Laser I/O Port Module (IOP)
Functional Specification
Preliminary

Revision/Update Information: Revision 1.1

Date: 14–August–1991

Authors:
Bill Lin, Roger D. Pannell, Elbert Bloom,
Craig Keefer, Andy Stewart, Doug Neale
Dan Wissell

The Laser I/O Port Module interfaces up to four separate I/O buses (I.E. XMI, FutureBus+, etc.) directly to the Laser System Bus.

HARDWARE ENGINEERING

Digital Equipment Corporation

14–August–1991

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

This document was prepared using VAX DOCUMENT, Version 1.2

CONTENTS

FIGURES

TABLES

Table 1: REVISION History

Revision	Date	By Whom	Description
0.1	25-Jun-1990	Elbert Bloom	Initial INTERNAL release
0.2	10-Jul-1990	Elbert Bloom	Initial External release
0.3	21-AUG-1990	Elbert Bloom	Updates from External Spec Review
1.0	28-Feb-1991	Elbert Bloom	Updates from design changes, etc
1.1	14-Aug-1991	Elbert Bloom & Co.	Updates from design changes, etc

CHAPTER 1

INTRODUCTION

Every Laser System includes one Laser I/O Port Module (IOP). The IOP interfaces up to four separate I/O buses (i.e. XMI, FutureBus+, etc.) directly to the Laser System Bus. The module number of the IOP is "E2044".

1.1 Scope

This specification describes the functional, physical, and environmental characteristics of the Laser I/O Port Module.

1.2 Related Documents

The following documents are related to or were used in the preparation of this document.

- Laser System Bus (LSB) Specification
- Laser to XMI Board (LAMB) Specification
- Futurebus+ Bus to Laser Gateway (FLAG) Specification
- Alpha SRM
- Laser Alpha Processor Module (LEP)

1.3 Terminology and Conventions

Below is a list of terms and conventions used in this specification to describe the function and operation of the IOP.

- IOP (E2044) - Laser I/O Port Module
- HIC (DC7304A) - Hose Interconnect Chip
- IPC (DC7303A) - I/O Port Chip
- Up HIC - Hose Interconnect Chip that receives transactions from the UP HOSE
- Down HIC - Hose Interconnect Chip that transmits transactions to the DOWN HOSE

- LSB - Laser System Bus
- LAMB - Laser to XMI Adapter Module
- FLAG - FutureBus+ to Laser Gateway Adapter Module
- FB+ - FutureBus+
- HOSE - The interface between the IOP and a single I/O Bus Adapter Module.
- DOWN HOSE - The cable which transmits data and control information from the IOP to an I/O Bus Adapter Module.
- UP HOSE - The cable which transmits data and control information from an I/O bus adapter module to the IOP.
- MORE - Refers to the MORE protocol as defined by the XMI Specification
- READ/MODIFY/WRITE - A READ/MODIFY/WRITE sequence is an atomic operation which the IOP executes to write less than a double hexaword memory block size to memory. Only the IOP can execute this sequence on the LSB (see Section 3.2.4).
- VORTEX - Refers to the Vortex Bus which interconnects the HIC and IPC Chips
- NUMBERING CONVENTION - All numbers are decimal unless otherwise indicated. Where there is ambiguity, numbers other than decimal are indicated with the name of the base following the number in parentheses, e.g., FF (hex).
- UNDEFINED - Operations specified as UNDEFINED may vary from moment to moment. UNDEFINED operations must not cause the I/O Port Module to hang, i.e., reach a state from which there is no transition to a normal state.
- INCREASING ADDRESS CONVENTION - Figures which depict registers or memory follow the convention that increasing addresses run from right to left, or from top to bottom.
- REGISTER BIT FIELD MNEMONICS - Bits or fields in figures are shown with a mnemonic which indicates the characteristic of the bit or field, as follows:
  - 0: - Indicates that the initialization state of a bit is "ZERO"
  - 1: - Indicates that the initialization state of a bit is "ONE"
  - R/W: - A read/write bit to the user. It may be read and modified by hardware.
  - RO: - A read-only bit to the user. Only hardware can change the value of the bit. Hardware can look at the bit field. User writes to the bit are ignored.
  - WO: - A write-only bit to the user. It is always read as a 0.
  - W1C: - A write-one-to-clear bit. It may be read by the user and may be cleared by writing a "1" to the bit. The hardware can change the value of the bit, and can look at the bit. A user write of "0" has no effect on the bit.
  - NI: - Not Implemented
  - MBZ: - Must Be Zero
  - RTC: - Read to Clear. Used specifically in the LILIDx Register

1.4 Overview

The Laser I/O Port Module (IOP) is a single extended hex module. Every Laser System includes one IOP. The IOP is always physically configured to node 8. This slot is dedicated as both the highest and the lowest arbitration priority node on the Laser System Bus (LSB), thus enabling the IOP to arbitrate for the LSB at either priority. The IOP will usually arbitrate for the LSB at the highest priority. Approximately every sixth arbitration cycle, however, it will arbitrate twice on the lowest priority in order guarantee that other nodes will never get totally locked out from winning the LSB. The LSB priority arbitration scheme is as follows:

```
 LOWEST                                                    HIGEST
PRIORITY                                                  PRIORITY
--------                                                  --------

 NODE  NODE  NODE  NODE  NODE  NODE  NODE  NODE  NODE  NODE
  0     1     2     3     4     6     7     8     9     5
```

The IOP is transparent to system software except for initialization and error handling.

1.4.1 Laser I/O Subsystem Overview

The interface path between the IOP and an individual I/O bus adapter module is known as the HOSE. A single HOSE consists of two unidirectional cables physically bundled together which may be up to 9 feet each in length. One half of the cable (Down Hose) transmits data and control information from the IOP to the I/O bus adapter module. The other half of the cable (Up Hose) transmits data and control information from the I/O bus adapter module to the IOP. The HOSE data path is a longword (32 bits) in width in each direction. Transfers across the HOSE are full duplex.

The IOP interfaces the Laser System Bus to up to four (4) different I/O buses via separate I/O bus adapter modules. Two I/O adapter modules currently being planned are the Laser to XMI (LAMB) adapter and the Laser to FutureBus+ Adapter (FLAG).

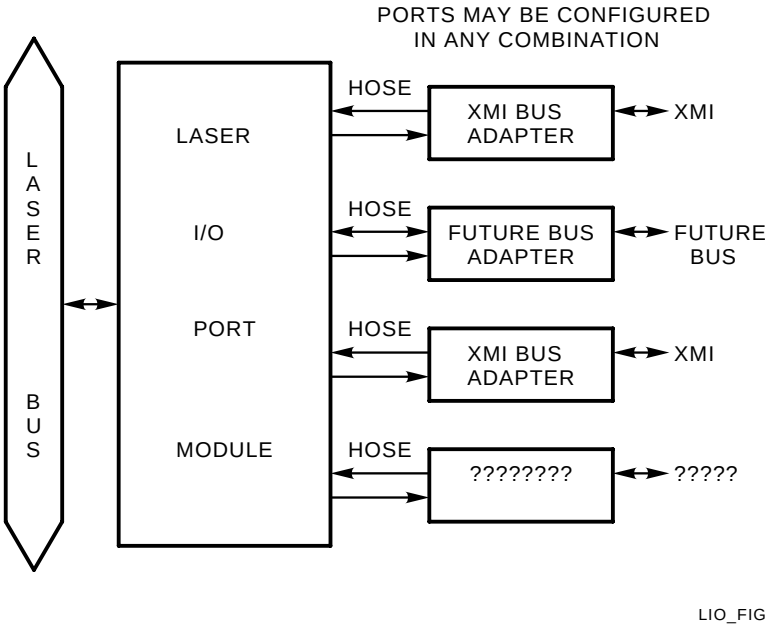
Figure 1–1 shows a top level block diagram of the Laser I/O subsystem.

1.4.2 IOP Module Overview

The IOP houses four large gate arrays. Two of these arrays are called HICs (Hose Interconnect Chips). The HICs are identical, but are configured to be an Up HIC and a Down HIC. The other two arrays are called IPCs (I/O Port Chips). See Figure 1–2.

The IPCs interface the IOP to the LSB. The IPCs are identical arrays. However, they do not function identically on the IOP. Much of the logic used in one array (IPC-A) is disabled in the other array (IPC-B). IPC-A, the primary control array, interfaces the low order quadword

Figure 1–1: Laser I/O Subsystem Block Diagram



and all of the LSB control signals to the IOP. It also houses most of the IOP's CSR Registers. IPC-B interfaces the high order quadword of the LSB to the IOP.

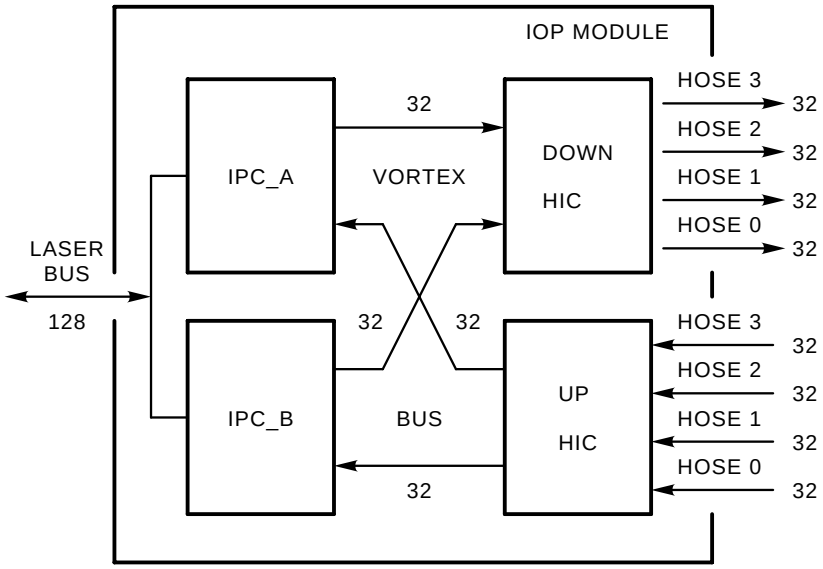
The IPC arrays interface to the HIC arrays via the Vortex Bus. The Vortex Bus is entirely internal to the IOP. Its sole purpose is to function as an interconnect between the IPCs and the HICs.

The HICs are also identical arrays. Each array also functions differently on the IOP. One HIC (the Up HIC) receives packets from the four UP HOSES and transmits them to the IPCs via the Vortex Bus. The other HIC (Down HIC) receives packets from the IPCs via the Vortex Bus and transmits them to the four DOWN HOSES. Consequently, as with the IPCs, there is also logic in each HIC which is used in one, but not the other.

1.4.3 IOP Module Transactions

The IOP, in conjunction with the I/O Adapter modules (i.e. LAMB and FLAG adapters), provides the interface path between the Laser System Bus (LSB) and the I/O devices. The IOP supports five transaction types between the LSB and I/O Adapters. These transactions are listed in Table 1–1.

Figure 1–2: Laser I/O Port Module Block Diagram



NOTE: Numbers in above diagram indicate width of data bus

IOP_FIG

Table 1–1: IOP Transactions

Transaction Type	Initiated by	packets involved
Mailbox	CPU	Mailbox Cmd, Mailbox Status Return
Interrupt	I/O Device or IOP	INTR/IDENT, INTR/IDENT Status Return
DMA Read	I/O Device	DMA READ, DMA READ Data Return
DMA READ/MODIFY/WRITE	I/O Device	DMA WMASK
		DMA IREAD, DMA READ Data Return
DMA Write	I/O Device	DMA Masked Write, DMA Unmasked Write

The IOP transfers information between the LSB and I/O Adapter modules by transmitting and receiving *packets* across the hose(s). Mailbox, Interrupt, and DMA READ/IREAD transactions consist of packet pairs (i.e. Command and Status Return packets). For example, the Mailbox transaction consists of a Mailbox Command Packet and a Mailbox Return Status Packet. The Interrupt Transaction consists of INTR/IDENT command and INTR/IDENT Status return packet. The DMA READ/IREAD transactions are made up of a DMA READ /IREAD Request Packet followed by a DMA Read Data Return packet. The DMA WRITE /WMASK transactions are "disconnected" (i.e. write-and-run) and therefore have no return status packet.

Internal Error Interrupts, however, are generated and ended entirely within the IOP. No Interrupt packet is received on the Up Hose and no INTR/IDENT Status return packet is sent back on the Down Hose.

The IOP may pipeline up to two DMA transactions at a time. That is, it may be processing up to two DMA transactions at any given time. This is necessary in order for the IOP to achieve its performance goal of 300+ Megabytes of raw data throughput.

The sections that follow give a general description of each of the five transaction types.

1.4.3.1 Mailbox transactions

CSRs that exist on external I/O buses are accessed via mailbox structures that exist in Laser main memory. When a CPU wants to read or write one of these CSRs, it builds a mailbox structure and loads the Mailbox Pointer Register (LMBPR) in the IOP. This causes the IOP to fetch the Mailbox structure from Laser memory and build a Mailbox Command packet for transmission across one of the four Down Hoses.

The IOP can have up to eight Mailbox Command packets pending at a time in its LMBPR Register, two per CPU. Only one Mailbox Command packet can be processed at a time, however, by the IOP.

