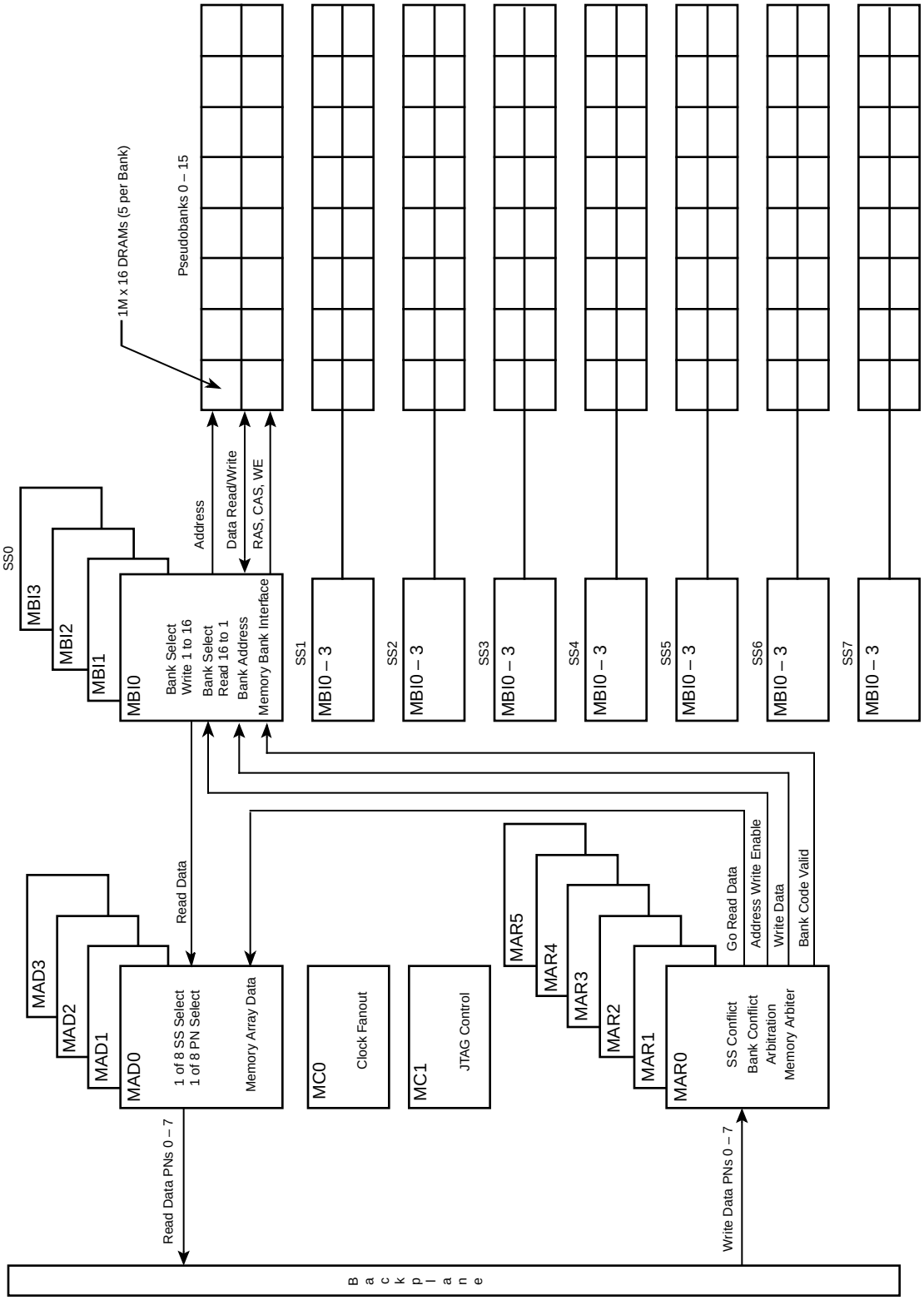


Figure 13. Memory Module Block Diagram

Memory ASIC Descriptions

A memory module contains the following four types of ASICs, for a total of 44 ASICs per memory module: 6 memory arbiter (MAR) ASICs, 4 memory array data (MAD) ASICs, 32 memory bank interface (MBI) ASICs, 1 maintenance and clock fanout (MC0) ASIC, and 1 maintenance and clock for JTAG control (MC1) ASIC. Refer to Figure 14 for a diagram of the memory module ASIC layout.

