CRAY SSD-T90: Input/Output Operations

Record of Revision

February 1997

Original printing.

Introduction

When an I/O failure occurs you may need to replace a field replaceable unit (FRU) or map out a bad component. An FRU for an I/O failure may be a processing element module, a ribbon cable, or a drop cable.

This document describes the components of an I/O controller, how the I/O controllers are cabled to the GigaRing channel, and the types of I/O transfers. This document also lists the diagnostic tests you can use to troubleshoot an I/O failure. Specifically, this document answers the following questions:

1. What is the I/O transfer rate of the CRAY SSD-T90 device?
2. What type of I/O errors can occur?
3. What is I/O configuration for the CRAY SSD-T90 device?
4. Can a bad I/O controller be disabled?

Terms

You will need to know how the following terms are used in this document to fully understand the material discussed in this document:

Node - A node contains a processing element (PE) and a network router.

Processing element (PE) - A PE contains a microprocessor, local memory, and support circuitry.

Microprocessor - The microprocessor is a reduced instruction set computer (RISC) 64-bit microprocessor developed by Digital Equipment Corporation.

Support circuitry - The support circuitry is a component of a PE that extends the control and addressing functions of the microprocessor in the PE.

E registers - E registers are latency-hiding registers in the support circuitry that are the source and destination for all global data transfers.

Local memory - With respect to the microprocessor in a processing element, local memory is memory that is physically located in the same PE as the microprocessor.

Packet - All request and response information is transferred through the network in the form of a packet. A packet contains a header and a body.

Acknowledge (Ack) packet - The Ack packet informs the PE that initiated a message that the destination PE accepted the message.

No acknowledge (Nack) packet - The Nack packet informs the PE that initiated a message that the destination PE did not accept the message. The Nack packet contains the same information as the original message.

Components of the I/O Controller

In the CRAY SSD-T90 device, I/O controllers transfer data between the PEs and the external devices (refer to Figure 1). Each CRAY SSD-T90 printed circuit board (PCB) contains one I/O controller. This I/O controller is connected to the network routers of the four nodes that reside on the PCB. Even though the I/O controller has direct connections to the four nodes on its PCB, the I/O controller can transfer data to any of the nodes within the CRAY SSD-T90 device.

The I/O controller uses the packet formats shown in Table 1 to send requests and responses to the network router. Each flit contains 96 bits of data; however, only 72 bits are used. Each flit divides into 24-bit minor flits, which are sent between the I/O controller and the network router. This 24-bit channel is referred to as a time-multiplexed channel. To transfer the 24 bits between the I/O controller and the network router, 4 bits are sent approximately every 2.2 ns (system clock speed [13.3 ns] divided by 6).

Each I/O controller consists of an I option and a GigaRing option. The I option performs messaging, master direct memory access (DMA) transfers, slave DMA transfers, boundary scan functions, and construct-a-command functions (refer to Figure 2). Information about these I/O transfers and the boundary scan and construct-a-command functions is provided later in this document. The GigaRing option receives packets from the GigaRing channel. The GigaRing option checks these input packets for parity errors and cyclic redundancy code (CRC) errors and buffers the input packets in the receive buffers; the packets remain in the buffer until the GigaRing option can send the packets to the I option (refer to Figure 3). The GigaRing option also buffers the output packets in the input virtual channel buffer before transferring the data to the positive or negative active send buffer. The GigaRing option also generates parity and CRC for each packet before the packet is sent to the GigaRing channel.

Figure 1. I/O Controller Block Diagram

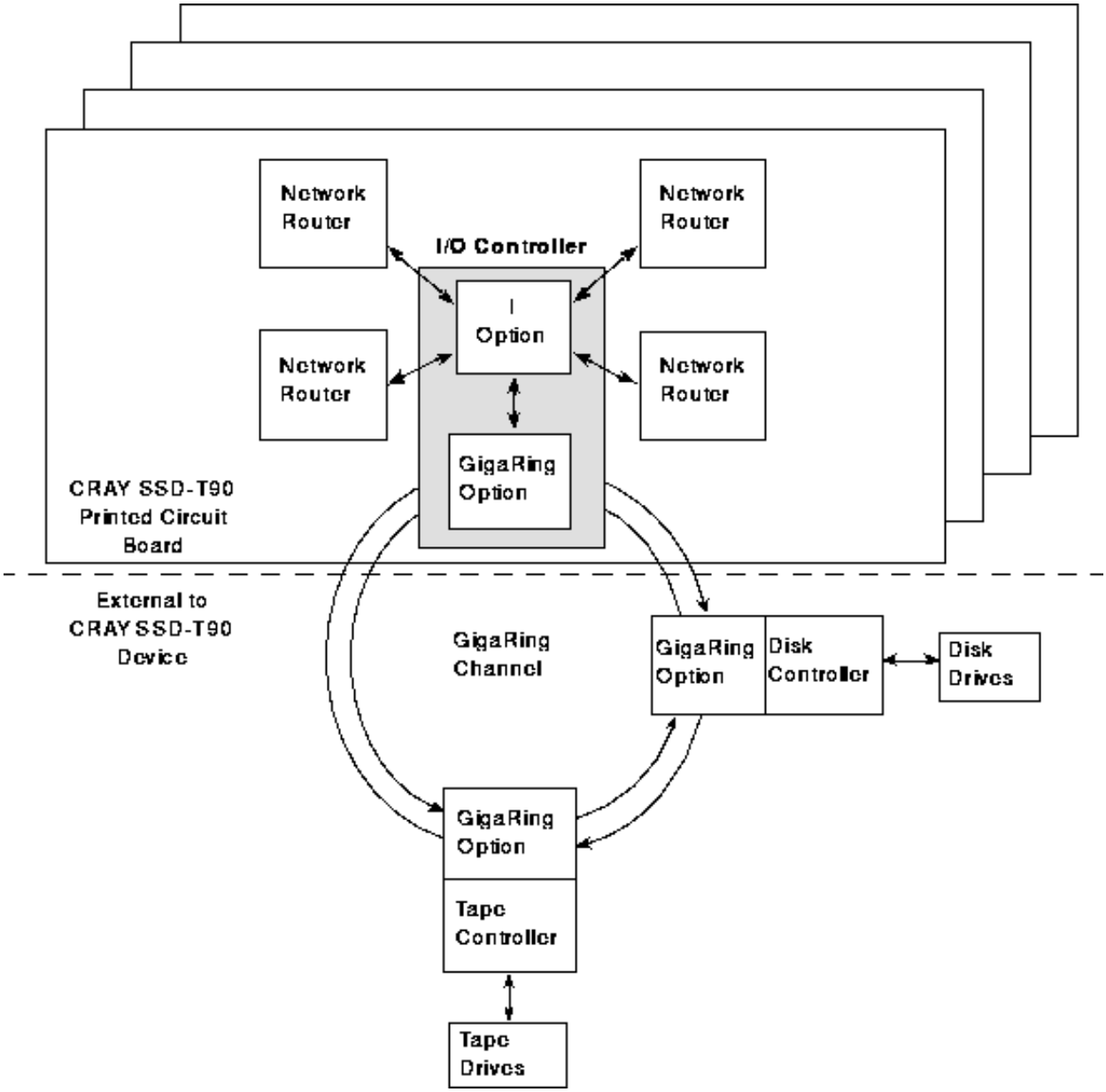
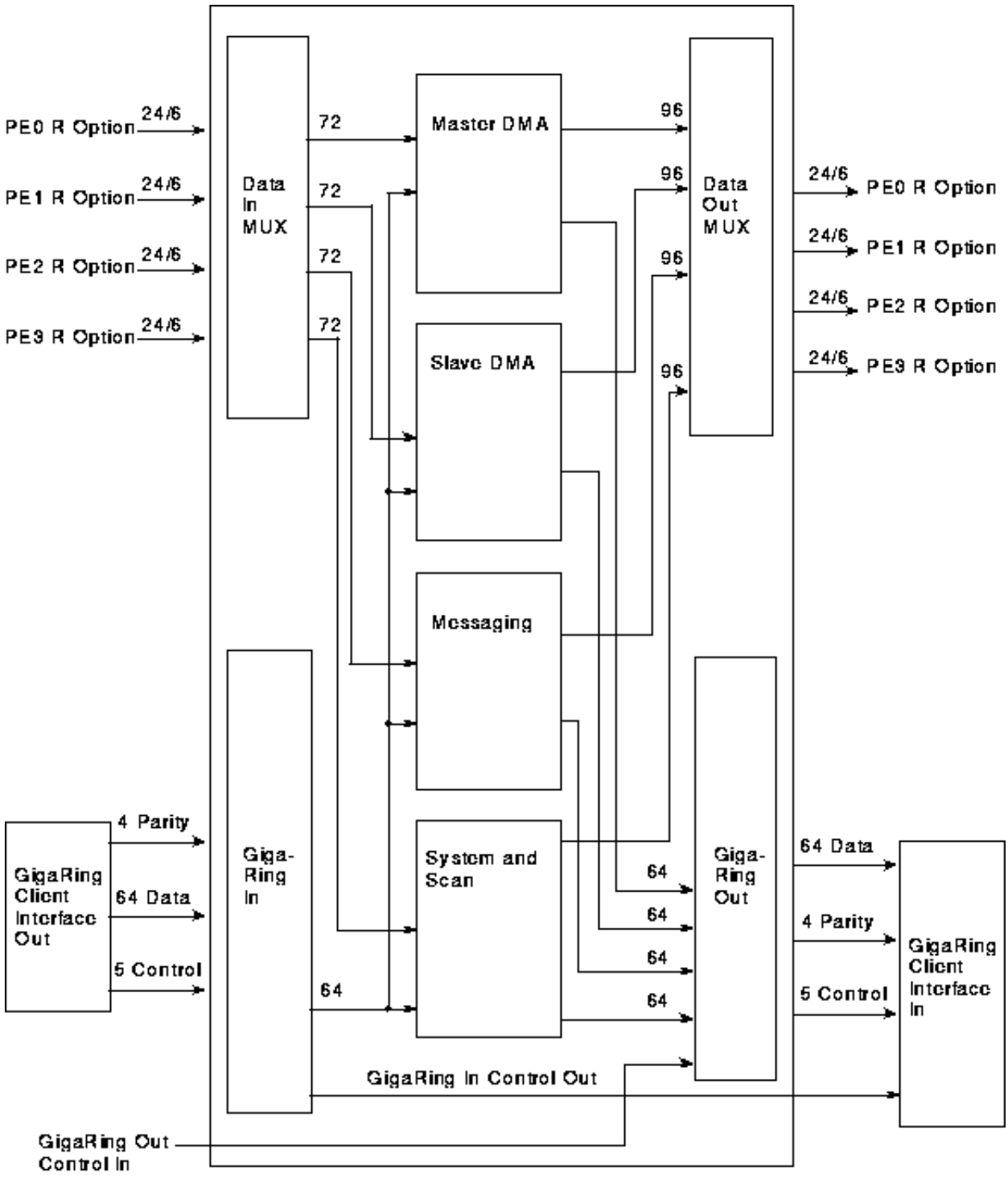


