Interrupt, Error, Status, and Control Information

HMM-143-0

Interrupt Information

The support circuitry uses two microprocessor pins to send interrupt signals to the microprocessor: the interrupt request 3 pin and the system machine check interrupt pin (refer to Table 1). The following subsections provide descriptions of the functions associated with each pin.

Table 1. Microprocessor Pins that Indicate Error or Interrupt Information

Pin Name	Pin Mnemonic	Description
Interrupt request 3	Irq_h[3]	This pin indicates that the support circuitry set an interrupt in the interrupt status (IR_STATUS) register. (The irq_h[0], irq_h[1], and irq_h[2] pins are not used.)
System machine check interrupt	sys_mch_chk_irq_h	This pin indicates that the support circuitry issued a PE prod operation.

Interrupt Request 3 (irq_h[3]) Pin

When set to 1, the irq_h[3] pin indicates that the support circuitry set an interrupt bit in the IR_STATUS register and that the corresponding bit of the interrupt enable (IR_ENABLE) register was set to 1. The interrupt bits in the IR_STATUS register indicate that the following types of interrupts occurred:

- Barrier and eureka interrupts
- Messaging interrupts
- Time of day (TOD) alarm interrupt
- Atomic memory mover interrupt
- Option-specific error interrupts

Barrier and Eureka Interrupts

Bits <31 : 0> (BAR bits) of the IR_STATUS register are the barrier and eureka interrupt bits.

Message Interrupts

Bit 32 (MSG_ADDR_VALID bit) and bit 33 (MSG_MULTIPLE bit) of the IR_STATUS register are the messaging interrupt bits.

TOD Alarm Interrupt

Bit 34 (ALARM bit) of the IR_STATUS register is the Time of Day (TOD) alarm interrupt bit. The TOD alarm interrupt bit sets when the value of the TOD clock (TOD_CLOCK) register and the value of the TOD alarm (TOD_ALARM) register are equal.

The TOD_CLOCK register is a free-running, 64-bit clock-period counter (refer to Table 2). When read from, the TOD_CLOCK register provides the present TOD_CLOCK value. When written to, the TOD_CLOCK register resets to the value that was written and then continues to count.

Table 2. TOD_CLOCK Bit Format

Bits	Description
<63 : 0>	Value of the TOD counter.

The TOD_CLOCK value is specified in units of microprocessor clock (cpuclk) periods; however, the counter is incremented during each system clock (sysclk) period. The sysclk to cpuclk ratio is set using bits <2 : 0> (SYSCLK_RATIO bits) of the C_SYS_CTL register. For example, when the SYSCLK_RATIO bits are set to 000, one sysclk period equals three cpuclk periods. In this case, the support circuitry increments the TOD_CLOCK value by three every sysclk period.

The TOD_ALARM register contains a software-specified TOD_CLOCK value (refer to Table 3). When bits <63 : 3> of the value stored in the TOD_ALARM register equal the value of bits <63 : 3> of the TOD_CLOCK register, the support circuitry sets the TOD alarm interrupt bit in the IR_STATUS register.

Table 3. TOD_ALARM Bit Format

Bits	Description
<63 : 0>	Value of the TOD_CLOCK register that will cause the support circuitry to set the TOD alarm interrupt.

NOTE: The TOD_ALARM interrupt is accurate to within 7 cpuclk periods because the support circuitry compares bits <63 : 3> of the TOD_ALARM register to bits <63 : 3> of the TOD_CLOCK register.

Atomic Memory Mover Interrupt

Bit 35 (MOVER bit) of the IR_STATUS register is the atomic memory mover interrupt bit. When set to 1, the atomic memory mover interrupt bit indicates that an atomic memory mover operation is complete.

Option-specific Error Interrupts

Bits <42 : 36> of the IR_STATUS register are the option-specific error interrupt bits. When set to 1, an option-specific error interrupt bit indicates that an error was detected and reported in one of the option-specific error registers. The “Option-specific Error Registers” section of this document provides more information on these error registers.

System Machine Check Interrupt (sys_mch_chk_irq_h) Pin

When the support circuitry sets the sys_mch_chk_irq_h pin to 1, this pin signals the microprocessor that the support circuitry initiated a PE prod. A PE prod is an operation that the support circuitry performs to recover a PE from a hung condition.

A PE is in a hung condition when it has stalled, waiting for an event that will never occur. As a simplified example, the following paragraphs describe a process that could cause a PE to enter a hung condition.