The Mailbox Command packet contains the I/O command, I/O target address, and data/mask to be written if the command is a WRITE. If the Command is not a WRITE the data/mask field is I/O Adapter implementation dependent (i.e LAMB or FLAG). The actual command (e.g. CSR Read, CSR Write, etc) is contained in the CMD<31:0> field of the packet.

After the Mailbox Command packet is sent down the hose, the I/O bus adapter module executes the decoded Mailbox Command over the target I/O bus (e.g. XMI or Futurebus+). Status for the successful or unsuccessful Mailbox Command is returned to the IOP via a Mailbox Status Return packet.

Upon receiving the Mailbox Status Return packet from the I/O Adapter module, the IOP executes a READ/MODIFY/WRITE operation on the Laser bus to fetch the Mailbox structure, merge the information from the Mailbox Status Return packet, and write the results back to main memory (see Section 3.2.4).

The READ/MODIFY/WRITE operation requires 1 Laser bus Read transaction followed by 1 Laser bus write transaction. First, the IOP executes a Read to the Mailbox structure in Laser memory. Eleven cycles later, when the Mailbox data is returned, it is merged with the information from the Mailbox Status Return packet and immediately written back to memory using the highest priority arbitration ID to guarantee atomicity.

The information from the Mailbox Status Return Packet that is merged into the Mailbox structure includes return data (if the Mailbox transaction was a read), a device specific field, an error bit if an error was detected, and a Done bit. If the Mailbox Transaction was not a READ, the Return Data Field is UNPREDICTABLE.

If an error is detected by the I/O adapter module during a Mailbox transaction, the Mailbox Status Return packet on the Up Hose will have the error bit set. If the error occured during a Mailbox transaction that was a *write*, the I/O adapter module may also send an INTR /IDENT packet over the Up Hose to notify the appropriate CPU(s). If the error occured during a Mailbox transaction that was a *read*, the I/O adapter module may return an error code in the device specific field. The IOP will merge this data back into the Mailbox structure

in Laser memory and the CPU will read the Mailbox structure to detect the error. The read return data is UNPREDICTABLE.

NOTE

The specific response of the I/O Adapter as described in the above paragraph is implementation dependent, and the appropriate adapter specification should be read.

If the IOP detects an error during a Mailbox transaction, it will log the error and generate an error interrupt to the appropriate CPU(s).

Note: Section 3.2.6 and Section 5.1.7 discuss the mailbox operation in greater depth.

1.4.3.2 Interrupt transactions

The IOP can generate two types of interrupts on the LSB as follows:

- Remote Bus Interrupts which originate from a remote node and are received by the IOP on the Up HOSE.
- IOP generated Error Interrupts which originate within the IOP as a result of an internally detected error condition

1.4.3.2.1 Remote Bus Interrupts

When an I/O interrupt is posted to a CPU, the CPU will subsequently read the *vector* from the IOP. This means that the INTR/IDENT packet generated by an I/O Adapter module must already contain the vector so that the IOP can load it into one of its "Interrupt Levelx Ident Registers" (LILIDx) and have it ready to return to the CPU.

Therefore, when an interrupt occurs on an I/O bus (e.g. XMI or FB+), the I/O bus adapter must first acquire the interrupt vector for that interrupt. On the Futurebus+, the vector is acquired as part of the INTR transaction. On the XMI bus, the vector is acquired using an *IDENT* transaction. If the interrupt was a WE (Write Error) IVINTR, the LAMB module will use a predefined vector instead of executing an IDENT.

Once the I/O bus adapter aquires the IDENT vector from the interrupting I/O device, it sends an INTR/IDENT packet to the IOP over the Up Hose. The INTR/IDENT packet includes the vector and IPL of the interrupt.

Only one Interrupt at a time may be posted by an I/O Bus Adapter at a given IPL from a single HOSE Interconnect.

The IOP loads the vector into the appropriate LILIDx register and generates an interrupt to one or more CPUs by writing to the I/O Interrupt Register (LIOINTR) in LSB broadcast space.

The CPU(s) targeted for the interrupt are determined from the "CPU Interrupt Mask Register" (LCPUMASK) in the IOP. Four separate copies of the IPL received from the HOSE Interconnect are written to the CPU's LIOINTR register, one for each LSB CPU. The IPL from the INTR/IDENT packet is AND'ed with the appropriate field of the LCPUMASK register to form the four copies of the IPL to be written to the CPU's LIOINTR register.

Upon receiving the interrupt request, one of the targeted CPUs will read the IOP's "Interrupt Levelx Ident Register" (LILIDx) corresponding to the requested IPL level causing the IOP to return the vector for the posted interrupt. Other targeted CPUs may either notice the relevant read to the LILIDx register and take a passive release or may actually execute their own read of LILIDx. If another interrupt at the relevant level is pending, the additional CPU read of LILIDx will return that IDENT vector. If no other interrupts are pending at the given level, the IOP will return zeros forcing the CPU to take a passive release.

After one of the CPUs targeted by the interrupt reads the LILIDx register, the IOP sends an INTR/IDENT Status Return packet to the I/O bus adapter module over the Down Hose. The arrival of the INTR/IDENT status return packet at the I/O bus adapter tells it that it can service another interrupt at that IPL, thus providing interrupt transaction flow control.

1.4.3.2.2 IOP Generated Error Interrupts

IOP generated Error Interrupts work the same as Remote Bus Interrupts with the following exceptions.

- IOP generated Error Interrupts always interrupt on IPL Level 17.
- IOP generated Error Interrupts transmit a special vector on the LSB which must be pre-loaded by system software into the IPC Interrupt Vector Register (IPCVR) at system initialization
- The IOP has a special IPC Interrupt Mask bit (INTR_NSES) which must be pre-loaded by software at system initialization. INTR_NSES enables all IOP specific gererated Error Interrupts to be enabled/disabled (I.E. LSB ECC Error, HOSE Parity Error, etc).
- IOP Generated Interrupts do not return an Interrupt Status packet to the Down HOSE.

It should also be noted that, due to the extra IOP generated Error Interrupt, as many as 5 interrupts could be pending on IPL level 17 at any given time; one IOP generated Error Interrupt and four remote node interrupts received by the IOP from the Up HOSE. However, the LSB Specification allows up to seven interrupts at a given IPL to be posted on the LSB at any given time. Therefore, there should never be any danger of overflowing the CPU's interrupt pending flags.

1.4.3.3 DMA Read transactions

I/O modules transfer large blocks of data directly to and from memory using DMA transactions. When an I/O device requests its local I/O bus for a DMA read transaction, the I/O bus adapter acknowledges the read transaction and, if the bus supports it, pends the transaction. This frees the I/O bus for other bus traffic. The I/O bus adapter then transmits a DMA Read Request packet to the IOP on the Up Hose. Included in the DMA Read Request packet is the target Laser address, a Tag field to allow the I/O bus adapter to associate the DMA read request with the DMA Return data packet, and the length code indicating the amount of data requested.

Upon receiving the DMA Read Request packet, the IOP generates a Laser System Bus read transaction. If the read is successful, the I/O port module transmits a DMA read data return packet to the I/O bus adapter on the Down Hose. The DMA read data return packet includes the tag from the corresponding DMA Read Request packet, the length code, an error bit indicating whether or not the DMA read request was successful, and the requested data. The I/O bus adapter then transmits the data across the I/O bus to the appropriate I/O device.

If the read is unsuccessful, the IOP generates an error interrupt and transmits a DMA read data return packet *with the error bit set* to the I/O bus adapter over the Down Hose. The I/O bus adapter then takes the appropriate action on the I/O bus for read errors.

1.4.3.4 DMA Interlock Read transactions

The Laser Memory System does not support hardware memory locks (I.E no VAX IREAD /UWMASK instruction pair equivalency exists on the LSB). VAX CI-Port Architecture devices such as the XCD and DASH, however, require hardware memory locks. Therefore, in order to support these devices on the Laser Platform, the IOP implements a special atomic memory READ/MODIFY/WRITE function which closely resembles a VAX IREAD instruction. Currently, this function is planned to be implemented only for XMI based nodes in conjunction with the LAMB Adapter module.

NOTE

In order for the atomic READ/MODIFY/WRITE function to work correctly changes must be made to the CI-Port Architecture Microcode and additional hooks must be added to the LAMB Module. These changes are not discussed in this specification

VAX CI-Port Architecture devices acquire hardware memory locks using DMA IREAD transactions. All DMA IREAD transactions are Quadword in length. When an I/O device on the XMI issues a DMA IREAD transaction, the LAMB Adapter acknowledges the IREAD and pends the transaction. This frees the XMI for other bus traffic. The LAMB then transmits a DMA IREAD Request packet to the IOP on the Up Hose. Included in the DMA IREAD Request packet is the target Laser address, a Tag field to allow the LAMB Adapter to associate the DMA IREAD request with the DMA Return data packet, and the length code indicating the amount of data requested.

After receiving the DMA IREAD Request packet, the IOP executes a read to Laser memory at the target address. Eleven cycles later, when the data is returned from Laser memory, the IOP immediately writes it back to memory using high priority arbitration to guarantee atomicity (READ/MODIFY/WRITE). The low order bit of the addressed quadword indicates the LOCK status of the location. "1" (one) equals LOCKED, "0" (zero) equals UNLOCKED. During the write cycle the IOP always forces the low order bit of the addressed quadword to a 1 (one) regardless of its original value.

If the READ/MODIFY/WRITE executed without errors, the IOP transmits a DMA Read data return packet to the LAMB adapter on the Down Hose. The DMA Read data return packet includes the tag from the corresponding DMA IREAD Request packet, the length code, an error bit indicating whether or not the DMA IREAD request was successful, and the requested data. Note that the IOP returns the read data to the LAMB unmodified, even though it wrote the data back to memory with the low order bit forced to a one (i.e If the low order bit was read as 0, then it returns a 0. If the low order bit was read as 1, then it returns a 1). The LAMB adapter then transmits the data across the XMI to the appropriate I/O device.

If no errors were detected, the DMA IREAD Request transaction is complete.

If an error is detected on the Up hose (e.g. parity error, sequence error, etc), or if the Laser bus READ/MODIFY/WRITE operation is unsuccessful, the IOP logs the error and generates an error interrupt to the CPU(s).

Note that there is no special DMA WRITE UNLOCK Request packet. The XMI I/O device simply writes the low order bit of the location to "0" via a generic DMA MASKED WRITE Request Packet when it is ready to release the LOCK.

1.4.3.5 DMA Write Transactions

When an I/O device requests a local I/O bus for a DMA write to memory, the I/O bus adapter transmits a DMA write request packet to the IOP over the Up Hose. There are two types of DMA write request packets; masked write and unmasked write. The main difference is that DMA masked write packets require a READ/MODIFY/WRITE (1 read and 1 write) operation on the Laser bus and can be byte masked by the I/O Adapter, whereas a DMA unmasked write packet only requires a single Laser bus write transaction and is always a double-hexaword.

A DMA write request packet is executed as a disconnected (write-and-run) operation and therefore has no status return packet associated with it. Once the I/O bus adapter transmits the DMA write request packet over the Up hose, it considers the transaction complete.

1.4.3.5.1 DMA Unmasked Write

The DMA Unmasked Write packet is the most efficient DMA write that the I/O Port supports. It has *two* major advantages over the DMA Masked Write packet. First, it only requires a single write on the Laser bus, whereas a DMA Masked Write packet requires a READ/MODIFY/WRITE operation on the Laser bus. Secondly, the data length of a DMA Unmasked Write packet is **always** a double-hexaword in Length, whereas a DMA Masked Write packet can contain as small as a byte of valid data (but will usually match the size of the DMA write on the I/O bus). On the XMI this will probably equate to Octaword writes. If so, it would require four READ/MODIFY/WRITE operations on the Laser bus to match just one unmasked double-hexaword write. As can be seen, there is a dramatic increase in DMA write performance whenever a DMA Umasked Write packet can be used in place of a DMA Masked Write packet.

Generally, an I/O adapter module should be able to exploit the DMA Unmasked Write packet whenever an I/O controller is using a "More" type protocol by appending the smaller size writes on the I/O bus into a double-HW and shipping it across the Up Hose as a DMA Unmasked Write packet. The DMA Unmasked Write packet includes the Laser target address for the data and a double-HW of write data.

NOTE

No DMA Unmasked Writes will occur from the XMI activity unless the "XMI MORE" protocol is used.

After receiving the DMA Unmasked Write packet, the IOP executes the write to memory over the Laser System Bus. If no errors were detected, the DMA write transaction is complete.

If an error is detected on the Up hose (e.g. parity error, sequence error, etc), or if the Laser bus write is unsuccessful, the IOP will log the error and generate an error interrupt to the CPU(s).

1.4.3.5.2 DMA masked write request to memory

The DMA Masked Write packet includes the target address for the data, the length code to allow for sequence checking, mask bits in the UPCTL field, and the amount of data required for the DMA masked write.

After receiving the DMA Masked Write packet, the IOP executes a read to Laser memory at the target address. Eleven cycles later, when the data is returned from Laser memory, the IOP merges the DMA write data (using the mask bits from the DMA write packet) and immediately writes it back to memory using high priority arbitration to guarantee atomicity. If no errors were detected, the DMA masked write transaction is complete.

If an error is detected on the Up hose (e.g. parity error, sequence error, etc), or if the Laser bus READ/MODIFY/WRITE operation is unsuccessful, the IOP logs the error and generates an error interrupt to the CPU(s).