Figure 2. I Option Block Diagram



NOTE: The 24/6 notational convention indicates that the I option receives or transfers 24 bits of data to the R option in one system clock period and it does so using six 4-bit transfers.

Figure 3. GigaRing Option Block Diagram

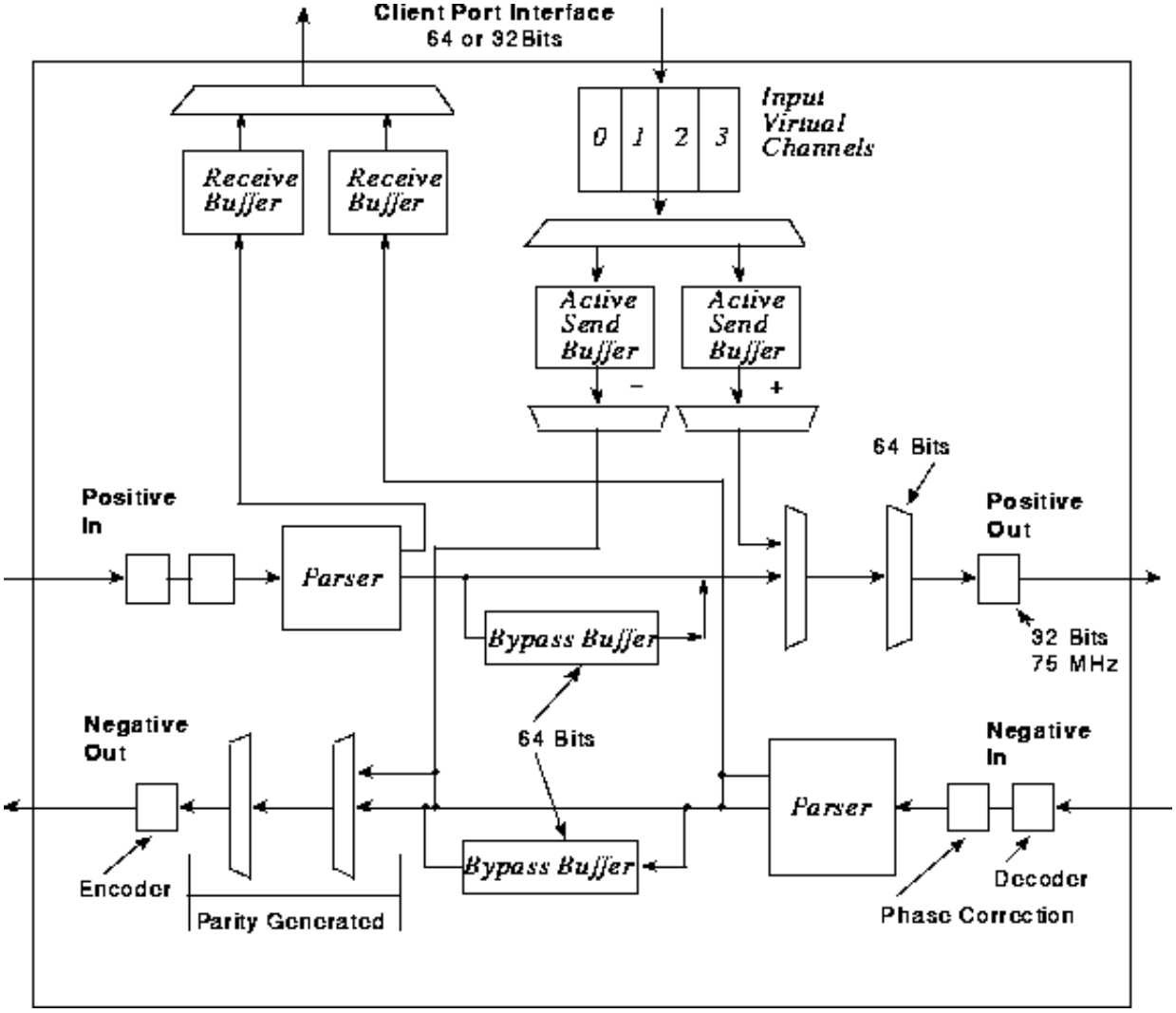


Table 1. I Option to R option Packet Formats

Command	Flit 0	Flit 1	Flit 2	Flit 3	Flit 4	Flit 5	Flit 6	Flit 7	Flit 8
DGET, GET8, GETV8, SGET Request, PUT8 Response, SEND ACK	Head	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
DPUT, SPUT, PUT8 Request	Head	Body	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SEND, PUTV8 Request	Head	Wd 0	Wd 1	Wd 2	Wd 3	Wd 4	Wd 5	Wd 6	Wd 7
GET8, SPUT Response, SEND NACK	Head	Body	N/A	N/A	N/A	N/A	N/A	N/A	N/A

I/O Configurations

A CRAY SSD-T90 module has one GigaRing option. Each set of bulkhead connections consists of four

separate connections to the GigaRing channel: input for the positive ring, output for the positive ring, input for the negative ring, and output for the negative ring. The CRAY SSD-T90 modules connect to the GigaRing bulkhead on the Y side of the chassis (bottom of chassis) (refer to Figure 4).

Types of I/O Transfers

The CRAY SSD-T90 device uses the GigaRing channel to communicate with external devices. The GigaRing channel defines an I/O protocol that is used by all GigaRing clients. This protocol supports two types of transfers: peer-to-peer messaging and direct memory access (DMA).

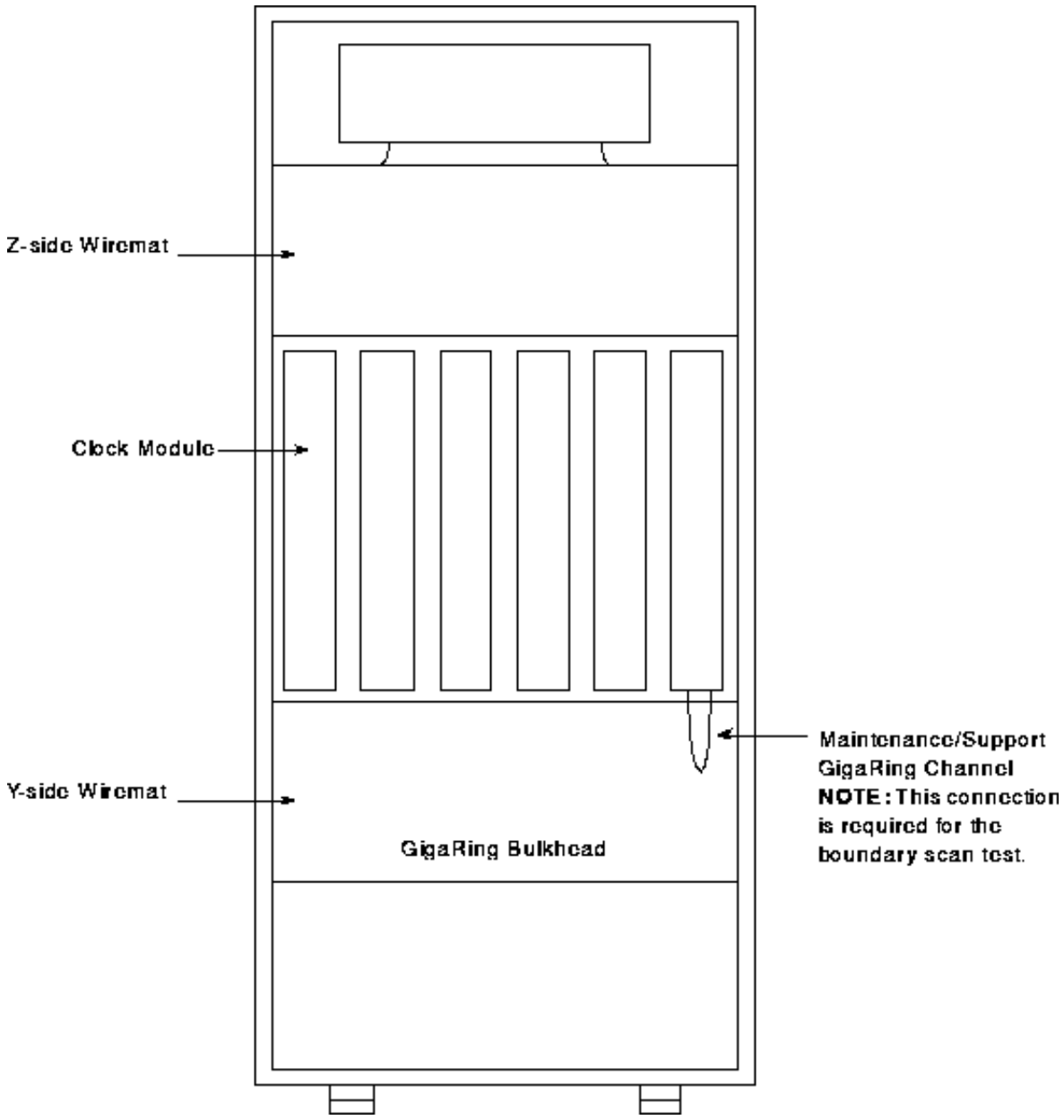
NOTE: The CRAY SSD-T90 device is a slave device to the CRAY T90 system; therefore, when the operating system is running in the CRAY T90 system, the CRAY SSD-T90 device will not generate peer-to-peer messages or master DMA transfers. The CRAY SSD-T90 device only generates peer-to-peer messages and master DMA transfers when you run offline diagnostics.