1. The microprocessor in a PE issues a remote GET operation that will fill E register 5 with data.

2. The support circuitry receives the GET command, sets the ERS code for E register 5 to EMPTY, and sends a GET request packet to a remote PE.

3. The microprocessor then issues a LOAD command from E register 5.

4. The microprocessor stalls on the LOAD command because it is waiting for the support circuitry to fill E register 5 with data and set the ERS code for E register 5 to a value other than EMPTY.

5. Until the GET command completes, the support circuitry cannot transfer data from E register 5 to the microprocessor. Instead, the LOAD command stalls waiting for a GET response packet from the remote PE. The response packet contains data destined for E register 5.

6. If the remote PE has a hardware failure that prevents it from creating the GET response packet, the local PE enters a hung condition because it is stalled, waiting for a GET response that will never occur.

The support circuitry contains a mechanism, called the PE prod, that releases a PE from a hung condition. Two methods may be used to initiate a PE prod operation: an automatic processor stall timeout or a manual network-router-delivered PE prod.

Processor Stall Timeout

The processor stall timeout (PST) mechanism automatically detects when the microprocessor bus has been idle for more than a specified amount of time. Software specifies the amount of time by using a register in the support circuitry called the processor stall timeout control (PST_CTL) register. Table 4 shows the bit format of the PST_CTL register.

Table 4. PST_CTL Register Bit Format

Bits	Name	Description
<35 : 0>	INTERVAL	Specifies the maximum amount of time in cpuclk periods that the microprocessor bus may be idle. When set to 0, the PST mechanism is disabled.
<62 : 36>	Not applicable	These bits are not used.
63	AUTO_PROD	When set to 1, this bit enables an automatic PE prod operation if the PST programmable timer decrements to 0.

The PST mechanism contains a programmable timer. When the microprocessor bus changes state, the support circuitry transfers the INTERVAL value from the PST_CTL register into the programmable timer. When the microprocessor bus is stalled, the programmable timer decrements in value. The programmable timer decrements in value until the microprocessor bus changes state again (then the support circuitry reloads the timer with the INTERVAL value) or until the programmable timer reaches a value of 0.

When the value of the programmable timer reaches 0, the support circuitry examines the AUTO_PROD bit of the PST_CTL register. When set to 1, the AUTO_PROD bit indicates that the support circuitry may initiate a PE prod operation.

The support circuitry begins a PE prod sequence by setting the sys_mch_chk_irq_h pin of the microprocessor to 1. In addition, the support circuitry sets bit 43 (the PROD bit) of the IR_STATUS register to 1.

A PE prod sequence remains in effect until software writes a 1 to the PROD bit of the IR_STATUS register. Until this occurs, the support circuitry automatically acknowledges all E-register directed actions, but does not perform any associated functions. (The support circuitry will perform all requested non-E-register related operations.)

Network-router-delivered PE Prod

The network-router-delivered PE prod mechanism provides a method for a remote PE (or other remote device) to manually issue a PE prod to the local PE. Software issues a network-router-delivered PE prod by writing a 1 to bit 12 (PROD bit) of the R_RESETS register in the network router. Software does this by using an SPUT E-register command. The “Network Router Control” section in this document provides more information on the R_RESETS register.

When software sets the PROD bit of the R_RESETS register to 1, the network router signals the support circuitry to issue a PE prod operation. As was described in the PST method, the support circuitry begins a PE prod sequence by setting the sys_mch_chk_irq_h pin of the microprocessor to 1. In addition, the support circuitry sets bit 43 (the PROD bit) of the IR_STATUS register to 1.

A PE prod sequence remains in effect until software writes a 1 to the PROD bit of the IR_STATUS register. Until this occurs, the support circuitry automatically acknowledges all E-register directed actions, but does not perform any associated functions. (The support circuitry will perform all requested non-E-register related operations.)

Error Information

The support circuitry uses several mechanisms to signal the microprocessor when an error occurs. These mechanisms are:

- The option-specific error registers
- The ERS code
- The fill error (fill_error_h) microprocessor pin
- The command fail (cfail_h) microprocessor pin

The following subsections provide descriptions of the functions associated with each mechanism.