1.5 Performance

At maximum throughput, the IOP is capable of utilizing 2/5 of the LSB. This means that for an LSB running at a 17ns cycle time, the IOP is capable of sustaining a total aggregate data throughput across the Laser System Bus of 300+ megabytes per second.

```
  64 bytes       1 transaction   1 LSB cycle    2
------------ X  ------------- X ----------- X --- = ~301 Mbytes/Sec
1 transaction    5 LSB cycles        17ns       5
```

If four I/O Bus Adapter modules are performing DMA simultaneously, each HOSE will be limited to 75 Megabytes per second. However, if less than four HOSES are being utilized, the 300+ Mbytes/Sec can be evenly distributed across the HOSES in use. For example, if a system is configured with three I/O Bus Adapter modules, the IOP can support up to 100 Mbytes/Sec for each HOSE. For systems with only two or one I/O Bus Adapter(s) installed, the HOSE bandwidth is limited by the bandwidth of the Up/Down HOSES (see Table 1–2).

The 300+ megabytes per second throughput rate of the IOP is based on the following assumptions:

- All transfers being received across the HOSES are double hexaword in size
- The Laser System Bus cycle time is 17 ns
- The IOP always wins the Arbitration on each cycle in which it requests the Laser System Bus.
- The IOP never experiences any Memory Bank Contentions

This throughput rate is idealistic. In a real operating system environment it would never be sustainable. It can only be approximated if all transactions from remote buses are packed into Double Hexawords. Interrupts, Mailbox transactions, and DMA Transactions of smaller than Double Hexaword sizes, Memory Bank Contentions, etc will all result in considerable bandwidth degradation. Appendix A describes some of the reasons for this performance degradation.

Table 1–2: IOP Maximum Throughput

Bus	Bandwidth	Notes
LSB	301.1 Mbytes/Sec	Based on 17ns LSB cycle time and 2/5 LSB utilization
DOWN HOSE	104.5 Mbytes/Sec	Based on 34ns Down HOSE cycle time
LAMB UP HOSE	105.2 Mbytes/Sec	Based on 32ns Up HOSE cycle time from LAMB
FLAG UP HOSE	134.7 Mbytes/Sec	Based on 25ns Up HOSE cycle time from FLAG

CHAPTER 2

LASER SYSTEM BUS ADDRESSING

The IOP supports the full address space of the Laser System Bus, or up to one terabyte of memory. This memory space is accessed via a 40-bit memory address space across the Laser System Bus.

Control/Status (CSR) registers which reside internal to the IOP are accessed by other Laser System Bus nodes via a 22 bit hexaword aligned address space. All IOP CSR Registers are located within node 8 CSR Register Address Space as follows:

50 0000 to 51 FFFE - Hexaword Aligned Address

The IOP also accesses the Laser I/O Interrupt (LIOINTR) Register in Laser Broadcast space via the same 22 bit hexaword aligned address range.

CSRs that exist on remote I/O bus nodes are accessed via the Laser System Bus Mailbox protocol.

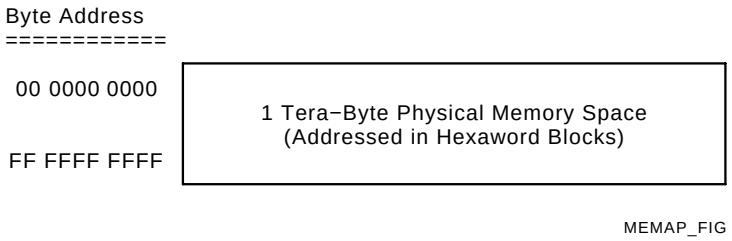
2.1 Laser System Bus Memory Map

The Laser System Bus (LSB) is hexaword aligned. Bits D<34:0> address a specific hexaword within the one terabyte of LSB memory space.

The IOP conforms to the Laser System Bus protocol. It accesses all of memory as 64-byte blocks, using bits D<34:0> of a command cycle as the memory address.

The IOP maps Bits D<34:1> of a command cycle to byte address <39:6>, specifying 2**34 64-byte blocks. Bit D<0> of a command cycle maps to byte address Bit D<5>, specifying which 32-byte sub-block hexaword is to be returned first.

Figure 2–1: IOP Memory Map



2.2 Laser System CSR Space

All Control/Status Registers (CSRs) that exist in Laser nodes are defined to be 32 bits wide and aligned on 64-byte boundaries. (The LMBPR in the IOP is an exception, since it is wider than 32 bits). Laser CSRs are accessed using the Read CSR and Write CSR commands. Bits D<22:1> of the address field in an LSB CSR read/write command cycle are used to specify all LSB CSRs (bits D<33:23> and bit D<0> are always zero during CSR command cycles) as shown in Figure 2–2.

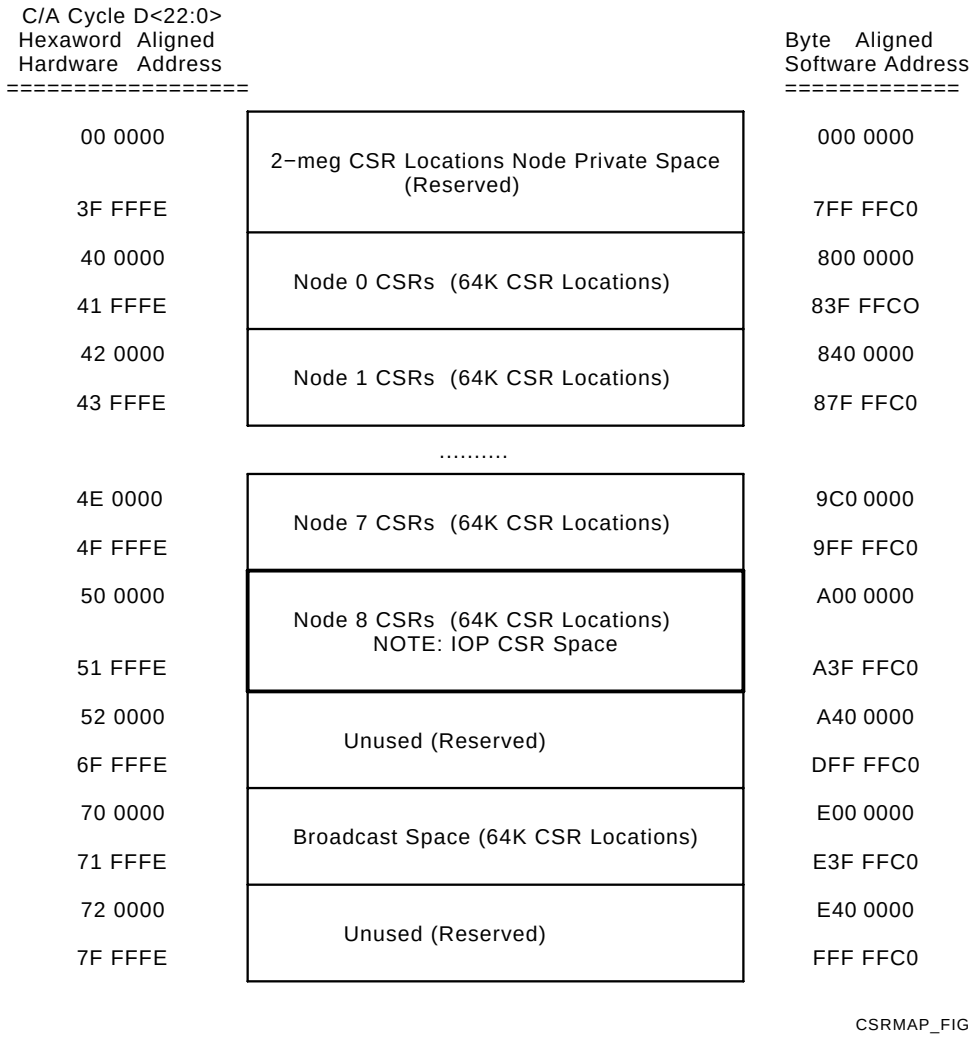
NOTE

It should be noted that D<22:0> of the address field in an LSB CSR read/write command cycle is a *hexaword aligned* address. Five more bits must be added to the least significant bit of the field in order to obtain the byte aligned address.

In this specification, all CSR locations are specified as a function of the state of LSB D<22:0> of a CSR read/ write command cycle.

If a non-existant register within the IOP's node space is addressed the IOP will respond in an UNDEFINED manner. Usually it will ACK the transaction and accept or return data from one of its defined registers, but this is not guaranteed.

Figure 2–2: Laser System CSR Space Map



The first 2-meg CSR locations of Laser CSR space is reserved for local use on each module. References to this region will be serviced by resources local to a module and will therefore never be asserted on the LSB. The IOP does not implement any CSRs in the Reserved portions of CSR Space.

Broadcast space is used for write-only registers that are written in all nodes in a single bus transaction. The IOP uses this region to implement interrupts.

2.3 Laser I/O Port Register Map

Figure 2–3 shows a list of all the CSR Registers implemented internal to the IOP.

Figure 2–3: IOP CSR Register Map

C/A Cycle D<22:0> Hexaword Aligned Hardware Address		Byte Aligned Software Address
LSB Required Registers		
50 0000	Laser Device Reg (LDEV)	A00 0000
50 0002	Laser Bus Error Reg (LBER)	A00 0040
50 0004	Laser Configuration Reg (LCNF)	A00 0080
50 0006	Laser Info Base Repair Reg (IBR)	A00 00C0
50 0010 to 50 001E	Laser Memory Mapping Regs 7−0 (LMMRx)	A00 0200 to A00 03C0
50 0030 to 50 0036	Laser Bus Err Syndrome Regs 3−0 (LBESRx)	A00 0600 to A00 06C0
50 0038 to 50 003A	Laser Bus Err Command Reg 1−0 (LBECRx)	A00 0700 to A00 0740
50 0050 to 50 0056	Interrupt level 3−0 Ident Reg (LILIDx)	A00 0A00 to A00 0AC0
50 0058	CPU Interrupt Mask Reg (LCPUMASK)	A00 0B00
50 0060 to 50 0066	Mailbox Pointer Regs 3−0 (LMBPRx)	A00 0C00
IOP Specific Registers		
50 0100	IPC Node Specific Error Reg (IPCNSE)	A00 2000
50 0102	IPC Vector Reg (IPCVR)	A00 2040
50 0104	IPC Mode Selection Reg (IPCMSR)	A00 2080
50 0106	IPC Hose Status Reg (IPCHST)	A00 20C0
50 0108	IPC Diagnostic Reg (IPCDR)	A00 2100

IOP_REGMAP_FIG

2.4 Accessing Remote I/O Node CSRs

CSRs that exist on external I/O buses (I.E. XMI or FutureBus+) are accessed via mailbox structures that exist in main memory. CSR requests are posted in a mailbox via software. The IOP reads the Mailbox and passes it as a packet to the appropriate bus adapter. When the bus adapter responds with a return status packet the IOP returns the status and CSR Read Data to the Mailbox in memory. Software must not overwrite a mailbox which is still in use (i.e DONE bit not set by the IOP).

The IOP services mailbox requests via a single mailbox pointer CSR (LMBPR) located in the IOP's nodespace. There are actually eight LMBPR registers, two for each CPU node. This allows a single CPU to have up to two mailboxes pending within the IOP at the same time. It also eliminates the possibility of lockouts when multiple CPUs access the LMBPR since each CPU has its own dedicated LMBPRs. However, the I/O system architecture requires that there be only a single software-visible LMBPR address, regardless of the node ID. Consequently, the interface logic inserts the two least significant bits of the CPU node ID in D<2:1> of the command cycle whenever a CPU writes to the LMBPR. The IOP uses the decoded value of D<2:1> to determine which one of its internal LMBPRs receive the data as shown in Table 2–1.

Table 2–1: LMBPR Register Map

D<2:1>	CPU Node
00	CPU Node 0
01	CPU Node 1
10	CPU Node 2
11	CPU Node 3

Mailbox pointers are managed by the IOP hardware. Once a CPU has two Mailbox Transactions pending, additional CSR write transactions to the LMBPR by the same CPU result in the LSB write transaction not receiving acknowledgment (CNF not asserted). Processors use the lack of CNF assertion on writes to the mailbox pointer CSR to indicate a busy status. The write must be re-issued at a later point in time (via software).

Note: Mailbox Transactions are discussed in greater detail in several sections throughout this specification such as Section 1.4.3.1, Section 3.2.6 and Section 5.1.7.

CHAPTER 3

LASER SYSTEM BUS INTERFACE

The Laser System Bus (LSB) is a 128-bit non-pended processor-memory interconnect. *Non-pended* means that the data cycles always bear a fixed relationship with the command /address cycle and the initiator of the transaction waits to send or receive the data. On the LSB, the data cycles always begin on the 11th bus cycle after the corresponding command /address cycle, assuming no node has asserted Laser Stall in the interim. This fixed delay is optimal for the current memory technology. This chapter discusses the LSB signals, transaction types, bus protocol and I/O Port module (IOP) specific behavior with respect to the LSB.

3.1 Laser Signals

The LSB physical interconnect consists of the signals as shown in Table 3–2. Table 3–1 defines the signal types of the LSB signals.