Peer-to-peer Message Transfers

Peer-to-peer messaging enables the GigaRing clients to communicate with each other without negotiating transfer rates. Peer-to-peer messaging does not require a response from the destination client.

Peer-to-peer messages are transferred in message packets (MsgPkts). The data payload of a MsgPkt can range from 0 to 32 64-bit words. When a client's message is larger than 32 words, the client must transfer the message in multiple MsgPkts or use a DMA transfer.

Figure 4. I/O Configuration for a CRAY SSD-T90 device



Sending a Peer-to-peer Message

The following text describes the steps in which the CRAY SSD-T90 device sends a peer-to-peer message to another device on the GigaRing channel. The step numbers correspond to the numbers in Figure 5.

1. When a PE in the CRAY SSD-T90 device needs to communicate with an external device, the PE initiates a peer-to-peer message transfer. To do this, the PE writes a MsgPkt that contains the I/O request parameters into its local memory. Next, the PE writes information that indicates the location of the MsgPkt in local memory and the length of the MsgPkt into a block of eight E registers.

2. The PE also issues a SEND command. This SEND command transfers the message information from the eight contiguous E registers to an I/O controller.

3. When the I option of the I/O controller receives the SEND command, the I option stores the message information in the I_MSG_OUT_Q register. This outgoing message queue register buffers a maximum of eight messages. The I option processes the messages in the order that it receives them. When the message

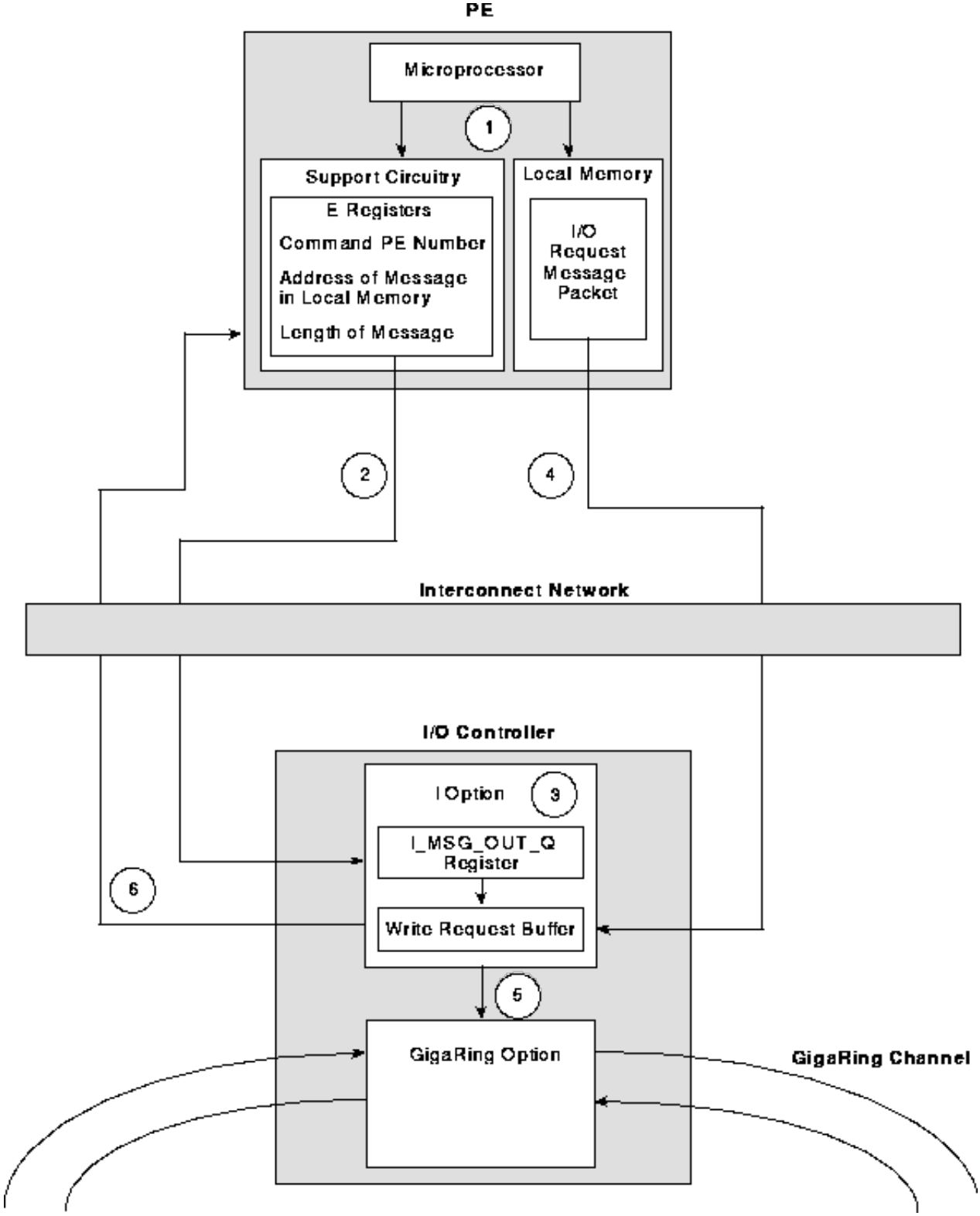
queue is full, the I option rejects the SEND command and returns a NACK packet to the PE.

4. When the I option processes the message, the I option reads the message out of the message queue and places it into its control registers. The I option reads the MsgPkt from the local memory of the PE and stores it into the write request buffer.

5. When the message passes through the write request buffer, the I option sends the message to the GigaRing option. The GigaRing option sends the MsgPkt out onto the GigaRing channel.

6. Once the message leaves the I option, the I option sends an acknowledge back to the PE.

Figure 5. Peer-to-peer Messaging



Receiving a Peer-to-peer Message

The following text describes the steps in which the CRAY SSD-T90 device receives a peer-to-peer message from another device on the GigaRing channel. The step numbers correspond to the numbers in Figure 6.

1. When an external device initiates a peer-to-peer message transfer with the CRAY SSD-T90 device, the external device sends a message to an I/O controller in the CRAY SSD-T90 device. The target identification number of the message specifies the I/O controller as a node on the GigaRing channel.

2. When the message contains a data payload, the I option of the I/O controller writes the data payload portion of the message to a location in the CRAY SSD-T90 memory. To do this, the I option retrieves an incoming message address from the I_MSG_IN_AD register. The incoming message address indicates the destination PE and the global virtual address of local memory. The I option uses the address and the data payload to generate PUT or PUTV commands. The PUT/PUTV commands instruct the destination PE to write the data payload portion of the message into its local memory. (Data is aligned on 64-word boundaries.)

   The I option increments bits <37 : 9> (global virtual address) of the I_MSG_IN_AD register and decrements bits <63 : 52> (limit) of the I_MSG_IN_AD register after it stores a message payload in local memory. The limit bits indicate the number of slots that are available to store message payloads. When the limit decrements to 0, the I option cannot write additional message payloads to memory.