Option-specific Error Registers

The options used in the support circuitry, network router, and I/O controller contain registers that provide information on the status of error conditions. These registers are:

- C_ERR registers in the C option
- M_ERR registers in the M option
- R_ERR registers in the R option
- I_ERR registers in the I option

C_ERR Registers

Bit 36 (C_ERR bit) of the IR_STATUS register is the C-option error interrupt bit. When set to 1, the C_ERR interrupt bit indicates that the support circuitry is reporting an error that is identified in the C-option error registers. The C option contains four error registers: C_ERR0, C_ERR1, C_ERR2, and C_ERR3.

NOTE: The hardware guarantees that all nondeferred software errors are reported in the C_ERR and IR_STATUS registers before the GETS bit or the PUTS bit of the EREG_PENDING register return to 0.

C_ERR0

The C_ERR0 register contains status bits that indicate what type of errors occurred. The state of these bits in the C_ERR0 register are not affected by the error enable bits in the C_ERR1 register.

C_ERR1

The C_ERR1 register enables the reporting of errors. When a bit of the C_ERR1 register is set to 1 and the corresponding status bit of the C_ERR0 register is set to 1, the support circuitry will set the C_ERR interrupt bit in the IR_STATUS register. When the reporting of an error is disabled, the corresponding status bit of the C_ERR0 register still indicates when an error condition occurs.

In addition to enabling interrupts, the C_ERR1 register controls the function of the *fill_nocheck_h* microprocessor pin. Bit 59 of the C_ERR1 register controls these functions. When set to 0, bit 59 signals the support circuitry to set the *fill_nocheck_h* microprocessor pin during any transfer of data from the support circuitry to the microprocessor. When set to 1, bit 59 signals the support circuitry to set the *fill_nocheck_h* microprocessor pin only during transfers of data from an auxiliary register in the support circuitry to the microprocessor.

C_ERR2

The C_ERR2 register identifies the first enabled status bit in the C_ERR0 register that sets to 1. Bits <5 : 0> of the C_ERR2 register identify one of the 64 possible bits in the C_ERR0 register.

As an example, software may set bits 3, 6, 19, and 22 of the C_ERR1 register to 1. This action enables the reporting of *channel data error 3*, *channel control error 2*, *E-register response FIFO parity error*, and *E-register RAM parity error* errors.

After receiving an interrupt due to the support circuitry setting the C_ERR bit of the IR_STATUS register to 1, the microprocessor reads the value of the C_ERR0 register. Bits 3 and 19 may be set to 1, which indicates that both a *channel data error 3* and an *E-register response FIFO parity error* occurred. The microprocessor must then read the value of C_ERR2 register bits <5 : 0> to determine which error occurred first. If bits <5 : 0> of the C_ERR2 register are set to 3, the *channel data error 3 error* occurred first. If bits <5 : 0> of the C_ERR2 register are set to 19, the *E-register response FIFO parity error* occurred first.

C_ERR3

If the C_ERR2 register is set to 60 (which indicates an *E-register context 0 software error* occurred first) or 61 (which indicates an *E-register context 1 software error* occurred first), the C_ERR3 register contains error payload information. If the C_ERR2 register is set to any value other than 60 or 61, the contents of the C_ERR3 register are undefined.

M_ERR Registers

Bits <40 : 37> (M[3 : 0]_ERR bits) of the IR_STATUS register are the M-option error interrupt bits. When set to 1, an M[3 : 0]_ERR interrupt bit indicates that the support circuitry detected an error that is identified in an M-option error register. Each M option contains four error registers: M_ERR0, M_ERR1, M_ERR2, and M_ERR3.

M_ERR0

An M_ERR0 register contains status bits that indicate what type of errors occurred. The states of these bits in the M_ERR0 register are not affected by the error enables in the M_ERR1 register.

M_ERR1

The M_ERR1 register enables the reporting of error conditions. When a bit of the M_ERR1 register is set to 1 and the corresponding status bit of the M_ERR0 register is set to 1, the support circuitry will set an M_ERR interrupt bit in the IR_STATUS register. When error reporting is disabled, the corresponding status bit of the M_ERR0 register still indicates the state of the interrupt.

M_ERR2

The M_ERR2 register identifies the first enabled status bit in the M_ERR0 register that sets to 1. Bits <4: 0> of the M_ERR2 register identify one of the 20 possible bits in the M_ERR0 register. As an example, software may set bits 0 and 2 of the M_ERR1 register to 1. This action enables the reporting of the *channel parity error* and *channel data error* errors.