Table 3–1: LSB Signal Types

Type	Description
CLK	This signal type is "input-only" for most Laser modules. Since the clock lines are driven by clock generator circuitry that resides on the IOP, the IOP has input-only module pins like all other Laser modules. In addition, it has output-only pins for supplying a private copy of the clock signals to each Laser module.
OC	OPEN COLLECTOR - This wire is a standard TTL level signal that may be driven by one or more modules. The pull-up is supplied by logic external to the LSB environment.
OD	OPEN DRAIN - This wire may be driven by one or more modules. If not driven, the wire defaults to a deasserted level. Voltage level details can be found in Chapter 9 of the LSB Specification.
STRAP	These wires are selectively connected to GND on the backplane to provide slot identification to the module.

All LSB data and control signals are asserted low, i.e. if the signal is to be asserted (TRUE or 1), the open drain driver pulls the wire low. In the absence of an assertion, the bus terminator pulls the wire high (i.e. de-asserted, FALSE, 0).

Please refer to the Laser System Bus Specification for detailed explanations of each of the following signals.

Table 3–2: Laser Signals

Signal	Wires	Type	Description
D<127:0>	128	OD	Data and command/address[1]
ECC<27:0>	28	OD	Error correction for data cycles
REQ<9:0>	6	OD	Arbitration request
STALL	1	OD	Responder stall
ERR	1	OD	Error detected by any module[2]
CNF	1	OD	Responder confirmation
CA	1	OD	Command cycle
RESET	1	OC	Reset everything
PH0	1	CLK	Clock Phase 0 (Sine)[5]
PH90	1	CLK	Clock Phase 90 (Cosine)[5]
UNIMPLEMENTED BY THE IOP			————————
SHARED	1	OD	Data is cached[3]
DIRTY	1	OD	Memory data is stale[3]
CRD	1	OD	Corrected Read Data[6]
LOCK	1	OD	Lockout[6]
NID<2:0>	3	STRAP	Slot identification[4]
D24SPARE	1	OD	Spare Signal Wire
D64SPARE	1	OD	Spare Signal Wire
	——		
Total	174		

[1]Only D<38:0> are used during command cycles and CSR data cycles. This unusual architectural feature was the result of the datapath partition chosen for the CPU-module-to-LSB-interface chips.

[2]See Laser System Bus Specification.

[3]The I/O Port Chip (IPC) does not need to monitor the LASER SHARED and LASER DIRTY signals because the IPC does not cache data. Furthermore, the READ/MODIFY/WRITE is performed as an atomic operation through the use of high priority on the Write portion of the transaction. Thus, the IPC does not require these two cache coherency signals.

[4]NID<2:0> are hardwired on backplane to provide individual codes for each slot. The I/O port module (IOP) is always the same node in the system therefore it does not need pins to identify its node id.

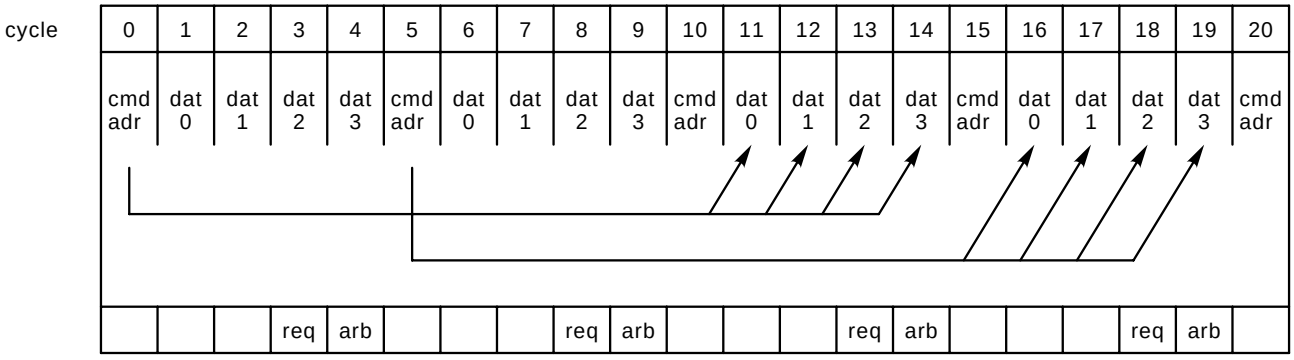
[5]The IOP Module houses the Laser System Bus Clock Generator. The Laser System Bus Clock Generator drives a private copy of the PH0 and PH90 clocks over the LSB backplane for each Laser module. The IOP logic receives PH0 from the LSB Backplane in the same manner as all other LSB Modules. The IOP does not use PH90 of the clock. Please refer to the Laser System Bus Specification, Chapter 9.

[6]The CRD and LOCK signals are CPU specific and, therefore, are not used by the IOP.

3.2 Laser Transactions

All Laser Bus transactions have one command/address (C/A) cycle and four data cycles. Because of memory cycle times, the data associated with a given transaction trails the C/A cycle by 11 cycles. For efficient operation under these conditions, bus transactions are interleaved. Figure 3–1 depicts the interleaved nature of the Laser transactions. In this example, the data that goes with the command/address in cycle 0 begins in cycle 11 and continues through cycle 14. The command/address in cycle 5 has cycles 16-19 as its data cycles. Note the request and arbitration (resolution) cycles that precede the command /address cycles.

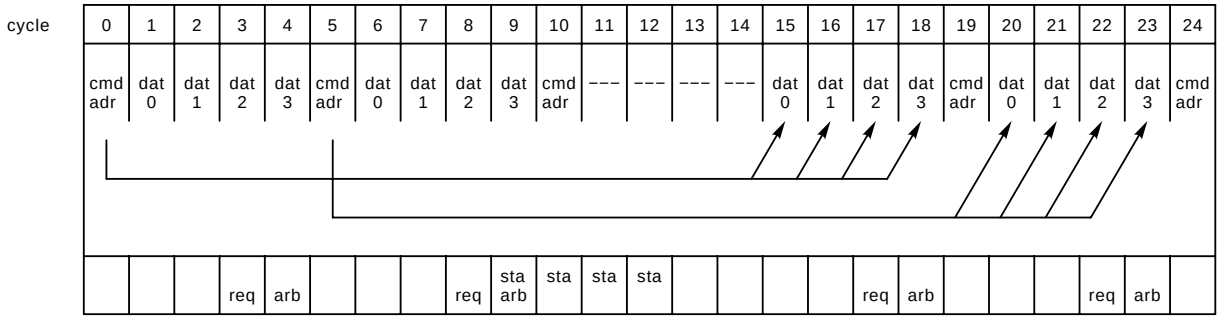
Figure 3–1: Laser Bus Cycles



LASER_BUS_CYCLES

The LSB protocol provides the Laser STALL signal for use by *any* Laser module during *any* transaction to indicate that the data cycles will be delayed by one or more cycles. The STALL signal may only be asserted starting two cycles prior to when the first data cycle would have been required on the bus. This allows the data cycles to be delayed by the same number of cycles as there are STALLs. The STALL has no effect on the cycle reserved for a command/address, i.e. the one following the first STALL. Figure 3–2 shows an example of STALL timing. Note that stalling takes effect two cycles after the signal's assertion. The IOP monitors STALL. The IOP also drives stall for certain rare cases in which multiple back-to-back reads of the IOP's LILID Registers cannot be processed rapidly enough by the IOP.

Figure 3–2: Laser Bus Stall Cycles



STA = STALL
--- = DON'T CARE OR UNDRIVEN CYCLES (undriven if READ, possibly Data0 if WRITE)

STALL_CYCLES

The IOP initiates and responds to the LSB transaction types shown in Table 3–3. The command encodings in the "command" column are for reference purposes only. Please refer to the LSB Specification for detailed descriptions of the various command types.

Table 3–3: IOP-Supported LSB Transaction Types

Command	Initiates	Responds to	Description
000	YES	NO	Read
001	YES	NO	Write
010	NO	NO	Reserved
011	NO	NO	Write victim
100	NO	YES	Read CSR
101	YES[1]	YES	Write CSR
110	NO	NO	Reserved
111	NO	NO	Private

[1]The IOP initiates Write CSR transactions to the LIOINTR register (BSB+00) in the CPUs only. This is how interrupts are posted on the LSB.

The LSB does not have a separate I/O address space. The target address space, whether memory or I/O, is determined by the Laser transaction type. Please refer to Chapter 2 for a description of the LSB address space and for a summary of the IOP Laser-required and node-specific registers and their corresponding addresses.

The IOP serves four primary functions on the LSB. The IOP handles DMA transactions, mailbox operations, interrupts, and CSR transactions. The DMA transactions are initiated by I/O devices to move (generally large) blocks of data between memory and the I/O devices. Write CSRs to Broadcast Space are used by the IOP solely for signaling interrupt requests. Mailbox operations also move data between memory and the I/O devices, but these special

transfers are a means of indirect CPU access to the remote bus CSR locations. The fourth and final function served by the IOP is to respond to LSB Read CSR and Write CSR transactions targeted to its range of node addresses starting at physical hexaword address, 50 0000 HEX.

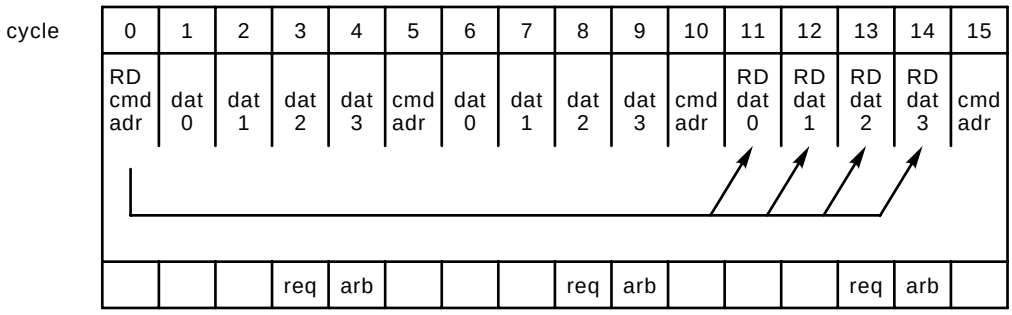
NOTE

It is important to note that the IPCs may pipeline up to two transactions at a time. A pipelined transaction is one that is initiated prior to a previous one's completion. Transactions shall always be serviced on a first in, first out basis regardless of whence they originated or are destined. This applies to both CPU initiated transactions and I/O DMA or interrupt traffic.

3.2.1 DMA Read Transactions

The IOP supports octaword, hexaword and double hexaword Reads from memory. However, Reads of all lengths look like double hexaword Reads on the LSB. Figure 3–3 shows a typical Read transaction on the LSB. The Read command/address is asserted on the LSB in cycle 0. The next 10 bus cycles are reserved for data cycles from previous transactions and for the next two transactions' command/addresses. The Read data is returned by the memory (or CPU cache) during cycles 11 through 14.

Figure 3–3: DMA Read Example



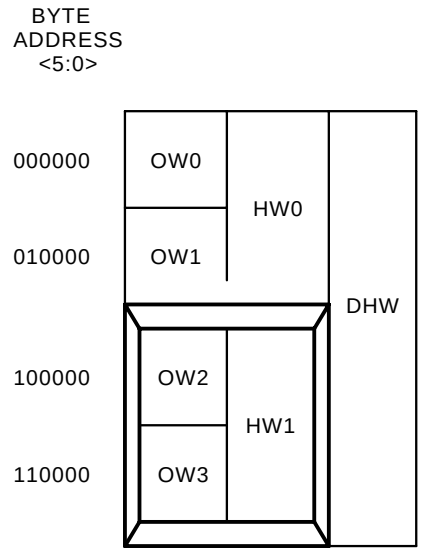
Wrapped Reads on hexaword boundaries are permitted on Laser. The IOP uses wrapped Reads on the LSB when doing so will decrease the latency perceived by the I/O device that is requesting the data. Whenever an I/O bus adapter requests an octaword or hexaword of data from an address with bit<5> asserted, the IOP shall issue a wrapped Read on the LSB. Table 3–4 shows the HOSE transaction lengths and memory addresses for which the IOP shall use a wrapped Read. Figure 3–4 shows yet another representation of when the IOP will generate wrapped Reads.

Table 3–4: Wrapped Reads

Transaction Length	Byte Address (<5:0>)	Wrapped
octaword	0xxxxx	NO
octaword	1xxxxx	YES
hexaword	0xxxxx	NO
hexaword	1xxxxx	YES
double hexaword	xxxxxx	NO

where "x" = don't care

Figure 3–4: Wrapped Reads on the LSB



OW = octaword = 16 bytes
HW = hexaword = 32 bytes
DHW = double hexaword = 64 bytes

Memory addresses/packet lengths in the highlighted box above are retrieved with a wrapped Read.

WRAPPED_READ_FIGURE

3.2.2 Interlocked Read/Unlock Write Transactions - a.k.a. VAX CI-Port Architecture Support

VAX CI-Port Architecture (VAXPort) devices require Interlocked Read (IREAD) and Unlock Write (UWMASK) transactions to access shared software data structures. On the XMI, an I/O device initiates an IREAD transaction to request exclusive access to a hardware-controlled primary lock variable which, when acquired, allows the I/O device to modify a secondary lock variable. This primary lock is a mechanism to insure that only one device has access to the secondary lock at any given time. In a Calypso class system, this primary hardware lock resides on the memory module.