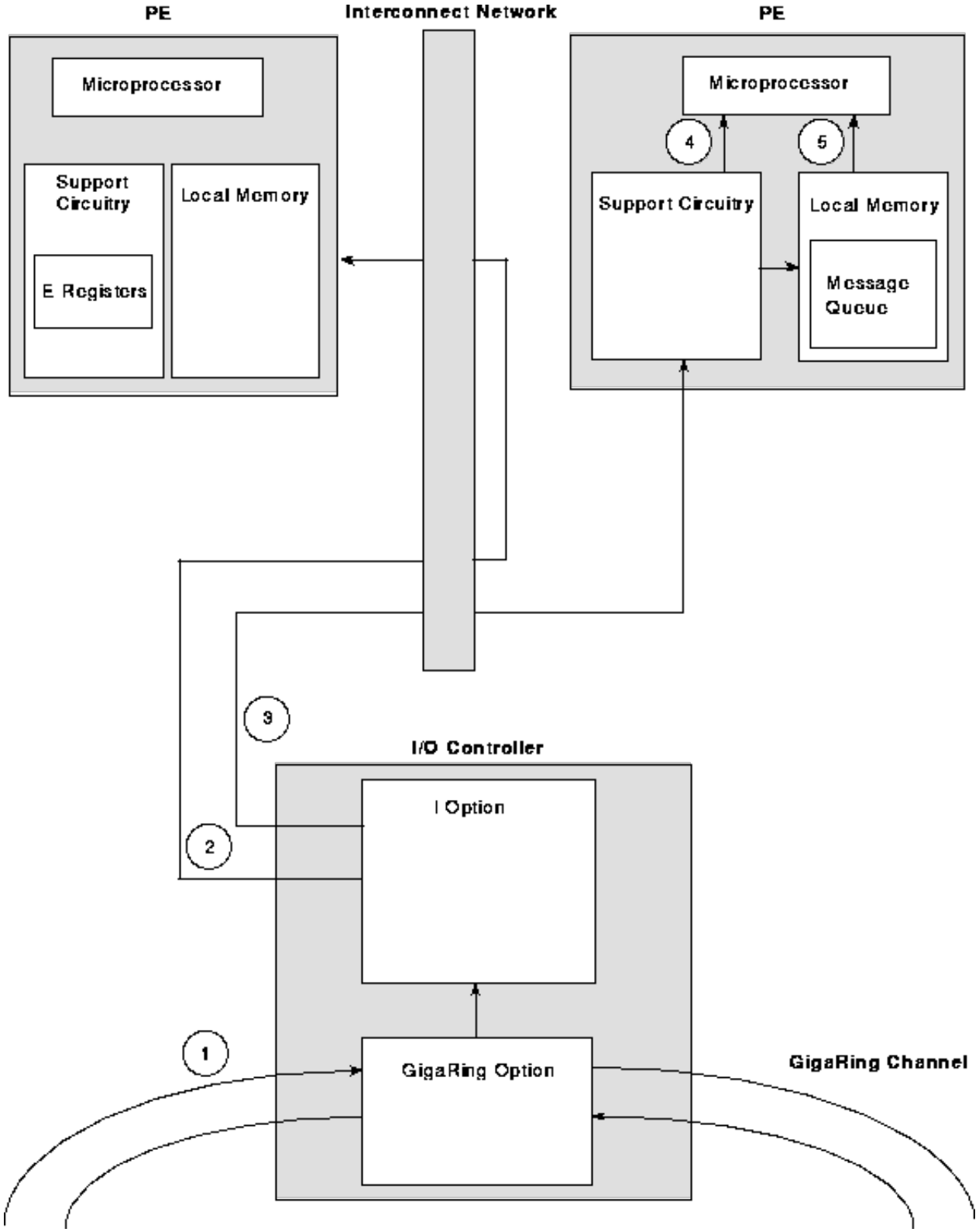
3. After the data payload portion of the message is written to memory, the I option generates a SEND command that transfers the header portion of the message to the message queue of the PE that the software designates. The software designates the PE by using the I_MSG_IN_MQCW register. This register also indicates the message queue address and controls incoming messages.

   When the controlling PE receives the SEND command, the support circuitry writes the message header into the specified location in the message queue.

4. When the interrupt is enabled, the support circuitry interrupts the microprocessor.

5. The microprocessor reads the message header from the message queue and performs the appropriate action for the data payload.

Figure 6. Receiving a Message



Direct Memory Access Transfers

A DMA transfer enables a GigaRing client to read from or write to the memory of another GigaRing client. The client that initiates the transfer is the master of the transfer and the other client is the slave of the transfer.

Unlike peer-to-peer messaging, the DMA transfer establishes a transfer rate between the master and slave device and acknowledges when the transfer is complete. The transfer rate specifies how many outstanding

requests the master device can have on the GigaRing channel. The transfer rate is established by a read block initiate (RdBlkInit) packet or a write block initiate (WrBlkInit) packet. The completion of the transfer is acknowledged by a read block done (RdBlkDone) packet or a write block done (WrBlkDone) packet.

For example, to establish the transfer rate, the master client of a DMA write transfer sends a WrBlkInit packet to the slave client. The slave client returns a write block initiate response (WrBlkInitResp) packet to the master client. This packet contains an 8-bit transfer rate value and a slave address. The master client uses the transfer rate value to determine how many outstanding write packets it can send to the slave client. The slave address indicates the starting address of the read operation or the write operation.

When the operating system is running in the CRAY T90 system, the CRAY SSD-T90 device is the slave of the DMA transfer (microprocessor is not used). When you are running offline diagnostic tests, the CRAY SSD-T90 device can be the master of a DMA transfer or the slave. When the CRAY SSD-T90 device is the master of a DMA transfer, the CRAY SSD-T90 device initiates the transfer.

Master DMA Write Transfer

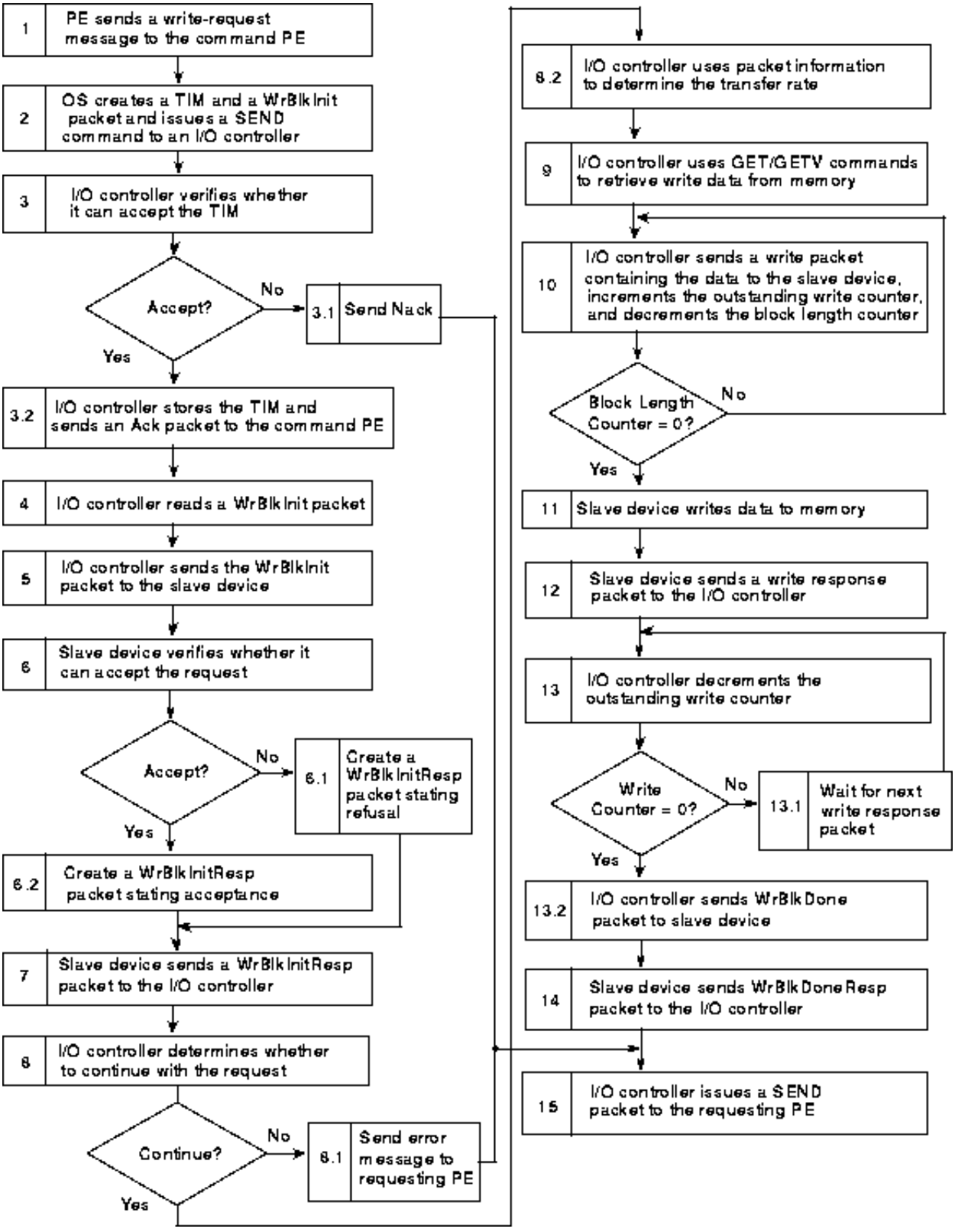
The CRAY SSD-T90 device becomes the master of a DMA write transfer when a PE needs to transfer data to an external device. The following text describes the steps in which the CRAY SSD-T90 device performs a master DMA write transfer. The step numbers correspond to the numbers in Figure 7.

1. The PE that needs to transfer data to an external device sends a message to a command PE, which is the PE that communicates with t3ems.

2. The command PE creates a WrBlkInit packet and a transfer initiation message (TIM); the command PE stores the WrBlkInit packet in local memory and the TIM in the E registers.

   To create a TIM, the command PE issues STORE commands that load a block of 8 contiguous E registers with information about the setup of the DMA transfer (command, stride, mask, block length, base address, and starting address). Once the E registers are loaded, the command PE issues a SEND command. The SEND command signals the support circuitry to transfer the information from the 8 E registers to a 64-byte slot in the I_MDMA_WR_Q register of an I/O controller.

   Figure 7. Master DMA Write Transfer



3. When the I option of the I/O controller receives the SEND command, the I option verifies whether there is room for the TIM in the I_MDMA_WR_Q register. The I option handles one write DMA transfer and buffers two messages.

    1. When no room is available in the register, the I option returns a no acknowledge (Nack) packet to the command PE.

2. When room is available in the register, the I option accepts the TIM, stores it in the I_MDMA_WR_Q register, and returns an Acknowledge (Ack) packet to the command PE.

4. The I option uses the information from the TIM to read the WrBlkInit packet from global memory. The address from the TIM specifies the location of the remote base address and remote mask.

5. After the I option reads the WrBlkInit packet, the I option sends the WrBlkInit packet to the GigaRing option. The GigaRing option sends this packet to the slave device via the GigaRing channel.

6. When the slave device receives the WrBlkInit packet, the I/O controller of this device verifies whether the device can accept the request.

    1. When the device cannot accept the request, the I/O controller generates a WrBlkInitResp packet that states the refusal of the request.

    2. When the device can accept the request, the I/O controller generates a WrBlkInitResp packet that contains the transfer rate value and a new slave base address.