After receiving an interrupt due to the support circuitry setting the M_ERR bit of the IR_STATUS register to 1, the microprocessor reads the value of the M_ERR0 register. Bits 0 and 2 may be set to 1, which indicates that both a *channel parity error* and a *channel data error* occurred. The microprocessor must then read the value of M_ERR2 register bits <4 : 0> to determine which error occurred first. If bits <4 : 0> of the M_ERR2 register are set to 2, the *channel data error* occurred first. If bits <4 : 0> of the M_ERR2 register are set to 0, the *channel parity error* occurred first.

M_ERR3

The M_ERR3 register contains error payload information.

R_ERR Registers

Bit 41 (R_ERR bit) of the IR_STATUS register is the R-option error interrupt bit. When set to 1, the R_ERR interrupt bit indicates that the support circuitry detected an error that is identified in the R-option error registers. The R option contains four error registers: R_ERR0, R_ERR1, R_ERR2, and R_ERR3.

NOTE: The R option may detect two errors that are not reported in an R_ERR or another error register. The first error occurs when the R option receives a request to access a register, but the opcode for the request is not a valid opcode. The second error occurs when the R option receives a request to access a hardware register, but the address for the register is not a valid address. If either of these errors occurs, the R option ignores the reference and does not create a response packet for the request that caused the error.

R_ERR0

The R_ERR0 register contains status bits that indicate what type of errors occurred. The state of these bits in the R_ERR0 register are not affected by the error enables in the R_ERR1 register.

R_ERR1

The R_ERR1 register enables the reporting error interrupts. When a bit of the R_ERR1 register is set to 1 and the corresponding status bit of the R_ERR0 register is set to 1, the support circuitry sets the R_ERR interrupt bit of the IR_STATUS register to 1. When the reporting of an error is disabled, the corresponding bit of the R_ERR0 register still indicates the state of the interrupt.

R_ERR2 and R_ERR3

These registers are not used. When software reads from these registers, the support circuitry returns all 0's.

I_ERR Registers

Bit 42 (I_ERR bit) of the IR_STATUS register is the I-option error interrupt bit. When set to 1, the I_ERR interrupt bit indicates that the support circuitry detected an error that is identified in the I-option error registers. The I option contains four error registers: I_ERR0, I_ERR1, I_ERR2, and I_ERR3.

I_ERR0

The I_ERR0 register contains status bits that indicate what types of errors occurred. The state of these bits in the I_ERR0 register are not affected by the error enables in the I_ERR1 register.

I_ERR1

The I_ERR1 register enables the reporting error interrupts. When a bit of the I_ERR1 register is set to 1 and the corresponding status bit of the I_ERR0 register is set to 1, the support circuitry sets the I_ERR interrupt bit of the IR_STATUS register to 1. When the reporting of an error is disabled, the corresponding bit of the I_ERR0 register still indicates the state of the interrupt.

I_ERR2 and I_ERR3

These registers are not used. When software reads from these registers, the support circuitry returns all 0’s.

ERS Code

Each E register has an associated ERS code that indicates the present state of the E register. Table 5 lists the possible ERS code values and describes each state.

Table 5. ERS Codes

Bit 1	Bit 0	Name	Description
0	0	EMPTY	The E register is empty and waiting for data.
0	1	FULL_F	The E register is full with a fault. A deferred software error was detected during one of the following E-register commands: GET STORE_CENT GETV The content of the E register is error payload information.
1	0	FULL	The E register is full.
1	1	FULL_SR	The E register is full; however, a previous message send (SEND) command that used the E register was rejected at the destination PE.

When the ERS code is set to FULL_F, a deferred software error occurred during one of the E-register commands listed in Table 5. A deferred software error is a software error that occurs, but does not need to be reported until the microprocessor attempts to read the data or use the data for another E-register command.

As an example of a deferred software error, the global index may be out of range while the support circuitry performs a GET E-register command. If this occurs, the support circuitry stores error payload information in the destination E register and sets the ERS code for the E register to FULL_F.

The microprocessor is notified of the error when it tries to use data from the E register that has the ERS code set to FULL_F. For example, the microprocessor is notified of an error when it tries to issue a LOAD command from the E register with the ERS code set to FULL_F, when it tries to issue a PUT or SEND command and the source E register for the command is the E register with an ERS code set to FULL_F, or when it tries to issue a command and the more operands (MO) block for the command contains the E register with the ERS code set to FULL_F.