There is no corresponding primary hardware lock in a Laser system. The IOP in conjunction with the XMI bus adapter (LAMB), however, provides the support necessary for these VAXPort devices to function in a Laser system.

To accomplish this, a hardware protocol has been defined to handle the locking and unlocking of a software data structure in Laser memory. These data structures are typically headers of queues that are shared by multiple devices. Bit<0> of a memory quadword is defined as the "lock bit." An asserted state indicates that the quadword is currently owned by some device.

When the IOP receives an IRead command from the LAMB, it performs an atomic READ /MODIFY/WRITE of the corresponding hexaword address (a wrapped read occurs if appropriate). When the Read data is returned by the memory or CPU cache, the IOP sends the requested quadword in the appropriate half of an octaword packet to the LAMB. The IOP also sets the lock bit (bit<0>) in the quadword of interest and completes the atomic operation with a Write back to memory. Please refer to Section 3.2.4 for an explanation of why a READ/MODIFY/WRITE sequence by the IOP is atomic.

The reason the IOP can "blindly" set the lock bit is as follows: if the bit was previously set by another device, then setting it again would have no effect. If the bit was not set previously, then the bit will now be set and the I/O device which initiated this IRead will own the quadword in memory. In either case, the I/O device will see the previous state of the lock bit and know whether it did or did not get ownership of the quadword.

The IOP does not have to perform any special functions to support the relinquishing of the lock variable by the I/O device. A generic quadword masked WRITE on the XMI will be converted to a generic octaword masked Write packet by the LAMB module and the IOP will execute a standard READ/MODIFY/WRITE on the LSB. The quadword of data that the I/O device sends in the masked WRITE command contains the original data with the correct state of the lock bit, i.e. bit<0> is clear.

For a more complete description of the Interlocked Read/Unlock Write transaction sequence, please refer to the Laser System Specification.

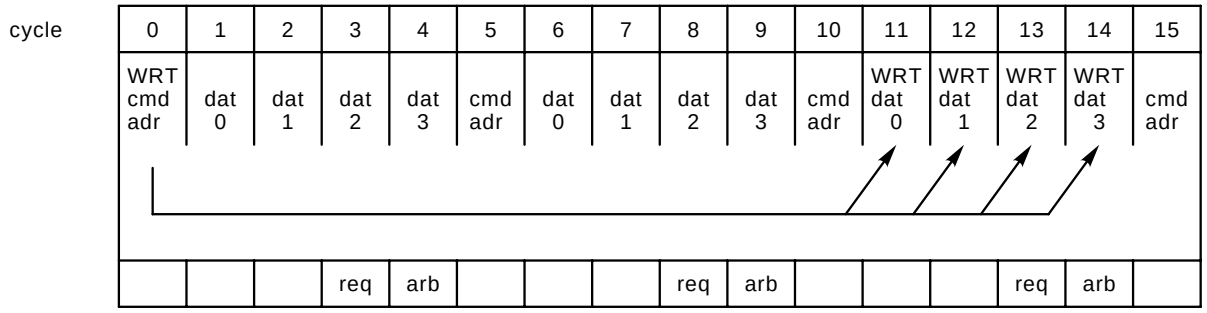
NOTE

There is no support for interlocked commands on the Futurebus+.

3.2.3 DMA Unmasked Write Transactions

The IOP supports unmasked double hexaword Writes to memory. Unmasked Writes map directly to block (64-byte) Writes on the LSB. Figure 3–5 shows a typical block Write on the LSB. The Write command/address is asserted on the LSB in cycle 0. The next 10 bus cycles are reserved for data cycles from previous transactions and for the next two transactions' command/addresses. The IOP drives Write data during cycles 11 through 14.

Figure 3–5: DMA Unmasked Write Example



DMA_UNMASKED_WRITE_EXAMPLE

3.2.4 DMA Masked Write Transactions

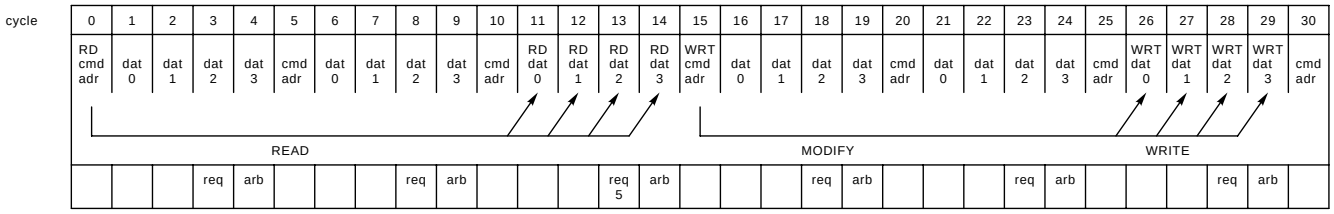
The IOP supports masked octaword, masked hexaword and masked double hexaword Writes to memory. The READ/MODIFY/WRITE sequence executed by the IOP for DMA Masked Write transactions is an atomic operation between the IOP and LSB memory.

This operation is guaranteed to be atomic due to the LSB memory architecture and bus arbitration protocol. Because of the memory bank access rule specified in Section 3.3 and the IOP's use of the highest arbitration level to get on the LSB to put the modified data back in memory, it is impossible for another node (i.e. a cpu) to touch that memory location anytime between the IOP's Read of the memory location and the completion of the Write back to memory. The IOP's Read command "ties up" that memory location for the first fifteen cycles while the IOP's *guaranteed* Write command immediately following the Read command ties up the same location for the remaining fifteen cycles. The IOP is the only LSB node capable of arbitrating at the highest priority level, REQ<5>.

Figure 3–6 shows a Masked Write as seen on the LSB. All transaction lengths will look the same. The IOP sends a Read command in cycle 0 with the address of the location to be written. In cycles 11 through 14, the memory (or cache) supplies the Read data. During cycle 13 while the Read data is still coming across the LSB, the IOP requests the bus using the REQ<5> request line, thereby guaranteeing bus ownership for the command/address cycle in cycle 15. The Write command/address is asserted on the bus during cycle 15. The IOP has until cycle 25 to perform a merge of the Read data with the Masked Write data. The

masked operation is completed by transmitting data to memory during cycles 26 through 29.

Figure 3–6: DMA Masked Write Example



DMA_MASKED_WRITE_EXAMPLE

Table 3–5 summarizes the types of Writes from the I/O Bus Adapters supported by the IOP, and the corresponding LSB transaction(s) performed in response to the HOSE Write packet.

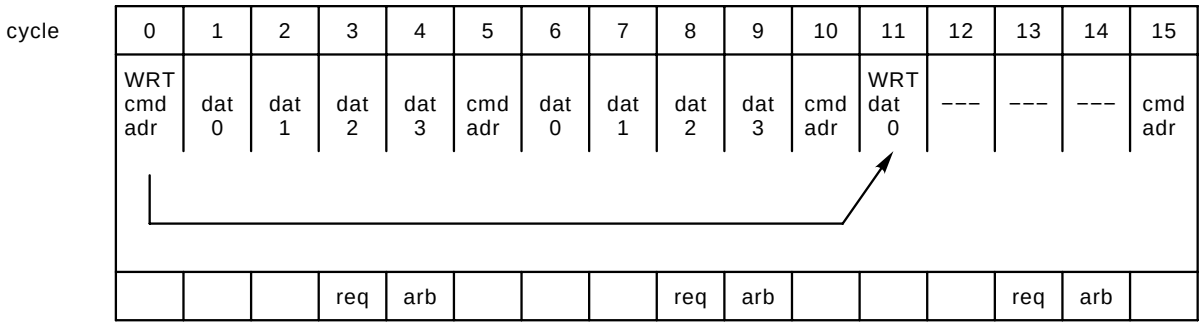
Table 3–5: Write Types

Transaction Length	Type	LSB Transaction(s)
octaword	masked	Read (modify) then Write
hexaword	masked	Read (modify) then Write
double hexaword	masked	Read (modify) then Write
double hexaword	unmasked	Write

3.2.5 Interrupt Transactions

The IOP uses Write CSR transactions to perform the interrupt function. First, the IOP looks up the CPU interrupt mask bits stored in the LCPUMASK register. By combining the desired interrupt level with the bits that are set in the LCPUMASK register, an interrupt destination mask is created. The IOP arbitrates for access to the LSB, then does the Write CSR to the CPU(s). Figure 3–7 shows the IOP writing the LIOINTR register in Broadcast Space. The Write CSR command and the LIOINTR address are driven during cycle 0. The interrupt destination mask (CPU mask and interrupt level) is driven during cycle 11. Figure 3–8 shows the format of the first data cycle in cycle 11. Cycles 12 through 14 are not driven, thereby defaulting to bad parity and uncorrectable ECC.

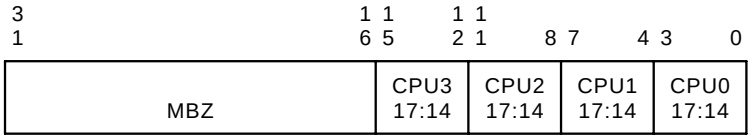
Figure 3–7: Write CSR Transaction



––– = UNDRIVEN CYCLES

WRITE_CSR_EXAMPLE

Figure 3–8: Write CSR (Interrupt) Data Format



Each 4−bit field in the lower word of data represents the four interrupt levels possible for the indicated CPU.

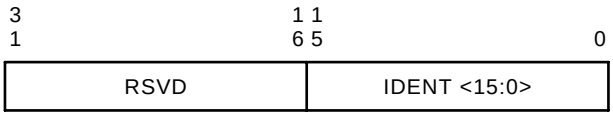
INTERRUPT_DATA_FORMAT

CPUs monitor Write CSR transactions to the LIOINTR to detect when they have been targeted by an interrupt. The IOP may send up to 4 interrupts at the same IPL (one from each HOSE) to a given CPU. These interrupts at the same IPL for one CPU would occur over separate Writes to the LIOINTR. When the CPU(s) is ready to service the interrupt, it performs a Read CSR of the LILIDx register in the IOP. The IOP returns the contents of the appropriate LILIDx register as shown in Figure 3–9.

NOTE

The "x" in LILIDx corresponds to the register number that serves a given interrupt priority level (IPL). LILID0 holds the IDENT vectors for IPL14 interrupts, LILID1 holds the IDENT vectors for IPL15 interrupts, and so forth.

Figure 3–9: Laser Interrupt Levelx Ident Register Format



There are two classes of interrupts from the IOP's perspective. The first class of interrupts includes those that are generated by devices external to the IOP, i.e. LAMB, FLAG, or I/O device. The second class of interrupts includes those generated by the IOP itself. The latter class serves to indicate error conditions detected by the IOP. Refer to the relevant I/O adapter specification for details on the handling of interrupts from I/O devices and from errors detected by the I/O adapters.

The LILID0, LILID1 and LILID2 registers are each actually a queue of 4 possible pending interrupts at IPL14, IPL15 and IPL16 respectively (one interrupt per hose per IPL). The vectors are returned to the CPUs in the order in which the interrupts arrived. The LILID3 register queue is actually five deep, one IPL17 interrupt per hose plus one internally generated IOP IPL17 error interrupt.

The IOP generates a high-level interrupt (IPL17) when it detects an error condition. This interrupt may be either enabled or disabled by setting or clearing the "INTR_NSES" bit in the IOP's IPCNSE Register. The IOP's interrupt is the most important possible within the I/O system and shall therefore take precedence over other IPL17 interrupts. The way this is implemented is by the IOP always returning *its* IDENT vector first when a cpu reads the LILID3 register, even though there may be "older" interrupts at IPL17 pending. The IOP shall make sure that it posts its interrupt as soon as possible.

It should be noted that at IPL 14, IPL15 and IPL16 the IOP will never post more than 4 interrupts to the CPUs at a given IPL (one per hose). The IOP, however, could post up to 5 interrupts to the CPUs at IPL17 (one per hose plus one IOP generated error interrupt).

When a CPU reads a specific LILIDx register that contains a valid vector the IOP builds an Interrupt Status Return Packet and returns it on the appropriate Down Hose to the I/O Adapter Module (I.E. LAMB or FLAG). The one exception to this rule is an IOP internally generated error interrupt. For this case no Interrupt Status Return Packet is required.

When a CPU reads a specific LILIDx register that does not contain a valid vector the IOP returns "0" (zero) to the CPU and takes no further action.

3.2.6 Mailbox Transactions

When a CPU successfully writes to the IPC's LMBPR register using a Write CSR command, the IPC begins a mailbox transaction. A mailbox is a *naturally aligned*, 64-byte data structure in memory that contains a CPU's request for access to a remote I/O bus. The mailbox format is as shown in Figure 3–10. The first 32 bytes of information are written by the CPU to tell the IPC which I/O bus is to be the recipient of the mailbox command, and to tell the remote bus adapter what transaction is to be performed and what the Write data is, if any.

The second 32 bytes of data are reserved for the completion status of the mailbox transaction. Read data, if any, is deposited back into memory as is error indication, a "DONE" flag, and device dependent completion codes. The IPC serves as an intermediary and forwards the request to the target I/O bus.