7. The slave device sends the WrBlkInitResp packet to the I/O controller.

8. The I option uses the information from the WrBlkInitResp packet to determine whether to continue with the request.

    1. When the WrBlkInitResp packet indicates an error or that the slave device refused the transfer, the I option issues a SEND packet to the message queue of the requesting PE. (The PE and message queue address are specified in the TIM.) This packet indicates that the I/O controller issued the error message. The I option also releases any reservations that apply to this write request and checks for any new write DMA requests.

    2. When the slave device can accept the request, the I option uses the information from the WrBlkInitResp packet to determine the slave base address and how often it can initiate write packets.

9. Next, the I option generates GET or GETV commands to retrieve the write data from system memory. The location of the data is encoded in the information from the TIM. The I option sends this information through a centrifuge to identify the PE number and address offset for the GET/GETV command. The I option sends the GET/GETV command to the PE via the interconnect network. The PE retrieves the data from memory and sends the data to the I/O controller.

10. After the I option of the I/O controller receives a block of write data, the I option creates a write packet that contains the data and the slave base address. The I option sends the write packet to the slave device, increments an outstanding write counter, and decrements a block length counter. The block length counter indicates how many blocks of data the CRAY SSD-T90 device will write to the slave device.

    1. When the block length counter is not equal to 0 and the number of outstanding writes does not

exceed the limit that was set by the transfer rate, the I option continues to send blocks of data to the slave device.

2. When the block length counter equals 0, the I option stops sending blocks of data to the slave device (all data has been sent).

11. The slave device uses the slave base address from the write packet to generate the necessary commands to write the data to its memory.

12. The slave device creates a write response packet and sends it to the I/O controller in the CRAY SSD-T90 device.

13. After the I option of the I/O controller receives the write response packet from the slave device, the I option decrements the outstanding write counter. When the block length decrements to 0, the I option uses this counter to determine when the write block transfer is complete.

    1. When block length is 0 and the outstanding write counter is not equal to 0, the I option waits for the next write response packet.

    2. When the outstanding write counter decrements to 0, the I option generates a WrBlkDone packet and sends this packet to the GigaRing option. The GigaRing option sends the packet to the slave device via the GigaRing channel.

14. After the slave device receives the WrBlkDone packet, the slave device acknowledges the completion of the request by creating a WrBlkDoneResp packet. The slave device sends the packet to the I/O controller in the CRAY SSD-T90 device.

15. After receiving the WrBlkDoneResp packet, the I/O controller issues a SEND packet to the requesting PE. This SEND packet indicates that the I/O request is complete. The I option also releases any reservations that apply to this write request and checks for any new write DMA requests.

Master DMA Read Transfer

The CRAY SSD-T90 device becomes the master of a DMA read transfer when a PE needs data from an external device. The following text describes the steps in which the CRAY SSD-T90 device performs a master DMA read transfer. The step numbers correspond to the numbers in Figure 8.

1. The PE that needs data sends a message to a command PE.

2. The command PE creates a TIM and a RdBlkInit packet. To create a TIM, the command PE issues STORE commands that load 8 contiguous E registers with information about the setup of the DMA transfer (command, stride, mask, block length, base address, and starting address). Once the E registers are loaded, the command PE issues a SEND command. The SEND command signals the support circuitry to transfer the information from the E registers to a 64-byte slot in the message queue of an I/O controller. The command PE stores the RdBlkInit packet in memory.

3. When the I option of the I/O controller receives the SEND command, the I option verifies whether there is room for the TIM in the I_MDMA_RD_Q register. This register can store one or two messages while the I option services a different message.

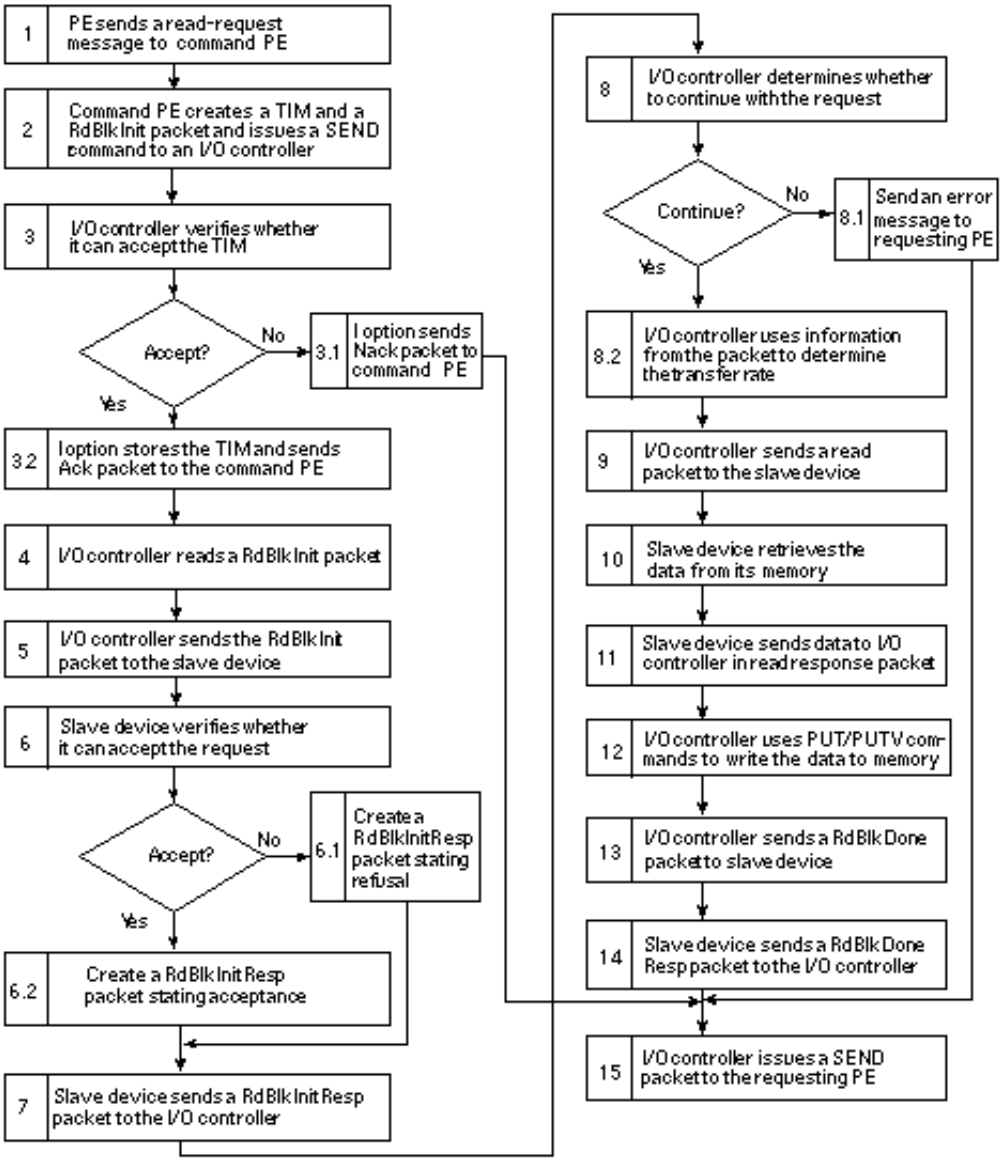
    1. When no room is available in the register, the I option returns a Nack packet to the command PE.

    2. When room is available in the register, the I option accepts the TIM , stores it in the I_MDMA_RD_Q register, and returns an Ack packet to the command PE.

4. The I option uses the information from the TIM to read the RdBlkInit packet from global memory. The address from the TIM specifies the location of the remote base address and the remote mask.

5. After the I option reads the RdBlkInit packet, the I option sends the RdBlkInit packet to the GigaRing option. The GigaRing option sends this packet to the slave device via the GigaRing channel.

Figure 8. Master DMA Read Transfer



6. When the slave device receives the RdBlkInit packet, the client of this device verifies whether the device can accept the request.

    1. When the device cannot accept the request, the client generates a RdBlkInitResp packet that states the refusal of the request.

    2. When the device can accept the request, the client generates a RdBlkInitResp packet that contains the transfer rate value and a new slave base address.

7. The slave device sends the RdBlkInitResp packet to the I/O controller of the CRAY SSD-T90 device.

8. The I option uses the information from the RdBlkInitResp packet to determine whether to continue with the request.

    1. When the slave device refuses the request, the I option issues a SEND packet that contains the GigaRing error information to the message queue of the requesting PE. (The PE and message queue addresses are specified in the TIM.) This packet indicates that the I/O controller issued the error

message. The I option also releases any reservations that apply to this read request and checks for any new DMA read requests.

2. When the slave device can accept the request, the I option uses the information from the RdBlkInitResp packet to determine the transfer rate.

9. Next, the I option creates a read packet that contains the slave device ID, the read command, the slave base address, and the source address. The slave device will return the source address in the read response packet.

   NOTE: Bits <23 : 0> of the I_MDMA_RD_LEN register indicate the total number of 64-bit words that transfer during a master DMA read transfer. For each block of 32 words, the I option sends a read packet to the slave device. For each read packet, the I option increments the slave base address by 32 and sends the source address through the centrifuge to determine the local CRAY SSD-T90 address.

   The I option sends the read packet to the GigaRing option. The GigaRing option sends this new packet to the slave device via the GigaRing channel.

10. The slave device uses the slave base address from the read packet to address its memory and reads the specified amount of data.

11. The slave device places this data into a read response packet along with the source address. The slave device assembles this packet according to the GigaRing protocol and sends the packet to the CRAY SSD-T90 device.

12. After the I option of the I/O controller receives the read response packet from the slave device, the I option generates PUT/PUTV commands, which write the data to memory.

13. After the I option writes all of the read data to memory (block length is 0 and the I option received the last read response packet), the I option creates a RdBlkDone packet and sends this packet to the GigaRing option. The GigaRing option sends this packet to the slave device via the GigaRing channel.

14. After the slave device receives the RdBlkDone packet, the slave device acknowledges the completion of the request by sending a RdBlkDoneResp packet to the CRAY SSD-T90 device.