When the microprocessor issues an E-register command that attempts to use information from an E register that has the ERS code set to FULL_F, a nondeferred software error occurs. This nondeferred software error is reported as error type 3, error type 4, or error type 5 in the C_ERR3 register.

The C_ERR3 register identifies the source or destination E register for a command, or it identifies an E register in the MO block of E registers for a command that has an ERS code set to FULL_F.

After identifying an E register that has an ERS code set to FULL_F, software may set the ERS code for that register to FULL (using an ERS_WRITE E-register command) and then read the contents of the register. The E register contains error payload information in one of three formats: Format A, Format B, or Format C.

NOTE: Hardware doe not guarantee that the error payload contents of the E register are not modified before software reads the information from the E register. It is possible for a STORE command that was in the command pipeline to emerge from the microprocessor and overwrite the contents of the E register before the microprocessor can issue a read command to retrieve the E-register contents.

Table 6. All Possible Deferred Software Errors

Error Type	Description	Valid Error Payload Fields
9	Global index out of range	B1 B2 B3 B4
11	User or global virtual address from local index is out of range, or bit 31 of the user or global virtual address is set to 1	B1 B2 B3 B4
14	Base virtual address (user or global) is out of range	B1 B2 B3 B4
15	Post centrifuge PE range error (more than 12 significant bits)	A1 A2 A4 A5
16	Post centrifuge index offset range error (more than 38 significant bits)	A1 A2 A4 A5
17	User or global virtual address generated from adding the base virtual address to the index offset is out of range, or bit 31 of the user or global virtual address is set to 1	A1 A2 A4 A5
18	Virtual PE from the STT is out of range	C1 C2 C3 C4 C5 C6 C7 C8 C5 is only valid for Global addresses
19	STT permissions violation	C1 C2 C3 C4 C5 C6 C7 C8 C5 is only valid for Global addresses
20	GTB range check violation	C1 C2 C3 C4 C5 C6 C7 C8

Software can enable or disable the reporting of deferred errors during a LOAD command. To do this, software uses bit 63 (NO_FAULT_ON_LOAD bit) of the USER_CTL register. When set to 1, this bit disables the reporting of deferred errors during the LOAD command.

Fill Error (fill_error_h) Pin

The support circuitry sets the fill_error_h pin of the microprocessor when either of the following conditions occur:

- A multiple-bit data error occurred when the microprocessor performed a direct local memory read command.
- An M-option to C-option channel parity error occurred during a direct local memory read reference.

When a multiple-bit data error occurs, software reads the value of bits <7 : 4> (MULTIPLE bits) of the M_ERR0 register to determine what event caused the support circuitry to set the fill_error_h pin.

Command Fail (cfail_h) Pin

The support circuitry sets the cfail_h pin of the microprocessor when the support circuitry detects a parity error in the following information from the microprocessor:

- Address
- Command
- WRITE_BLOCK related data

The support circuitry ignores information on the microprocessor buses for approximately 1 microsecond after setting the cfail_h pin. This delay gives the microprocessor time to stabilize after the cfail_h event occurs.

NOTE: The support circuitry sets the cfail_h pin without setting the command acknowledge (cack) pin. This action causes the microprocessor to enter a partial reset sequence.

Status Information

This section describes the following types of status information: outstanding E-register commands, network router state, and option revisions.

Outstanding E-register Commands

Software can monitor the status of outstanding E-register commands. An outstanding command is a command that generated a request for which the support circuitry did not receive a response. As an example of monitoring outstanding E-register commands, software may monitor when an E-register command received a response to ensure that the command modified a memory location before software issues a subsequent command that uses the same memory location. Outstanding commands are grouped into two types: GET-related commands and PUT-related commands.

GET-related Commands

GET-related commands return data from global memory to an E register or otherwise modify an E register's contents. The following E-register commands are GET-related commands:

- CSWAP
- GET_ADD
- DGET
- GET_INC
- GET
- MSWAP
- GETV
- SGET
- STORE_CENT
- STORE_PLZ

Bit 0 (GETS bit) of the E-register pending (EREG_PENDING) register indicates the status of GET-related commands. When set to 1, the GETS bit indicates that at least one GET-related command is outstanding.