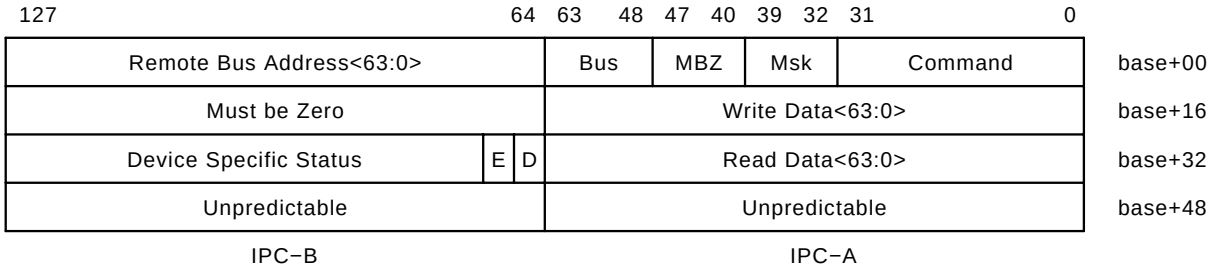
The IOP implements the LMBPR as a set of eight registers, two per cpu. These registers that comprise the LMBPR are serviced in strict first-come, first-served order. Any other Writes to the LMBPR by a CPU which already has two mailbox transactions pending will be NOACKed until at least one mailbox transaction from that CPU has completed. A further constraint is that only one mailbox transaction may be processed by the IOP at any time. All further mailbox transactions in the LMBPR "queue" will be put on hold until the previous mailbox transaction completes. Completion is signaled by a mailbox status packet being returned to the IOP by an I/O bus adapter.

When a CPU loads a Mailbox Pointer Register (LMBPR) in the IOP the IOP reads the Mailbox structure from Laser memory and transmits it on the Down Hose as a Mailbox Command packet.

Note that the IOP forces bits 13 and 12 of the Mailbox Command field to 10 binary, which is the Hose code for a Mailbox Command Packet. Therefore, these bits have no meaning to software. This is the only modification the IOP makes to the Mailbox Command Packet before transmitting it on the Down Hose.

Eventually, the IOP should receive a Mailbox Status Return Packet on the targeted Up Hose, which indicates that the Mailbox Command Packet has completed. Upon receiving the Mailbox Status Return packet the IOP executes a READ/MODIFY/WRITE transaction on the Laser bus to fetch the Mailbox structure, merge the information from the Mailbox Status Return packet, and write the results back to main memory. Section 3.2.4.

Figure 3–10: Mailbox Data Structure



NOTE 1: This figure shows the IOP's view of the Mailbox data structure. Also note that this figure is based on the 128−bit width of the LSB, not the 64−bit width handled by software.

NOTE 2: Bits 13 and 12 of the Mailbox Command field are always forced to 10 binary by the IOP, which is the code for a Mailbox Command Packet

MAILBOX_DATA_STRUCTURE

3.2.7 IOP as Responder

CPUs may access the IOP's internal registers via Read CSR and Write CSR transactions. The IOP's responder logic uses the transaction address to select the appropriate IPC register. The LSB command determines whether data is written into the selected IPC register or whether the register's contents are queued for return to the CPU that made the request. There is also a special class of transactions that is initiated by a CPU's Write CSR transaction to the IOP's LMBPR, the mailbox pointer register. This was previously discussed in Section 3.2.6.

3.3 Memory Access Rule

An LSB node is not permitted to arbitrate for memory access when its memory access would conflict with another access to the same memory bank, i.e. if an access to the same memory bank has not yet completed.

LSB memory is divided into banks. The theoretical maximum number of banks is 64; up to 8 banks per memory module on 8 modules. In practice, the number of banks will be less because of the limited number of slots in the system. Each cpu module takes away a slot that could potentially have been a memory module. The LMMRx registers (where x is from 0 to 7) in conjunction with memory bank mapping logic provide sufficient information to determine a unique memory bank number from byte address<39:5>. Section 3.3.1 contains a copy of the LSB Specification's DECSIM BDS code that describes the LSB memory bank algorithm.

Individual Laser Memory banks shall be accessed at most once every 15 LSB cycles. Since all transactions take one command/address cycle and four data cycles, LSB transactions are interleaved to make full use of the bus bandwidth. In order to meet the memory access restriction mentioned above, each potential bus master node (CPUs and IOP) must monitor bus transactions and keep track of the last two memory banks that were accessed.

3.3.1 Memory Bank Algorithm

A flexible memory interleave scheme has been implemented on the LSB. Such flexibility has a correspondingly complex implementation penalty. The DECSIM BDS code below, taken from the LSB Specification, describes the LSB memory bank scheme.

```
BANK<5:0> = 0;                          ! (BANK = OR of 8 inputs)

FOR N FROM 0 TO 7 DO BEGIN              ! For each of 8 LMMRs

          ! Names for LMMR bits

    MODULE_ENABLED[N] = LMMR[N]<0>;             ! EN
    MODULE_ADDRESS[N]<14:0> = LMMR[N]<31:17>;   !
    INTRLV_ADDRESS[N]<1:0> = LMMR[N]<4:3>;      ! IA
    INTRLV_WIDTH[N]<1:0> = LMMR[N]<2:1>;        ! INT

          ! Decoded mask fields from LMMR

    MODULE_ADDRESS_MASK[N]<14:0> =              ! AW
           000000000000000#2 SR1 LMMR[N]<8:5>;

    INTRLV_ADDRESS_MASK[N]<1:0> =               ! INT
           000#2 SL1 INTRLV_WIDTH[N]<1:0>;

    BANK_MASK[N]<2:0> =                         ! NBANKS
           000#2 SL1 LMMR[N]<10:9>

          ! 3 least sig bits of LMMR address

    LMMR_NUMBER[N]<2:0> = N;

          ! selected if enabled and high bits match
          ! and interleave matches
          ! (If LMMR's are properly programmed, only one bit
          ! in BANK_SELECTED[N] will be set.)

    MODULE_SELECTED[N] = MODULE_ENABLED[N]
                         AND
        ((A<34:19> AND MODULE_ADDRESS_MASK[N]<14:0>) EQL
                MODULE_ADDRESS[N]<14:0>)
                         AND
        ((A<2:1> AND INTRLV_ADDRESS_MASK[N]<1:0>) EQL
                INTRLV_ADDRESS[N]<1:0>);

          ! which of the 1, 2, 4, or 8 banks on the module?

    BANK_IN_MODULE[N]<2:0> =
        ((A<5:1> SR0 INTRLV_WIDTH[N]<1:0>) AND BANK_MASK[N]]<2:0>);

          ! OR the results onto BANK

    BANK<5:0> = BANK<5:0> OR
        ((MODULE_SELECTED[N] SXT 5#10) AND
        (LMMR_NUMBER[N]<2:0> & BANK_IN_MODULE[N]<2:0>));

END;
```

For purposes of this algorithm, all CSR accesses are treated as references to bank 0.

IMPLEMENTATION NOTE

Since all CSR accesses are treated as references to bank 0, the memory bank contention system effectively prevents interleaving of CSR accesses.

3.4 Bus Arbitration

Laser arbitration is performed over a set of six, fixed-priority lines. Fairness amongst the CPUs is achieved through dynamic reallocation of priority levels used. The CPUs use levels 1 through 4. The I/O node uses a different scheme from the CPUs. The IPC has available to it both the lowest (priority 0) and the highest (priority 5) priority levels. In typical operation, the IPC shall use high Priority Level 5. After a predetermined number (default=1) of consecutive accesses to a given memory bank, the IPC shall switch to low Priority Level 0 when going after the same memory bank. This guarantees that CPUs will not be locked out from memory. Through the round robin scheme, the CPUs will all eventually gain access to a given memory bank.

In theory, the IPC is allowed to use Priority Level 5 exclusively until it has accessed all memory banks, even if these transactions occur back-to-back. In practice, however, the IPC will not be able to sustain such a bandwidth and will therefore leave many transaction opportunities open to the CPU nodes.

Several mode-selectable lockout avoidance algorithms are implemented which guarantee that the IOP will eventually allow other nodes to access a given memory bank on the LSB, while allowing software to fine-tune I/O performance. See Table 3–6. A CPU node is deemed "locked out" if it cannot access a given memory bank for a long period of time. The default mode of three high priority arbitration opportunities followed by a single low priority arbitration attempt shall guarantee correct LSB operation with optimal I/O performance. The Write portion of a READ/MODIFY/WRITE operation shall *always* be performed at high priority to guarantee atomicity of the transaction pair. The other IPCMSR selectable arbitration modes permit fine-tuning of IOP performance relative to the other bus traffic. There is also a fixed high priority mode. The IOP does not normally operate in this fixed high priority mode unless the system software explicitly programs the IOP to do so via bit<0> of the IPCMSR.

Table 3–6: Arbitration Mode Selection Portion of the IPCMSR

Name	Bit(s)	Type	Init	Description
ARB_CTL<1:0>	2:1	R/W	0	ARBITRATION CONTROL These bits can be programmed to select the number of times the I/O Port Module arbitrates at high priority before switching to low priority arbitration for up to 2 requests. Note that only the request for the READ portion of a READ/MODIFY/WRITE transaction is counted as a request in this algorithm. The request to perform the WRITE portion of the READ/MODIFY /WRITE must *ALWAYS* use high priority arbitration (REQ 5) and is not counted in the back off algorithm.

Table 3–6 (Cont.): Arbitration Mode Selection Portion of the IPCMSR

Name	Bit(s)	Type	Init	Description
				The ARB_CTL field is encoded as follows
			00	Arbs six times at high priority, then switches to low priority arbitration for up to 2 requests. This is the DEFAULT at power up or initialization.
			01	Arbs four times at high priority, then switches to low priority arbitration for up to 2 requests.
			10	Arbs twice at high priority, then switches to low priority arbitration for up to 2 requests.
			11	Arbs at low priority always.
				It is *STRONGLY* recommended that 11 (arb low always) *NOT* be selected during normal operation as I/O performance will be adversely affected.
ARB_HIGH	0	R/W	0	This bit when set to a 1, forces the I/O Port Module to always arbitrate at high priority (REQ 5). When set, this bit overrides the ARB_CTL field.
				It is *STRONGLY* recommended that the ARB_HIGH bit *NOT* be used during normal operation as potential lockouts from memory could occur.

In general, READ/MODIFY/WRITEs are costly operations to perform and require attention to be paid to cache coherency, however, in order to avoid the latter concern, the IPC shall always use Priority Level 5 to gain bus access on the Write portion of the operation. That makes the two non-overlapped, back-to-back transactions (a Read followed by a Write) an atomic operation and cache coherence is maintained without any extra logic.

3.5 Error Detection Schemes

The LSB uses ECC protection for all memory data cycles, odd parity protection on all command/address cycles and CSR data cycles, and transmit check logic by all nodes that are responsible for driving the bus. More detail is provided in Chapter 8.

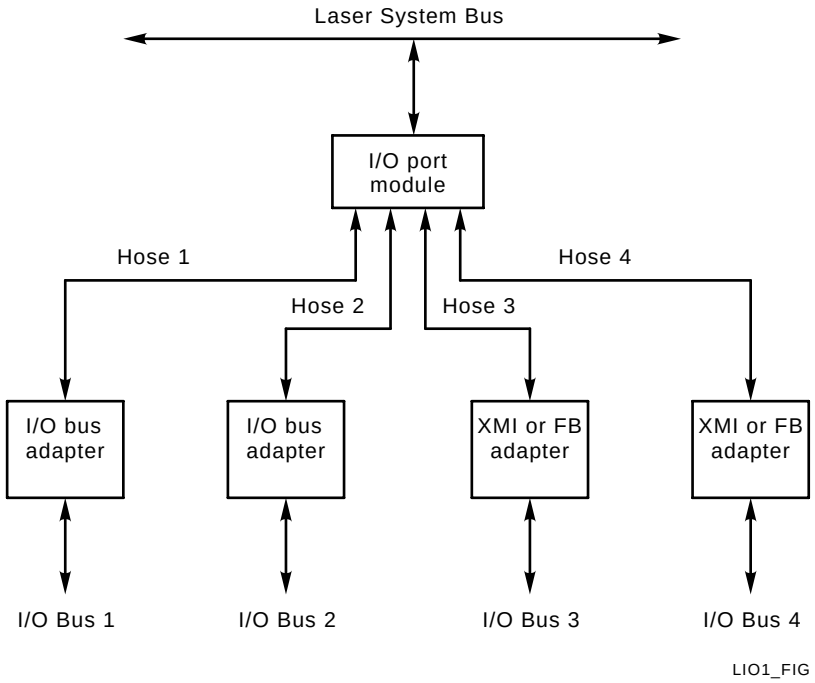
CHAPTER 4

HOSE/IOP INTERFACE

4.1 Introduction

The Laser I/O subsystem architecture has four separate I/O bus adapters that communicate with the IOP over four dual-cable buses. See Figure 4–1. This chapter describes the IOP to I/O adapter bus, called the *"HOSE"*.

Figure 4–1: Laser I/O Subsystem



Each HOSE consists of two separate unidirectional interconnects, a **DOWN HOSE** which transmits from the IOP to the I/O bus adapter module, and an **UP HOSE**, which transmits from the I/O bus adapter module to the IOP.

The DOWN HOSE is synchronous with the IOP and transmits a longword (32 bits) of data every other Laser System Bus clock cycle, (i.e. DNCLK = ½ the LSB clock frequency). The raw data bandwidth of the DOWN HOSE is 100-125 Mbytes/sec.