15. After the I/O controller receives the RdBlkDoneResp packet, the I/O controller issues a SEND packet to the requesting PE. This SEND packet indicates that the I/O request is complete. The I option also releases any reservations that pertain to this read request and checks for any new DMA read requests.

Slave DMA Write Transfer

The CRAY SSD-T90 device becomes the slave of a DMA write transfer when an external device needs to transfer data to the CRAY SSD-T90 device. The following text describes the steps that the CRAY SSD-T90 device performs during a slave DMA write transfer. The step numbers correspond to the numbers in Figure 9.

1. The external device initiates the DMA transfer by sending a WrBlkInit packet to an I/O controller in the CRAY SSD-T90 device. This WrBlkInit packet contains stride, mask, starting index, base address, and block-length information.

2. When the I option of the I/O controller receives the WrBlkInit packet, the I option determines whether it can accept the request. Each I option can handle up to 32 different slave operations (read or write).

    1. When the I option cannot accept the request, the I/O controller creates a WrBlkInitResp packet that states the refusal.

    2. When the I/O controller can accept the request, the I/O controller stores the information from the WrBlkInit packet in the control registers for 1 of 32 buffers (also referred to as windows). The I option also creates a WrBlkInitResp packet that contains an 8-bit transfer rate and the buffer address.

3. The I option sends the WrBlkInitResp packet to the GigaRing option. The GigaRing option sends this packet to the master device via the GigaRing channel.

4. After receiving the WrBlkInitResp packet from the CRAY SSD-T90 device, the master device creates write packets that contain the write data and the buffer address.

5. The master device sends the write packets to the CRAY SSD-T90 device.

6. Using the buffer address from the write packet, the I option of the CRAY SSD-T90 I/O controller retrieves a stride, a mask, a block length, a base address, and a starting index from the buffer. The I option uses this information to determine the PE number and global virtual address.

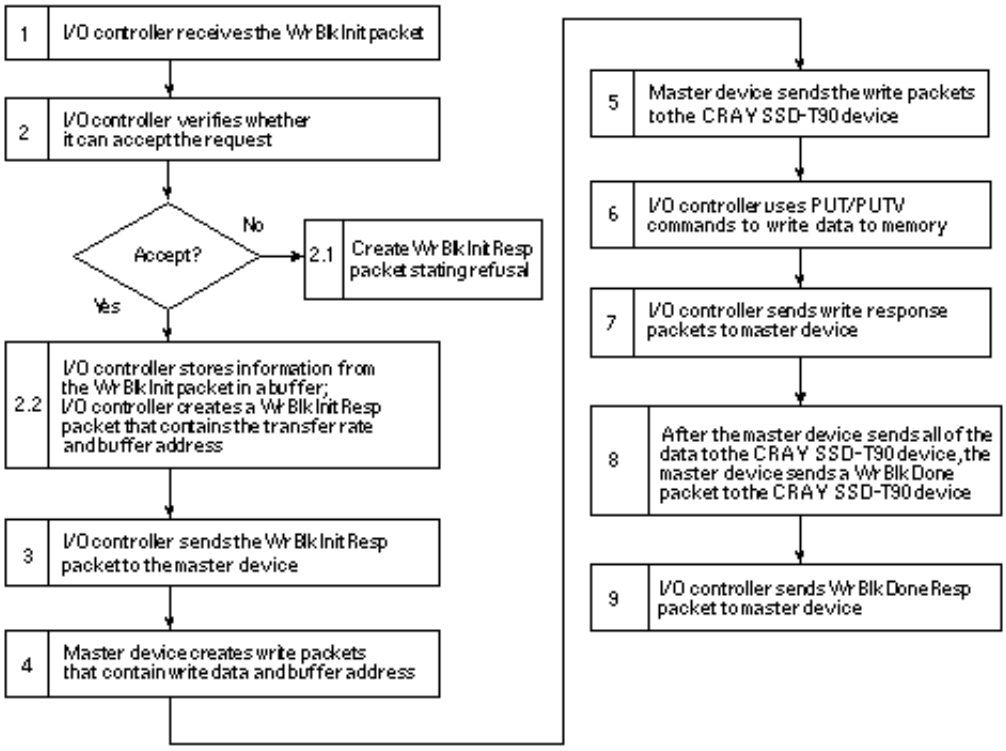
   The I option also generates PUT/PUTV commands that it sends to the destination PE via the interconnect network. The I option continues to generate PUT/PUTV commands until all of the data is written to the memory of the CRAY SSD-T90 device.

7. After each PUT/PUTV command completes, the destination PE responds with a PUT/PUTV response to the I option. After receiving the PUT/PUTV response, the I option sends a write response packet to the master device.

8. After the master device sends all of the write data to the CRAY SSD-T90 device and the CRAY SSD-T90 device responds with write response packets, the master device sends a WrBlkDone packet to the CRAY SSD-T90 device.

9. When the I option of the I/O controller receives this packet and all of the outstanding PUT/PUTV commands are complete, the I option generates a WrBlkDoneResp packet and sends this packet to the GigaRing option. The GigaRing option sends the WrBlkDoneResp packet to the master device via the GigaRing channel.

Figure 9. Slave DMA Write Transfer



Slave DMA Read Transfer

The CRAY SSD-T90 device becomes the slave of a DMA read transfer when an external device needs data from the CRAY SSD-T90 device. The following text describes the steps that the CRAY SSD-T90 device performs during a slave DMA read transfer. The step numbers correspond to the numbers in Figure 10.

1. The external device initiates the DMA transfer by sending a RdBlkInit packet to an I/O controller in the CRAY SSD-T90 device. This RdBlkInit packet contains stride, mask, starting index, base address, and block-length information.

2. When the I option of the I/O controller receives the RdBlkInit packet, the I option determines whether it can accept the request. Each I option can handle up to 32 different operations (read or write).

    1. When the I option cannot accept the request, the I/O controller creates a RdBlkInitResp packet that states the refusal.

    2. When the I/O controller can accept the request, the I/O controller stores the information from the RdBlkInit packet in the control registers for 1 of 32 buffers (also referred to a windows). The I/O controller also creates a RdBlkInitResp packet that contains the transfer rate and the return buffer address.

3. The I option sends the RdBlkInitResp packet to the GigaRing option. The GigaRing option sends this packet to the master device via the GigaRing channel.

4. After receiving the RdBlkInitResp packet from the CRAY SSD-T90 device, the master device creates read packets that contain the buffer address.

5. The master device sends the read packets to the CRAY SSD-T90 device.

6. Using the buffer address from the read packet, the I option of the CRAY SSD-T90 I/O controller retrieves a stride, a mask, a block length, a base address, and a starting index from the control registers of the buffer. The I option uses this information to determine the PE number and the global virtual address. To determine the location of the read data, the I option adds the starting index to the stride.

   The I option also generates GET/GETV commands to read information from memory. The I option continues to generate GET/GETV commands until the block of data is read from the memory of the CRAY SSD-T90 device.

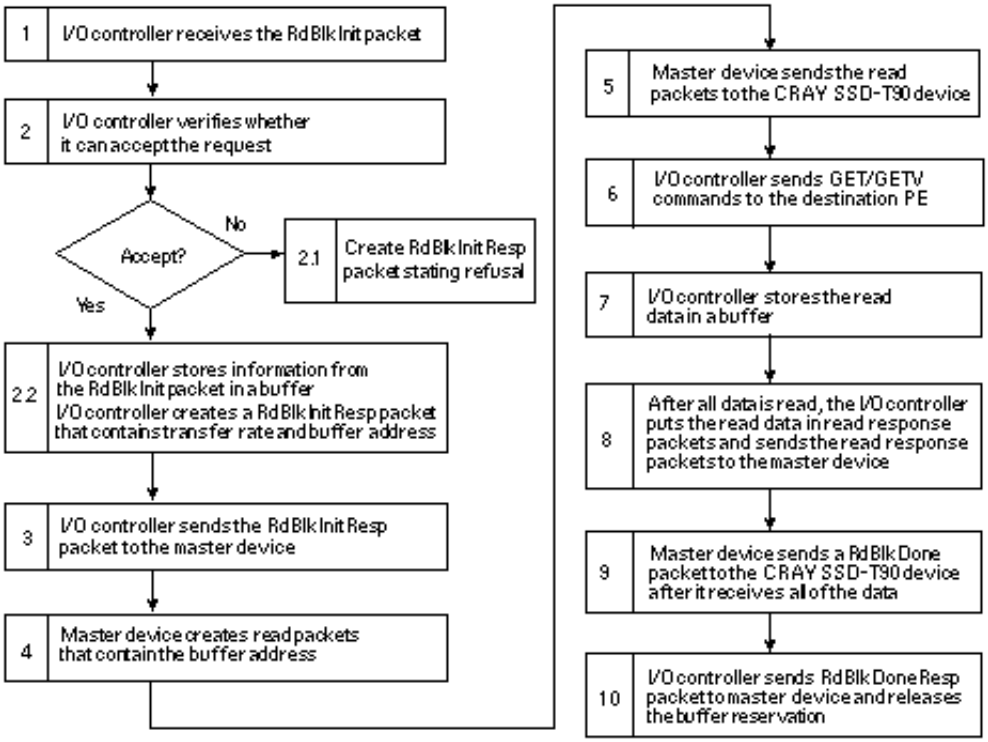
7. When the I option receives the read data, the I option stores the read data in the buffer.

8. Once the I option receives all of the read data, the I option places the data in read response packets and sends the packets to the GigaRing option. The GigaRing option sends the read response packets to the master device via the GigaRing channel.

9. The master device sends a RdBlkDone packet after it receives all of the read data from the CRAY SSD-T90 device.

10. When the I option of the I/O controller receives this packet, the I option releases the reservation on the buffer and generates a RdBlkDoneResp packet. The I option sends the RdBlkDoneResp packet to the GigaRing option. The GigaRing option sends the RdBlkDoneResp packet to the master device via the GigaRing channel.