The MAR ASICs

- Buffer incoming requests
- Queue requests to their target subsections
- Track bank busy and bank access times
- Drive references at appropriate times
- Control the MAD ASICs

The MAD ASICs

- Steer the 72-bit read data
- Control the order in which data is sent to the VB ASICs on the processor modules

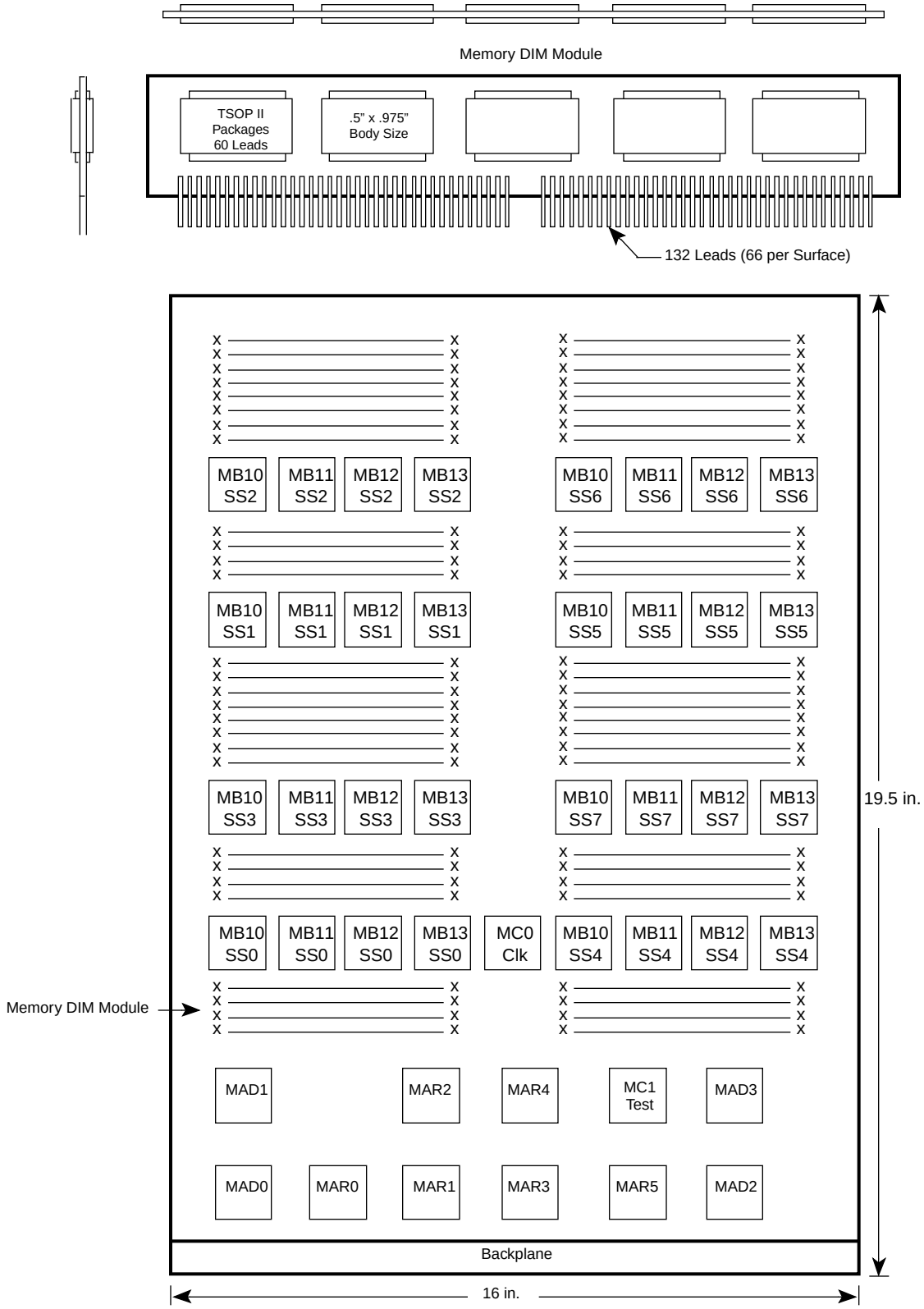
The MBI ASICs

- Send write data and addresses from the MAR ASICs to the DRAMs
- Send read data addresses from the MAR ASICs to the DRAMs and steer the read data from the DRAMs to the MAD ASICs

The MC0 ASIC fans out the clock signal to all ASICs on the memory module.

The MC1 ASIC provides the JTAG, boundary scan, stop clock, and reset maintenance functions for the memory module.