The UP HOSE is synchronous with the I/O bus adapter module, which is currently planned to be an XMI adapter (LAMB) module or a Futurebus+ adapter (FLAG) Module. The transmit cycle time for the UP HOSE of the LAMB module is half the 64 nsec XMI cycle time, or 32 nsec, resulting in a raw data bandwidth of 125 Mbytes/sec. The FLAG module transmit cycle time for the UP HOSE has a specified goal of 25ns.

4.1.1 HOSE Protocol Description

In the Laser system, the HOSE connects the IOP with one of four I/O bus adapter modules. The physical distance between these can be as great as 3 meters, so the HOSE can be up to 3 meters long. Since signals take around 20 ns to propagate from one end of the HOSE to another, acknowledging the arrival of data on a per-cycle basis would take *at least* the round trip propagation time, which is 40 ns, per data unit (longword).

Acknowledging each longword would require too many transitions across a synchronization boundary to keep up with the HOSE data transfer rate, so separate flow control methods for both the UP HOSE and the DOWN HOSE have been devised to take advantage of each of their strengths.

On the DOWN HOSE, the IOP can **always** send a packet to the I/O Adapter module without fear of overflowing the I/O Adapter's FIFO. This is because the flow control is maintained by the I/O Adapter itself. There are 3 packet types that can be transmitted across the DOWN HOSE. They are MAILBOX Command, DMA Read Return Data, and INTR/IDENT Return status packets. The MAILBOX Command packet is issued by a CPU, but the IOP assures that only 1 MAILBOX Command packet can be issued down the hose at a time. DMA Read Return Data and INTR/IDENT Return status packets are a result of DMA Read and INTR /IDENT packets issued by the I/O Adapter across the UP HOSE. Therefore, as long as the I/O Adapter has room in it's DOWN HOSE FIFO to store 1 MAILBOX Command packet, and the number of "return" packets based on the number of DMA Read and INTR/IDENT packets it has sent across the UP HOSE, the DOWN HOSE FIFO will never overflow.

A summary of this protocol is that the I/O Adapter module must allocate a DOWN HOSE buffer before transmitting a DMA Read or INTR/IDENT packet across the UP HOSE. It must also always maintain a spare buffer to receive a MAILBOX Command packet across the DOWN HOSE from a CPU.

The advantage of this DOWN HOSE protocol is that the IOP can transmit packets across the DOWN HOSE as fast as it is capable, thus maximizing the overall DOWN HOSE performance. This protocol also has an ancillary benefit in that it prevents one DOWN HOSE FIFO from backing up into the IOP's DOWN FIFO and thus shutting down all four DOWN HOSES. This was a problem for other flow control methods that were considered that allowed the I/O Adapter to stop the IOP from transmitting DOWN HOSE packets when its FIFO was full.

The UP HOSE is flow controlled via the Decrement Packet Count (DECPKTCNT) signal. The I/O bus adapter module keeps count of how many packets it has transmitted across the UP HOSE at any given time. The I/O bus adapter can send up to 4 packets before the IOP buffers will be filled. Each time the IOP removes an UP HOSE packet from its buffer, it will assert DECPKTCNT for one DOWN HOSE cycle. The I/O adapter module will use this

signal to decrement its packet counter so that it can keep a running count of how many free buffers there are in the IOP at any given time. As long as the count is less then 4, the I/O bus adapter module can transmit an UP HOSE packet.

The advantage of this UP HOSE protocol is that it avoids the round trip delay of acknowledeging each packet before the next one can be sent.

In most cases, the IOP will be able to unload its buffers faster than they can be loaded by the I/O Adapter module across the UP HOSE. Therefore the IOP buffers will only become full if the LSB is stalled, or too many bank conflicts prohibit the IOP from generating transactions on the LSB as fast as they are received across the four UP HOSEs. As the LSB frees up, the UP HOSE packets will begin to flow again causing the IOP to assert DECPKTCNT, allowing the I/O bus adapter to start transmitting UP HOSE packets again.

On both the UP HOSE and DOWN HOSE, once a packet transmission is started, longwords must be transmitted over the HOSE contiguously, until the last longword of the packet has been transmitted. The deassertion of the Data Valid signal signifies the end of the packet. "Idle cycles" or any interruption of the data flow that occurs before the end of the packet will result in a sequence error for that packet.

4.2 HOSE Signals

4.2.1 Assertion Levels

Many of the tables and in this chapter show the function of a signal when it is a 1 and/or when it is a 0. In all cases, a one (1) simply means the the signal is asserted, or TRUE. In all cases, a zero (0) simply means the the signal is deasserted, or FALSE.

a. For a HIGH true signal this means that:

- 1 = asserted = TRUE = HIGH = +5 volts
- 0 = deasserted = FALSE = LOW = 0 volts

b. For a LOW true signal it means that:

- 1 = asserted = TRUE = LOW = 0 volts
- 0 = deasserted = FALSE = HIGH = +5 volts

This convention is used throughout the chapter.

Table 4–1 lists the signals of the UP HOSE and DOWN HOSE. The following two sections describe each of the hose signals.

Table 4–1: Hose Signal List

DOWN HOSE	Signal Count	Synchronous to	UP HOSE	Signal Count	Synchronous to
DND<31:0> L	32	DNCLK	UPD<31:0> L	32	UPCLK
DNP L	1	DNCLK	UPP L	1	UPCLK
DNDATAVAL L	1	DNCLK	UPDATAVAL L	1	UPCLK
DNCLK H	1	—	UPCLK H	1	—
DECPKTCNT L	1	DNCLK	UPCTL<3:0> L	4	UPCLK
DNRST L	1	Asynchronous	UPRST L	1	Asynchronous
ERROR L	1	Asynchronous	CBLOK L	1	Asynchronous
			PWROK H	1	Asynchronous
DOWN HOSE Total	38		UP HOSE Total	42	

4.2.2 DOWN HOSE

4.2.2.1 DND<31:0> L

Down Data Lines - These lines are asserted by the IOP and can be Command/Address, Data or Transaction Status information.

4.2.2.2 DNP L

Down Parity - This line contains the parity information for the Down Hose. DNP provides odd parity across DND<31:0>. Down Hose parity should be checked by the I/O Adapter module every Down Hose cycle that "DNDATAVAL L" is asserted.

NOTE

The I/O bus adapter module should ignore DNP if "ERROR L" is asserted.

4.2.2.3 DNDATAVAL L

Down Data Valid - This line is asserted by the IOP for each valid cycle of a Down Hose packet. The I/O bus adapter module should latch DND<31:0> and check parity during DOWN HOSE cycles that have DNDATAVAL asserted. There is one (1) hose cycle minimum spacing between DOWN HOSE packets to allow DNDATAVAL to deassert at the end of the first packet and assert at the start of the next packet. This allows the I/O bus adapter module to check the packet for correct length (i.e. sequence error).

NOTE

The I/O bus adapter module should ignore DNDATAVAL if "ERROR L" is asserted.

4.2.2.4 DNCLK H

Down Clock - This wire is the clock sent by the IOP to the I/O bus adapter module. DNCLK is a square wave running at ½ the LSB clock frequency. The rising edge of this line is used to clock the data on the Down Data, Down control, and Down Parity Lines, when the Down Data Valid line is asserted. The Down Clock is always running, and can be used as a clock for state machines in the DOWN HOSE interconnect in the I/O bus adapter.

4.2.2.5 DNRST L

Down Reset - This line is asserted by the IOP for 128ns duration to reset the logic of the I/O bus adapter module. The IOP asserts DNRST to **all** four hoses whenever LSB Reset is asserted or whenever the IOP's Node Reset bit (LCNR<30>) is set. The IOP can also be programmed to assert DNRST to a particular hose by setting the appropiate HOSE_RST<3:0> bit in the IPCMSR register. HOSE_RST<3:0> correspond to hoses 3 to 0 respectively.
The assertion of DNRST must return the I/O bus to its reset state. The definition of reset state is I/O bus specific.

NOTE

The I/O bus adapter module should always attempt to respond to DNRST, even of "ERROR L" is asserted.

4.2.2.6 ERROR L

Error - This signal has a pullup resistor on both the IOP and I/O bus adapter, and is a wired-OR signal that can driven low by both the IOP or I/O bus adapter module.

In normal operation, ERROR is asserted by the I/O Adapter module whenever it encounters a non-recoverable error (fatal error). The I/O Adapter module then enters a quiesent state and ignores further DOWN HOSE transactions and will not generate any more UP HOSE transactions. The I/O Adapter module must also disable checking of DOWN HOSE parity when ERROR is asserted. The assertion of ERROR causes the IOP module to generate an error interrupt to the CPU(s), if interrupts are enabled.

During IOP internal diagnostic Loopback testing, the IOP asserts ERROR to effectively disable the I/O adapter module from decoding DOWN HOSE packets, checking DOWN HOSE parity, or transmitting UP HOSE packets while the IOP is performing internal loopback diagnostics.

ERROR is used in conjunction with CBLOK and PWROK to indicate the status of the hose (Refer to Table 4–3 of this chapter).

4.2.2.7 DECPKTCNT L

Decrement Packet Count - This line is asserted by the IOP to the I/O bus adapter module to indicate that an UP HOSE packet has been removed from the IOP's buffer. The IOP will assert this signal for one DOWN HOSE cycle each time it removes an UP HOSE packet from its buffers or if the UP HOSE packet contained an error and was never loaded into the IOP's buffer. The I/O bus adapter module should use DECPKTCNT to keep an running count of how many free buffers there are in the IOP at any given time. Once the count reaches 4, the IOP buffers are full, and no more UP HOSE packets can be generated until the IOP asserts DECPKTCNT. DECPKTCNT can be asserted in consecutive back-to-back DOWN HOSE cycles (e.g. DECPKTCNT could be asserted for two or more consecutive cycles if IOP buffers were unloaded fast enough to cause back-to-back assertions of DECPKTCNT).

NOTE

The I/O bus adapter module should ignore DECPKTCNT if "ERROR L" is asserted.

NOTE

When tracking the number of available buffers in the IOP, the I/O bus adapter should increment its packet counter as soon as it starts transmitting the packet across the UP HOSE. The I/O bus adapter should NOT wait until it's done transmitting the packet. If the I/O bus adapter were to wait until the last cycle of the packet, and the HIC were to detect and error on the first cycle of the packet, the IOP could assert DECPKTCNT to the I/O bus adapter before the I/O bus adapter had time to finish transmitting the UP HOSE packet and incrementing its packet counter. This would result in a temporary underflow of the packet counter.

4.2.3 UP HOSE

4.2.3.1 UPD<31:0> L

Up Data Lines - These lines are asserted by the I/O bus adapter modules and can be Command/Address, Data or Transaction Status information.

4.2.3.2 UPP L

Up Parity - This line contains the parity information for the UP HOSE. UPP provides odd parity across UPD<31:0> and UPCTL<3:0>. UP HOSE parity will be checked by the IOP every UP HOSE cycle that "UPDATAVAL L" is asserted.

4.2.3.3 UPCTL<3:0> L

Up Control Lines - These lines indicate the Packet Type during the first HOSE cycle of a packet and the mask bits during data cycles of a DMA Masked Write packet. The codes for the packet types and mask bits are:

Table 4–2: UPCTL<3:0> Codes

	meaning during first HOSE cycle of a packet
UPCTL<3:0>	Packet Type[1]
0001	DMA Read
0010	IRead
0100	MAILBOX Status Return
0101	DMA Unmasked Write w/data
0111	DMA Masked Write w/data
1000	INTR/IDENT

	meaning during Data cycles of a DMA Masked Write packet
UPCTL<3:0>	Mask[2]
0001	UPD<7:0> are valid
0010	UPD<15:8> are valid
0100	UPD<23:16> are valid
1000	UPD<31:24> are valid
1111	UPD<31:0> are valid

[1]All other UPCTL<3:0> Packet Type codes are reserved and will cause an Illegal Packet Error when detected by the IOP

[2]Any combination of mask bits can be used. The table simply attempts to show the relationship between UPCTL<3:0> bits and the bytes they mask

4.2.3.4 UPDATAVAL L

Up Data Valid - This line is asserted by the I/O Adapter module for each valid cycle of an Up Hose packet. The IOP will latch UPD<31:0> and UPCTL<3:0>, and check parity during UP HOSE cycles that have UPDATAVAL asserted. There is one (1) hose cycle minimum spacing between UP HOSE packets to allow UPDATAVAL to deassert at the end of the first packet and assert at the start of the next packet. This allows the IOP to check the packet for correct length (i.e. sequence error).

NOTE

The I/O bus adapter module should not generate any UP HOSE packets if "ERROR L" is asserted. (i.e. the I/O bus adapter module should not assert UPDATAVAL)

4.2.3.5 UPCLK H

Up Clock - This wire is the clock sent by the I/O bus adapter module to the IOP. The rising edge of this line is used to clock the data on the Up Data Lines and the Up Parity Line, when the Up Data Valid line is asserted. The Up Clock is always running, and can be used as a clock for state machines in the UP HOSE interconnect in the IOP.

4.2.3.6 UPRST L

Up Reset - This signal has a pullup on the IOP and if asserted by the I/O bus adapter, will cause CCL RESET to be cycled, thus causing the entire Laser system to be reset. This allows remote boot capability. Refer to the CCS (Cabinet Control System) Specification for more details on how CCL RESET provides a system-wide reset request.