Figure 10. Slave DMA Read Transfer



Boundary Scan and Construct-a-command Functions

The I/O controllers can perform boundary scan functions and construct-a-command functions. For the boundary scan function, the I/O controller initiates the boundary scan test and buffers the results. For the construct-a-command function, the I/O controller receives data from an external device and sends this data to a network router.

Boundary Scan

The boundary scan function enables an external device to test the connections between the options and the modules in the CRAY SSD-T90 device.

An external device (usually the system workstation) initiates the boundary scan test by sending a write scan request packet to the scan master I/O controller. A CRAY SSD-T90 device has only one scan master I/O controller. The scan master is located in slot 1 of chassis 0.

The write scan request packet contains a command and data. The command instructs the I/O controller to use the data from the packet as test vectors for the boundary scan test. The I/O controller sends the data from the packet to the boundary scan logic one bit at a time. The data bits are propagated through the boundary scan logic, where they test for continuity between the options and the modules. When the I/O controller sends the last bit to the boundary scan logic, the I/O controller sends a write-scan response packet to the external device.

The scan master I/O controller buffers the results of the boundary scan test. To view the results, the external

device sends a read-scan request packet to the scan master I/O controller. After receiving the request, the scan master I/O controller retrieves the test results from the buffer, places the results in a read-scan response packet, and sends the read-scan response packet to the external device.

Construct-a-command

The construct-a-command function enables an external device to write data to, or read data from, the network routers. For example, a programmer can use the construct-a-command function to create SPUT commands and SGET commands that initialize the network routers.

An external device initiates the construct-a-command function by sending a construct-a-command request to an I/O controller in the CRAY SSD-T90 device (this can be any I/O controller in the CRAY SSD-T90 device).

After the I/O controller receives the construct-a-command request, the I/O controller determines whether the request is a read or a write. When the request is a write, the I/O controller sends the data from the packet to the specified network router without checking the validity of the packet. When the request is a read, the I/O controller retrieves the data from the specified network router, places the data in a construct-a-command response packet, and sends the packet to the external device.

Time-multiplexed Channels

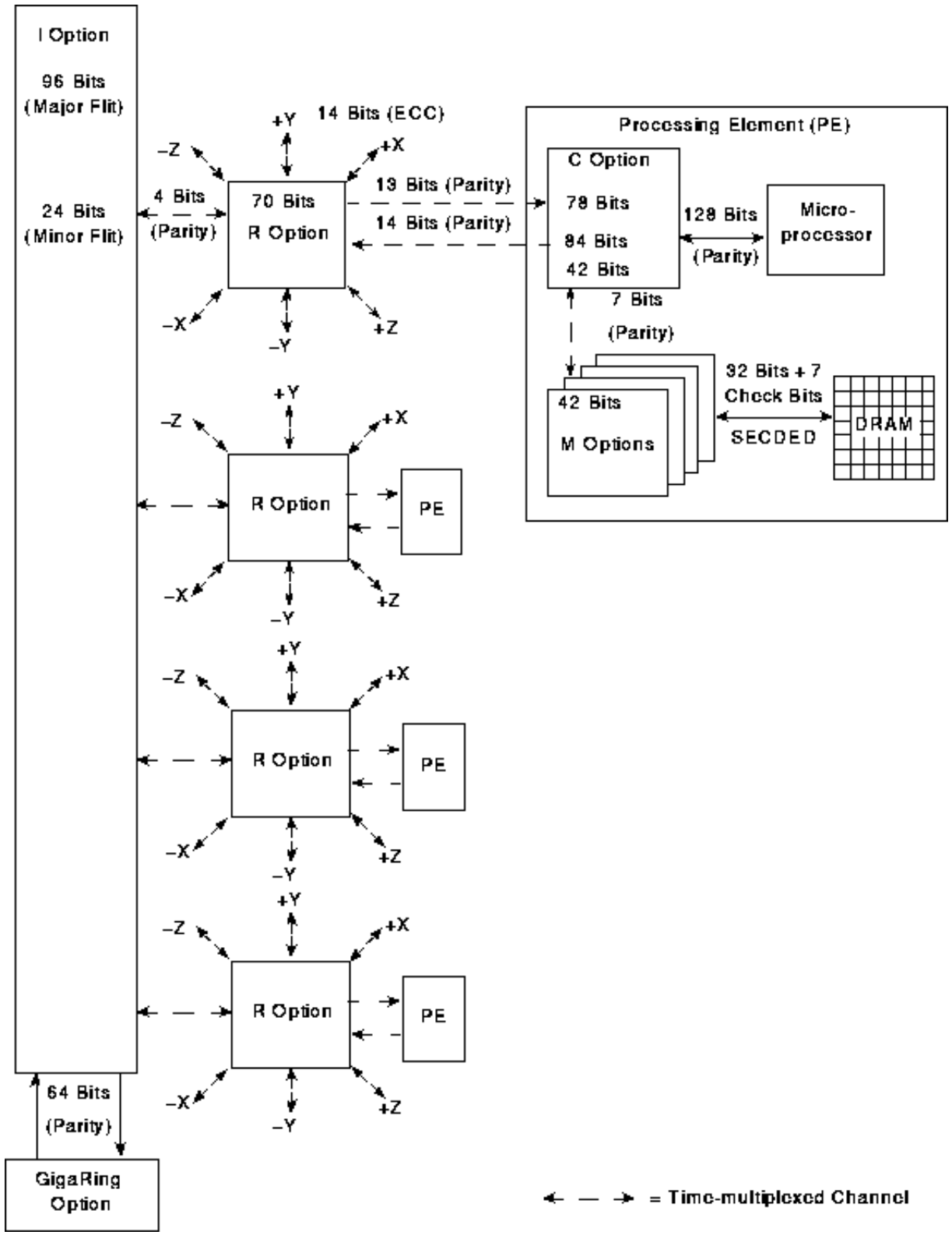
Because each logic option has a limited number of I/O pins and a large amount of data is transferred between options, the CRAY SSD-T90 device uses time-multiplexed channels. A time-multiplexed channel is a channel that makes several transfers of data within one system clock period.

For the C option, the M option, and the R option, the amount of data that transfers over a time-multiplexed channel in one system clock period (13.3 ns) is referred to as a *flit* (flow control unit). The size of a flit varies among options. For the I option, the amount of data that transfers over the time-multiplexed channel is referred to as major and minor flits (refer to Figure 11); a major flit is 96 bits and consists of four 24-bit minor flits.

A flit or minor flit is divided into *phits*. A phit, which also varies in size among options, is the amount of data that transfers across a time-multiplexed channel in one transfer. For example, a phit for the channel between the R option and the I option is 4 bits. This means that when the I option transfers a 24-bit minor flit to the R option in one system clock period, the I option sends a 4-bit phit to the R option at a rate of 2.2 ns (13.3 ns divided by 6).

The time-multiplexed channels are always active; however, when there is no data that needs to be sent between two options, the options send idle packets across the channel.

Figure 11. Time-multiplexed Channels



I/O Errors

The I option detects three types of errors: GigaRing interface errors, I-option-to-network-router channel errors, and internal I-option errors.

GigaRing Interface Errors

When the I option receives information from the GigaRing option, the I option checks the information for parity errors (refer again to Figure 11). When the I option detects a parity error, the I option logs the error in the I_ERR0 register and reports the error to the network router by setting the error bit in an idle packet. The network router passes the error information to its PE and the PE interrupts the microprocessor.

I-option-to-network-router Channel Errors

When the I option receives information from a network router, the I option checks the information for parity errors. When the I option detects a parity error on this channel, the I option logs the error in the I_ERR0 register and reports the error to the network router that sent the corrupted information by setting the error bit in an idle packet. The network router passes the error information to its PE and the PE interrupts the microprocessor.

Internal I-option Errors

The following internal I-option errors may occur:

- Input buffer parity error
- Output buffer parity error
- Input DMA FIFO parity error
- Output DMA FIFO parity error
- Message Nack limit exceeded

The parity errors occur when the I option reads data out of one of the buffers and the parity computes as even.

A Message Nack limit exceeded error occurs when the I option tries repeatedly to send a message to a PE but the PE does not accept the message. When the number of times the I option sends the message to the PE exceeds a specified limit, an error occurs.

The I option logs all of these errors in the I_ERR0 register and reports the errors to the network router by setting the error bit in an idle packet.

There are four I_ERR registers: I_ERR[3 : 0]. The I_ERR0 register contains status bits that indicate the types of errors that occurred for the I/O controller. The I_ERR1 register enables error interrupts. The I_ERR[3 : 2] registers are not used.

I_ERR0

When a bit in the I_ERR0 register is set to 1, the bit indicates that the corresponding error occurred (refer to

Table 2 and Figure 12).