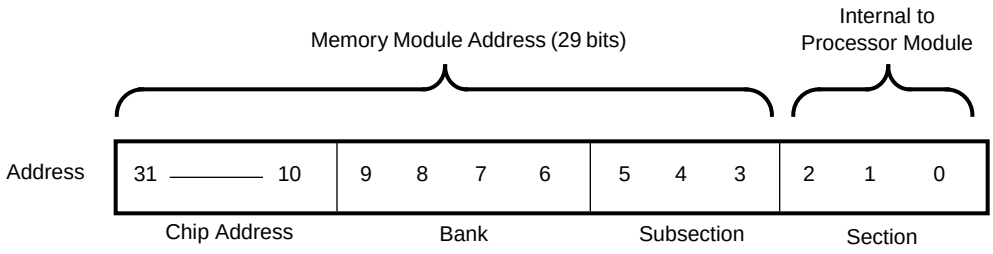
Figure 14. Memory Module ASIC Layout



Memory Addressing

There are 32 absolute address bits, but only 29 address lines are carried onto the backplane. Absolute address bits 0 through 2 select the memory section; bits 3 through 5 select the subsection; bits 6 through 9 select the bank; and bits 10 through 31 select the chip address. Figure 15 shows the four memory address fields. Table 7 lists memory addressing information for the 2 X 2, 4 X 4, and 8 X 8 systems. Refer again to Figure 9 through Figure 12 for the memory sections on each memory module.

Figure 15. Memory Address Bits



NOTE: Subsection bits 4 and 5 are not used in a 2 X 2 system.

Table 7. Memory Addressing

Backplane Configuration	Memory Sections	Memory Module
2 X 2	0, 2, 4, 6 1, 3, 5, 7	0 1
4 X 4	0, 4 1, 5 2, 6 3, 7	0 1 2 3
8 X 8	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7

The memory addressing scheme uses a rotating priority through the 8 sections of memory with 2-section spacing between the slots. Each slot has highest priority to 1 memory section each CP. This is called the slot's *natural* priority. A natural slot priority that is not in use may be *borrowed* by another slot. A

borrowing priority exists for the three *non-natural* slots. I/O operations are unslotted and may use any available slot. I/O priority is configured from lowest to highest priority, depending on system requirements. All read and write requests to memory are handled on a slot basis. Ports A and B of CPU 0 share slot 0; ports A and B of CPU 1 share slot 1; ports A and B of CPU 2 share slot 2; ports A and B of CPU 3 share slot 3; and so on. Refer to the *CRAY J90 Series System Programmer Reference*, Cray Research publication number CSM-0301-0B0, for more information on memory addressing.

Memory Paths

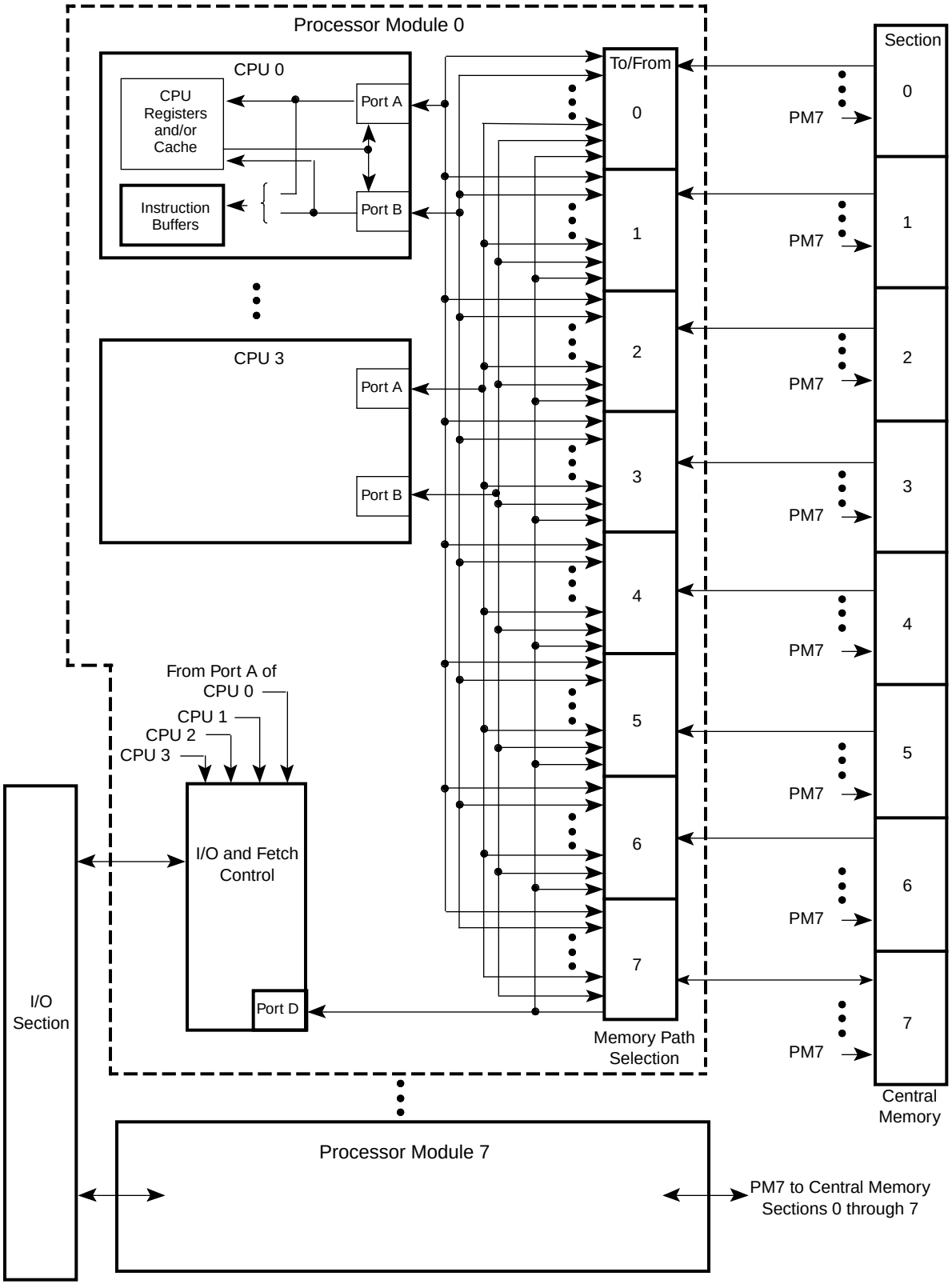
Each processor module has an independent path into each memory section. Figure 16 shows the central memory architecture. The 4 CPUs and the I/O on a processor module share these eight paths. Each of the eight paths is capable of sending one request to memory per CP and receiving read data from memory at the same rate. Each CPU can have overlapping references in different sections without restrictions. Simultaneous references from a processor module to the same section are not permitted because only one physical path into each memory section exists for requests and write data.