STORE and ERS_WRITE commands do not affect the status of the GETS bit. Hardware guarantees that a STORE command always completes after a prior command or before a subsequent command. As an example, hardware guarantees that a GET or PUT command that uses the same E register as a subsequent STORE command will complete before the STORE command can overwrite data in the E register. ERS_WRITE commands do not affect the status of the GETS bits because they do not modify the contents of an E register.

The hardware guarantees that all nondeferred software errors generated for a GET-related command are reported in the C_ERR and IR_STATUS registers before the GETS bit of the EREG_PENDING register sets to 0.

PUT-related Commands

PUT-related commands request to transfer data from a source E register to global memory. The following E-register commands are PUT-related commands.

- DPUT
- PUT
- PUTV
- SEND
- SPUT
- CSWAP
- GET_ADD
- GET_INC
- MSWAP

Bit 1 (PUTS bit) of the EREG_PENDING register indicates the status of PUT-related commands. When set to 1, the PUTS bit indicates that at least one PUT-related command is outstanding.

The hardware guarantees that all nondeferred software errors generated for PUT-related commands are reported in the C_ERR and IR_STATUS registers before the PUTS bit sets to 0.

NOTE: The support circuitry considers SPUT commands complete as soon as they enter the interconnect network. The PUTS bit of the EREG_PENDING register does not reflect the completion state of an SPUT command. An SGET command may be used to determine the completion of an SPUT command.

E-register Pending Context Select

Software can monitor the outstanding GET-related requests and PUT-related requests for either, or both, E-register contexts. To do this, software sets bits <1 : 0> (CONTEXT bits) of the E-register pending context select (EREG_PENDING_CONTEXT) register to the appropriate value.

Network Routers

Software can monitor the status of several network router parameters. To do this, software reads the value of one of the router state (R_STATE[8 : 0]) registers. Table 7 lists the R_STATE[8 : 0] registers and shows what channels are assigned to each register.

Table 7. R_STATE[8 : 0] Registers

Register	Assigned to
R_STATE[0]	+X channel (the network router port assigned to +X)
R_STATE[1]	+Y channel (the network router port assigned to +Y)
R_STATE[2]	+Z channel (the network router port assigned to +Z)
R_STATE[3]	–X channel (the network router port assigned to –X)
R_STATE[4]	–Y channel (the network router port assigned to –Y)
R_STATE[5]	–Z channel (the network router port assigned to –Z)
R_STATE[6]	Support circuitry channel (the network port)
R_STATE[7]	I/O channel (the I/O controller port)
R_STATE[8]	Outstanding transaction identification (TID) flags

R_STATE[7 : 0] Registers

Software can determine whether packet information is flowing through the virtual channel first-in first-out (FIFO) buffers of the network router ports. To do this, software reads the value of the R_STATE registers for the network router ports (R_STATE[7 : 0]).

When a block trap bit (for example, BLK_TRAP_VC1 bit) of an R_STATE register is set to 1, a 0 value in the corresponding acknowledge trap bit (for example, ACK_TRAP_VC1 bit) indicates that the packet information has not yet flowed through the virtual channel. Software can use this information to determine when too much time passes between when a block is detected and when the acknowledge is received.

R_STATE8 Register

Software can determine when a transaction is outstanding by reading bits <31 : 0> (TIDS_OUT bits) of the R_STATE8 register. When set to 1, a TIDS_OUT bit indicates that the corresponding transaction is outstanding. The support circuitry resets the TIDS_OUT bits when the support circuitry receives a flow control reset from the network router.

Option Revision

Several hardware options in the CRAY T3E system contain a read-only register that identifies the option and indicates the revision level of the option. Table 8 lists the option revision registers and shows which option they reside in.

Table 8. Option Revision Registers

Register	Option	Description
C_REV	C	The C option is part of the support circuitry.
M_REV	M	The M option is part of the support circuitry.
R_REV	R	The R option is the network router.
I_REV	I	The I option is part of the I/O controller.

The formats of the option revision registers are identical. Table 9shows the bit format of an option revision register.

Table 9. Option Revision Register Bit Format.

Bits	Name	Description
0	Not applicable	This bit is not used.
<11 : 1>	MANUFACTURER_ID	On all CRAY T3E options, these bits are set to 54 to indicate that the manufacturer is LSI.
<27 : 12>	PART_NUMBER	These bits identify the option and are equal to the last four digits of the CRI part number for the option. 2000 = M option 2100 = C option 2200 = I option 2300 = R option
<31 : 28>	VERSION	These bits identify the revision level of the option. 0 = A0 1 = A1 etc.
<63 : 32>	Not applicable	These bits are not used.