Table 2. I_ERR0 Register Bit Format

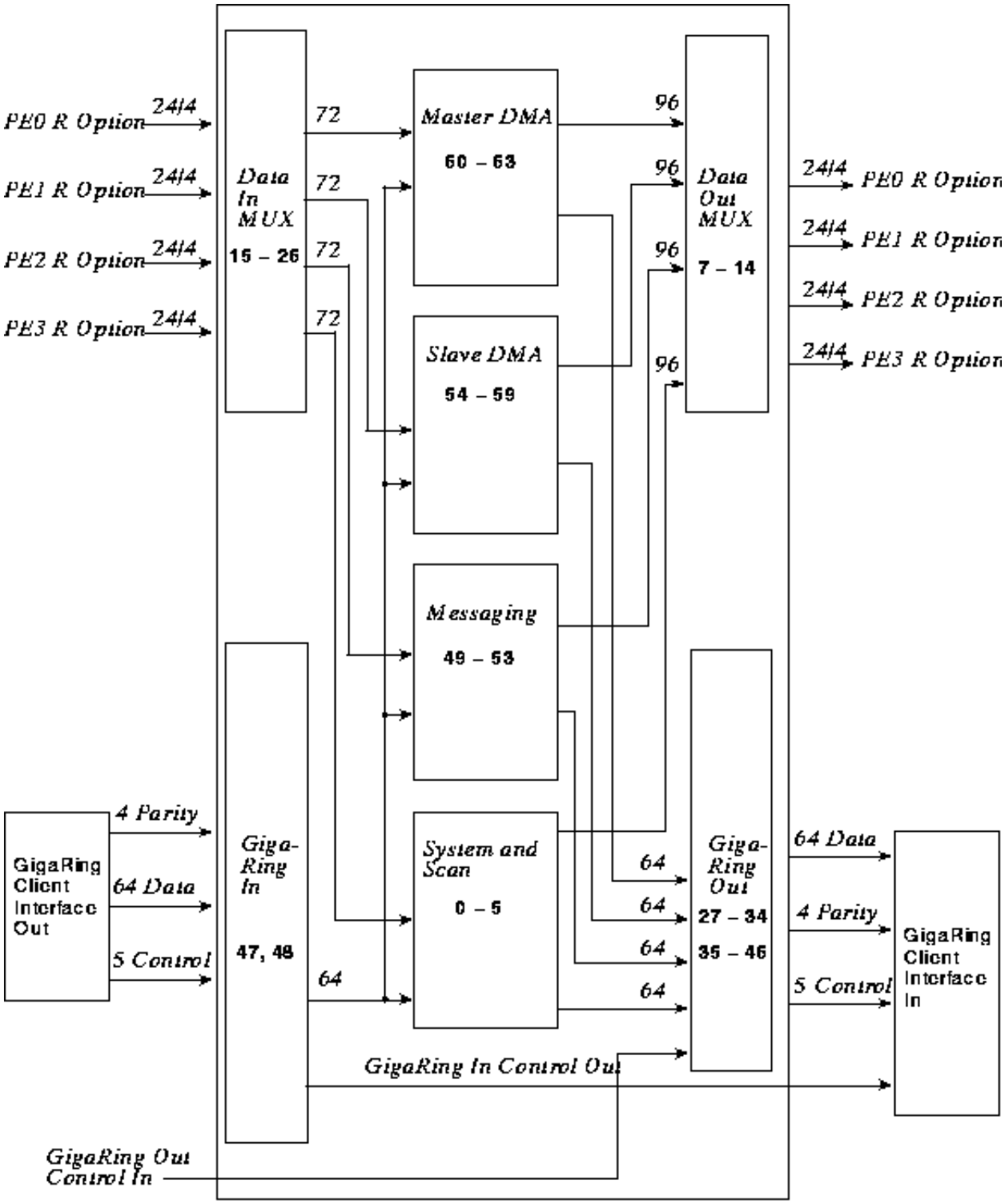
Bits	Description
<6 : 0>	When set to 1, each of the following bits indicates that an error occurred in the system and scan control. 0 = Input MUX request data payload parity error 1 = Output MUX data from static random access memory (SRAM) parity error 2 = GigaRing output data from SRAM parity error 3 = Scan data from SRAM parity error 4 = GigaRing input data parity error 5 = Input multiplexer (MUX) data parity error 6 = Not used
<14 : 7>	When set to 1, each of the following bits indicates that an error occurred in the output MUX. 7 = Port 3 overflow/underflow error 8 = Port 3 SRAM parity error 9 = Port 2 overflow/underflow error 10 = Port 2 SRAM parity error 11 = Port 1 overflow/underflow error 12 = Port 1 SRAM parity error 13 = Port 0 overflow/underflow error 14 = Port 0 SRAM parity error
	When set to 1, each of the following bits indicates that an error occurred in the input MUX. 15 = Port 0 flit ECC error 16 = Port 0 flit parity error 17 = Port 0 SRAM parity error 18 = Port 1 flit ECC error 19 = Port 1 flit parity error 20 = Port 1 SRAM parity error

<26 : 15>	21 = Port 2 flit ECC error 22 = Port 2 flit parity error 23 = Port 2 SRAM parity error 24 = Port 3 flit ECC error 25 = Port 3 flit parity error 26 = Port 3 SRAM parity error ECC = Error correction code
<34 : 27>	When set to 1, each of the following bits indicates that an error occurred in the GigaRing output channel control. 27 = Acknowledge 0 overflow error 28 = Acknowledge 1 overflow error 29 = Acknowledge 2 overflow error 30 = Acknowledge 3 overflow error 31 = Acknowledge 0 overflow error 32 = Acknowledge 1 overflow error 33 = Acknowledge 2 overflow error 34 = Acknowledge 3 overflow error
<46 : 35>	When set to 1, each of the following bits indicates that an error occurred in the GigaRing output channel control first-in first-out (FIFO) buffers. 35 = FIFO 3 overflow error 36 = FIFO 2 overflow error 37 = FIFO 1 overflow error 38 = FIFO 0 overflow error 39 = FIFO 3 underflow error 40 = FIFO 2 underflow error 41 = FIFO 1 underflow error

	42 = FIFO 0 underflow error 43 = FIFO 3 parity error 44 = FIFO 2 parity error 45 = FIFO 1 parity error 46 = FIFO 0 parity error
<48 : 47>	When set to 1, each of the following bits indicates that an error occurred in the GigaRing input channel control. 47 = SRAM parity error 48 = GigaRing channel parity error
<53 : 49>	When set to 1, each of the following bits indicates that an error occurred in the message control. 49 = Message input parity error 50 = Message input send rejected error 51 = Message response parity error 52 = Message response packet error 53 = Message request parity error
<59 : 54>	When set to 1, each of the following bits indicates that an error occurred in the slave direct memory address transfer control. 54 = Control GigaRing channel error 55 = Write initialization GigaRing channel error 56 = Write initialization register parity error 57 = Data buffer parity error 58 = Response packet error 59 = Response parity error
<63 : 60>	When set to a 1, each of the following bits indicates that an error occurred in the master direct memory address transfer control. 60 = The read control block received a SEND response with the state flags equal to SendErr, an illegal combination, or a Send Nak and the retry count exceeded the limit. 61 = The write control block received a SEND response with the state flags equal to SendErr, an illegal combination, or a Send Nak and the retry count exceeded the limit.

	62 = A parity error occurred in the request, response, or GigaRing blocks. 63 = A format error occurred in a response packet.

Figure 12. I Option Error Reporting



NOTE: The bold numbers correlate to bits 0 through 63 of the I_ERR0 register.

I_ERR1

The I_ERR1 register enables the error interrupts. When a bit of the I_ERR1 register is set to 1, the corresponding interrupt of the I_ERR0 register is enabled. When a bit of the I_ERR1 register is set to 0, the corresponding interrupt for the I_ERR0 register is disabled. Although the interrupt is disabled, the corresponding bit of the I_ERR0 register for that interrupt still indicates the state of the interrupt.

NOTE: Bit 42 of the IR_STATUS register sets when a bit in the I_ERR0 register sets and the corresponding bit in the I_ERR1 register is already set.

I_ERR[3 : 2]

These registers are not used and are read as 0's.

I_STATUS Register

The I_STATUS register contains control and status information for the I/O controller. When an I/O controller fails, software can disable the ports from the I/O controller to the network routers by setting bits <3 : 0> of the I_STATUS register to 1 (refer to Table 3).

Table 3. I_STATUS Register Bit Format

Bits	Name	Description
<3 : 0>	PORT_DISABLE	When set to 1, this bit disables the network ports. 0 = Network port to network router 00 1 = Network port to network router 01 2 = Network port to network router 10 3 = Network port to network router 11
<5 : 4>	Not Applicable	These bits are not used.
<7 : 6>	Not Applicable	These bits are not used.
<14 : 8>	BRD_REV	These bits indicate the revision level of the printed circuit board.
15	Not Applicable	These bits are not used.
16	SME	When set to 1, this bit enables the scan master.
<63 : 17>	Not Applicable	These bits are not used.

Offline Diagnostics

The offline diagnostic tests for the I/O controller are `cit` and `gnt`. The offline utility is `gru`. There are two offline diagnostic scripts that test the I/O controller: `giga_diag.pgm` and `giga_pkt.pgm`. The I/O utility script

is `giga_dump.pgm`.

cit

The client interface test (`cit`) verifies the I options with the exception of the boundary scan circuitry and the logic analyzer circuitry. This test does not test the I options on the PCBs that do not contain GigaRing options.

giga_diag.pgm

The `giga_diag.pgm` script tests the deadstart path from the SWS to the CRAY SSD-T90 device. It tests the GigaRing option, the I option, and the path between the I option and the local network routers (on the same PCB). This script requires that the GigaRing is functional.

The `giga_diag.pgm` script is a low-level test; run this test when you suspect a GigaRing problem.

giga_pkt.pgm

The `giga_pkt.pgm` script allows you to interactively generate and execute a variety of packet types. Use this script to further isolate the problems that were detected by `giga_diag.pgm`. If the `giga_diag.pgm` script runs without error, you do not need to run the `giga_pkt.pgm` script.

giga_dump.pgm

The `giga_dump.pgm` script allows you to dump the contents of the memory-mapped registers that are located in the GigaRing option, the I option, and the network routers (R options).

gnt

The GigaRing node option test (`gnt`) verifies the functionality of the GigaRing option. One of the four PEs that reside on a PCB executes the test sections. This PE targets the I option and the GigaRing option that reside in its PCB.

gru

The GigaRing utility (`gru`) resets all CRAY SSD-T90 client interfaces (I option) and GigaRing nodes (GigaRing option) except the boot node. The GigaRing utility also dumps the memory-mapped registers of the I option and the GigaRing option to memory checkpoints.