A rotating priority scheme gives the 4 CPUs on a processor module equal access to each section of memory. All memory references for a CPU are sent to each memory section in the proper order. However, no order is guaranteed for memory references between the various CPUs in the system. Each memory section buffers the requests as required by bank busy signals and requires activity from all CPUs in the system. A memory section guarantees order for a request from a single CPU but not for requests between CPUs.

Memory Ports

Each CPU has two ports: port A and port B. Each port has a specific function that is defined jointly by the read mode bits and the port bits in the exchange package. Ports A and B are both read and write ports, but they allow only one write operation to be active at a time. However, both ports may process read operations simultaneously. Also, a read operation may occur on one port while a write operation occurs on the other port, if the bidirectional mode bit (BDM) is set in the exchange package. A third port, port D, handles I/O and instruction fetch operations.

Figure 16. CPU Central Memory Architecture



System Clock

The CRAY J90se series system uses two types of maintenance and clock (MC0) ASICs: the MC0 ASIC that is located on the master clock board, and the MC0´ ASIC that is located on all processor and memory modules in the system. The crystal oscillator on the master clock board distributes the 25-MHz (40-ns) system reference clock to the MC0 ASIC, which fans out the clock signal to the MC0´ ASIC on all modules (CPU, memory, and clock). The MC0´ ASIC that is located on each module fans out the system reference clock to all ASICs on the module. The phase lock loop (PLL) on each ASIC multiplies the 25-MHz system reference clock by 4 (4 X 25 MHz = 100 MHz or 10 ns). The PLL, which is called the *on-chip clock*, is fanned out to all flip-flops within the ASIC.

NOTE: The PC+ ASIC operates at twice the speed of the PC ASIC in CRAY J90 series systems. A special clocking scheme and additional interface logic synchronize the data that enters and leaves the PC+ ASIC.

The crystal oscillator on the master clock board generates the nominal clock period. When clock margins are run, an external clock is brought onto the master clock board through an external port. The onboard crystal oscillator is turned off when this external port is used.

The master clock board also contains the MC1 ASIC, which provides stop clock functions. The master clock board connects to the CC ASIC on the paddle card of the channel adapter board by means of the console bus.

Boundary Scan

The boundary scan IEEE standard was developed by the Joint Test Action Group (JTAG). This standard is a collection of design rules that are applied principally at the integrated circuit level. The primary benefit of this standard is its capability to transform difficult PCB-testing problems into well-structured problems that diagnostic software can analyze.

The boundary scan feature is a function of the MC1 (internal maintenance) ASIC. To perform a *boundary scan* means to test the input and output latches on the processor module ASICs and the interconnections between the latches. The boundary scan test also checks the connectivity of the backplane. Boundary scan provides test access to device pins by associating a serial shift register element, or scan cell, with each signal pin. The boundary scan cells link to form a shift register chain around the device boundary. These scan cells are then used to observe and control the device pins.

NOTE: The boundary scan feature does not test the ASIC interconnections on a CRAY J90se channel adapter board when it is installed on a processor module. The boundary scan test is performed on channel adapter boards in a manufacturing test environment.