Control Information

Software can control some functions of the hardware. These functions include:

- Data and instruction stream buffer usage
- Deferred error reporting on LOADs
- Advanced fill usage
- Hardware resets
- I/O controller port usage

Data and Instruction Stream Buffers

Software can control two aspects of the data and instruction stream buffers: the prefetch aggressiveness and the stream buffer depth. To control the prefetch aggressiveness, software uses the bits <3 : 0> of the user control (USER_CTL) register.

To control the stream buffer depth, software uses bit 4 (DOUBLE_ENABLE bit) of the C-option system control (C_SYS_CTL) register. When set to 1, this bit indicates that the maximum number of stream buffers that may be assigned to a stream is 4. When set to 0, this bit indicates that the maximum number of stream buffers that may be assigned to a stream is 2.

The assignment of stream buffers is a dynamic processes. For example, at one point in time, the stream buffers may support six streams and each stream uses two stream buffers. At another point in time, the stream buffers may support five streams and one of the streams uses four stream buffers while the other four streams use two stream buffers; however, the DOUBLE_ENABLE bit must be set to 1 for this condition to occur.

The 12 stream buffers support up to three doubled streams or six nondoubled streams. In addition, the stream buffers support combinations of doubled and nondoubled streams. As an example, the stream buffers support two doubled streams and two nondoubled streams.

Deferred Error Reporting on LOADs

Software can enable or disable the reporting of deferred errors during a LOAD command. To do this, software uses bit 63 (NO_FAULT_ON_LOAD bit) of the USER_CTL register. When set to 1, this bit disables the reporting of deferred errors. The “ERS Codes” section of this document provides more information on deferred errors.

Advanced Fill

Software can enable or disable advanced fill operations. During an advanced fill operation, the support circuitry initiates a direct local memory load or a LOAD E-register sequence in the same clock period as it receives the request from the microprocessor. To enable advanced fill, software uses bit 3 (ADV_FILL_ENABLE bit) of the C_SYS_CTL register. When set to 1, this bit enables the advanced fill operation.

Hardware Resets

Software can reset various components in the system hardware. To do this, software writes a 1 to the appropriate bits in the network router resets (R_RESETS) register (refer to Table 11).

Table 10. R_RESETS Register Bit Format

Bits	Name	Description
<11 : 0>	Not applicable	These bits are not used.
12	PROD	When set to 1, this bit initiates a PE prod operation. More information on the PE prod operation is provided in the “System Machine Check Interrupt” section of this document.
13	R	When set to 1, this bit initiates a reset of the R option (network router).
14	I	When set to 1, this bit initiates a reset of the I option (I/O controller).
15	PE	When set to 1, this bit initiates a reset of the microprocessor, the C option (support circuitry), and the M option (support circuitry).
16	DISABLE_E	When set to 1, this bit initiates a reset of only the microprocessor.
<63 : 17>	Not applicable	These bits are not used.

I/O Controller Ports

Software can enable or disable various ports in an I/O controller. To do this, software sets the bits of the I-option status and control (I_STATUS) register to the appropriate values (refer to Table 11). When a port is disabled, data that arrives on the port is accepted, and acknowledges are sent over the outgoing channel; however, the data is ignored.

Table 11. I_STATUS Register Bit Format

Bits	Name	Description
0	PORT 00 DISABLE	When set to 1, this bit disables the port between the I/O controller and the network router in position 00 on a printed circuit board (PCB).
1	PORT 01 DISABLE	When set to 1, this bit disables the port between the I/O controller and the network router in position 01 on a PCB.
2	PORT 10 DISABLE	When set to 1, this bit disables the port between the I/O controller and the network router in position 10 on a PCB.
3	PORT 11 DISABLE	When set to 1, this bit disables the port between the I/O controller and the network router in position 11 on a PCB.
4	SCX_OUT	When set to 1, this bit disables all GigaRing output packet transfers.
5	SCX_IN	When set to 1, this bit disables all GigaRing input packet transfers.
<7 : 6>	Not applicable	These bits are not used.
<14 : 8>	BOARD_REV	These bits indicate the revision level of the PCB.
15	Not applicable	This bit is not used.
16	SME_ENABLE	When set to 1, this bit enables the scan master.
<63 : 17>	Not applicable	These bits are not used.

Index

R

S

T