4.2.3.7 CBLOK L

Cable OK - This signal has a pullup resistor on the IOP and is grounded by the I/O Adapter module. If CBLOK is deasserted, then either the I/O Adapter module is not installed, or the HOSE cable is not connected.

CBLOK is used in conjunction with ERROR and PWROK to indicate the status of the hose (Refer to Table 4–3).

4.2.3.8 PWROK H

I/O Adapter Power OK - This signal has a pulldown resistor on the IOP and needs to be driven high (asserted) by the I/O Adapter module. The assertion of PWROK indicates that the I/O Adapter is "ready" to try to receive and process packets. The I/O Adapter deasserts PWROK whenever it has detected an I/O bus power failure. The transition of PWROK from high-to-low or low-to-high causes the IOP module to generate an error interrupt to the CPU(s), if interrupts are enabled.

PWROK is used in conjunction with CBLOK and ERROR to indicate the status of the hose (Refer to Table 4–3).

Table 4–3: Hose Status Signals

The IOP module will interrupt the CPU(s), if interrupts are enabled, on the following transitions			
CBLOK L	PWROK H	ERROR L	Meaning
X	L -> H	X	I/O Adapter just finished powering up - Adapter "Ready" to try to receive and process packets. IOP generates interrupt to CPU(s) on transition of PWROK
X	H -> L	X	I/O Adapter detected power failure. IOP generates interrupt to CPU(s) on transition of PWROK
L	H	H -> L	I/O Adapter detected a Fatal Error. I/O Adapter enters quiesent state and ignores further HOSE traffic. IOP generates interrupt to CPU(s) on high-to-low transition of ERROR

Meaning of Hose Status Signals when read via the IOP's status register			
CBLOK L	PWROK H	ERROR L	Meaning
L	H	H	I/O Adapter ready - Adapter "Ready" to try to receive and process packets
H	X	X	HOSE cable is disconnected or bad, or I/O Adapter module is not plugged in
L	L	X	HOSE cable OK - No power on I/O adapter Module
L	H	L	I/O Adapter detected a fatal error or IOP is in internal diagnostic loopback mode

4.3 HOSE Packet Specifications

The packet types are different for each HOSE because the nature of information sent from I/O to memory (UP HOSE) is different than that transmitted from memory to I/O (DOWN HOSE).

Most of the data transferred by I/O to and from memory is actual data, for example: disk blocks, messages from terminals. Most of the data transferred by CPUs to and from I/O devices is control and status information.

As a result of the asymmetrical nature of the data transfer over the UP and DOWN HOSES, the packet types differ: The possible packets for the DOWN HOSE are:

1. Mailbox Command
2. DMA read data return
3. INTR/IDENT status return

The possible packets for the UP HOSE are:

1. Mailbox status return

2. DMA read
3. IRead
4. DMA masked write with data
5. DMA unmasked write with data
6. INTR/IDENT

4.3.1 DOWN HOSE Packet Specifications

During the first cycle of a DOWN HOSE packet, a "packet type" command field is encoded into DND<13:12> by the IOP. These bits indicate the type of packet being transmitted across the DOWN HOSE. The codes for the packet types are:

Table 4–4: DOWN HOSE Packet Type Codes

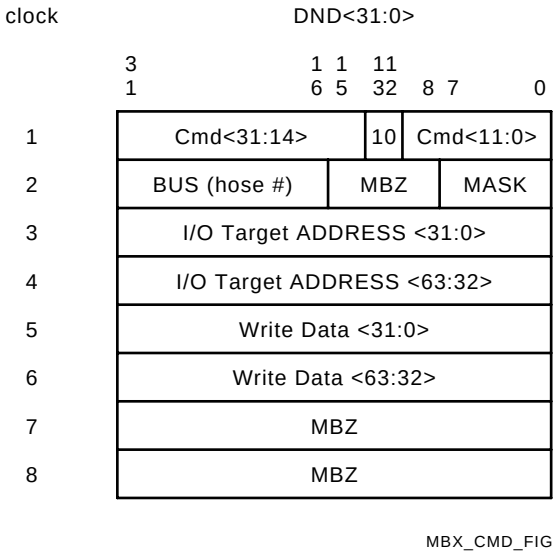
DND<13:12>[1]	Packet type
00	INTR/IDENT status Return
01	DMA Read Data Return
10	MAILBOX Cmd
11	reserved

[1]Encoding during first cycle of DOWN HOSE packet

4.3.1.1 MAILBOX Command

The MAILBOX Command packet is used by processors to access Control and Status Registers in adapters on the XMI or Futurebus+.

Figure 4–2: Mailbox Command Packet



The Command field (except Command<13:12>) is specific to the remote bus (e.g. XMI or Futurebus+), rather than the IOP and contains the remote bus operation. It can include fields such as read/write, address only, address width, data width, etc. Command<13:12> bits are forced by the IOP to indicate a Mailbox Command Packet (e.g. 2 hex).

The HOSE number indicates to which DOWN HOSE the MAILBOX Command packet is transmitted.

The MASK field provides the mask bits for the write data when the MAILBOX Command packet is a write. For MAILBOX Command packets that are reads, the MASK and Write Data fields are unpredictable.

The I/O address to be written is located in the I/O Target Address fields. 64 bits of address are supported by the DOWN HOSE, however the I/O bus adapter may or may not support this capability.

The MAILBOX Command packet supports a quadword data length write in the Write Data field, but the I/O bus adapter module might elect to only use the lower longword (Write Data<31:0>) if it only supports longword access to CSR space.

Status for the MAILBOX Command packet is returned in a separate packet on the UP HOSE called a MAILBOX Status Return packet.

Only one MAILBOX Command packet can be issued at a time by the IOP regardless which DOWN HOSE it is destined. A MAILBOX Status Return packet must be sent on the UP HOSE (for the currently outstanding MAILBOX command packet) before the IOP will issue another MAILBOX Command packet down any HOSE.

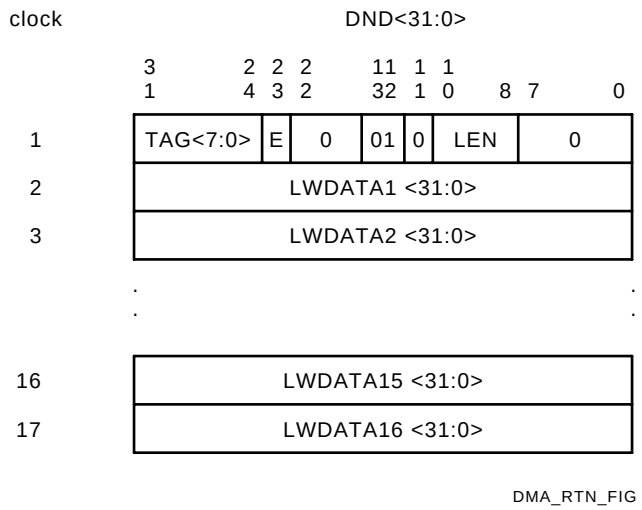
All MAILBOX Command packets that are writes can be byte masked. Any combination of mask bits is allowed by the DOWN HOSE, however the I/O bus adapter may or may not support this capability. The mask bits are Mask Disable bits as defined in the Alpha SRM

and therefore writing a 1 will disable the byte from being written. Refer to the Alpha SRM for more information on the MAILBOX structure in memory.

4.3.1.2 DMA Read Data Return

The DMA Read Data Return packet returns data previously requested by an UP HOSE DMA Read packet.

Figure 4–3: DMA Read Data Return Packet



A DMA Read Data Return packet has three possible packet lengths; Octaword, Hexaword, and Double-Hexaword. However, the Length code for the Octaword Packet may indicate that only a Longword or Quadword of data is needed. This length code is looped back from the initiating DMA Read packet and allows the I/O bus adapter module to use the length code to extract the correct amount of information from the octaword DMA Read Data Return packet.

NOTE

Even though the length code in an octaword packet may be for a longword or quadword, the parity across the HOSE must be good for all cycles of the DMA Read Data Return packet.

Table 4–5: DMA Read Data Return Packet Sizes

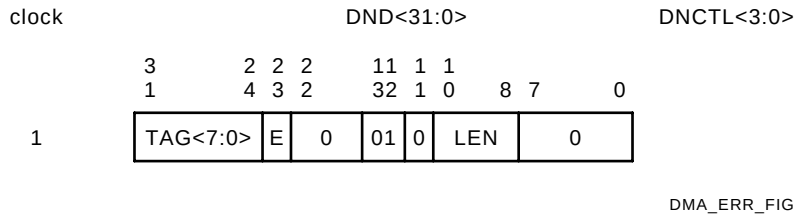
Length Code	Packet Data Length	Data Valid in Packet	Number of DOWN HOSE Cycles Required
001	Octaword	Longword	5
010	Octaword	Quadword	5
011	Octaword	Octaword	5
000	Hexaword	Hexaword	9
100	Double-Hexaword	Double-Hexaword	17

DND<13:12> of the first cycle of a DMA Read Return Packet is driven with an 01 by the IOP to indicate the packet type to the I/O Adapter module.

The TAG<7:0> field associates the DMA Read Data Return with the corresponding DMA Read packet on the UP HOSE. The TAG is generated by the I/O bus adapter and sent to the IOP as part of a DMA Read packet

If the IOP detects an error while trying to process a DMA Read packet, it will log the error and generate an error interrupt to the CPU(s), if interrupts are enabled. If the error condition doesn't prohibit it, a DMA Read Data Return packet with the error bit set will be sent across the DOWN HOSE. It will be 1 HOSE cycle long and is shown in Figure 4–4. Any UP HOSE or Vortex errors detected by the IOP will prevent the IOP from returning DMA Read Data Return packet. If an error occurs after the DMA Read packet has sucessfully made it to the IPC chips, a DMA Read Data Return packet with the error bit set **will** be returned across the DOWN HOSE.

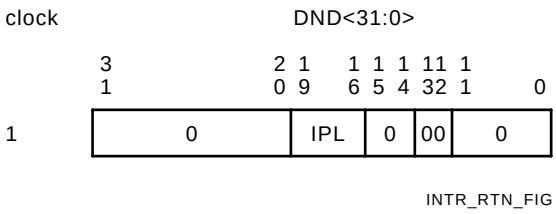
Figure 4–4: DMA Read Data Return PACKET With Error



4.3.1.3 INTR/IDENT Status Return

The INTR/IDENT Status Return packet returns the status for an INTR/IDENT packet previously transmitted on the UP HOSE. The INTR/IDENT Status Return packet is a flow control message. Receipt of an INTR/IDENT Status Return packet by an I/O bus adapter allows the I/O bus adapter to issue another INTR/IDENT packet to the IOP at the IPL returned in the INTR/IDENT Status Return packet.

Figure 4–5: INTR/IDENT Status Return Packet



DND<13:12> of a INTR/IDENT Status Return Packet is driven with a 00 by the IOP to indicate the packet type to the I/O Adapter module.

The IOP will drive zeros for bits <31:20>, <15:14>, and <11:0>.

The IPL field is the Interrupt Priority Level of the interrupt request. Bits <19:16> correspond to IPL17 - IPL14 respectively (refer to Figure 4–6). Only one IPL bit will be set per packet.

Figure 4–6: INTR/IDENT Status Return Packet bits <19:16>

INTR/IDENT Status Return Packet bits <19:16>)	
Bit<19>	IPL 17
Bit<18>	IPL 16
Bit<17>	IPL 15
Bit<16>	IPL 14

INTR_IPL_RTN_FIG

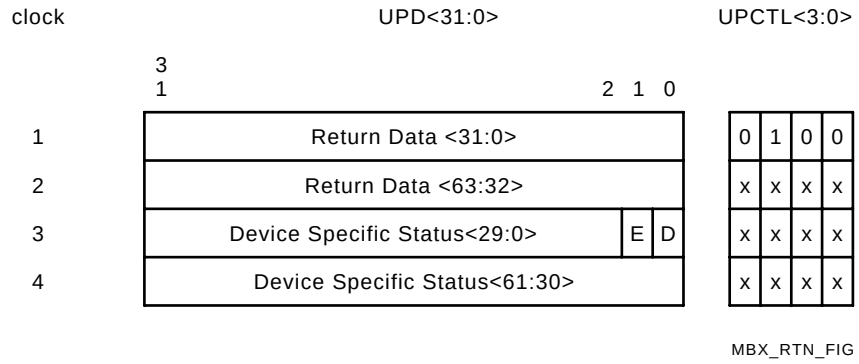
If the IOP detects an error while trying to process an INTR/IDENT packet, it will log the error and generate an error interrupt to the CPU. If the error condition doesn't prohibit it, an INTR/IDENT Status Return packet will be sent across the DOWN HOSE so the I/O bus adapter can still clear its INTR pending bit. Any HOSE or Vortex errors detected by the IOP will prevent the IOP from returning an INTR/IDENT Status Return packet. If the CSR write to the LIOINTR register in LSB broadcast space (in order to post the interrupt to the CPU) fails, or if the C/A cycle of the CPU read of the IOP's LILID register has a parity error, the IOP will NOT return an INTR/IDENT Status Return packet.

4.3.2 UP HOSE Packet Specifications

4.3.2.1 MAILBOX Status Return

The MAILBOX Status Return packet returns the status for a previously issued DOWN HOSE MAILBOX Command packet.

Figure 4–7: Mailbox Status Return Packet



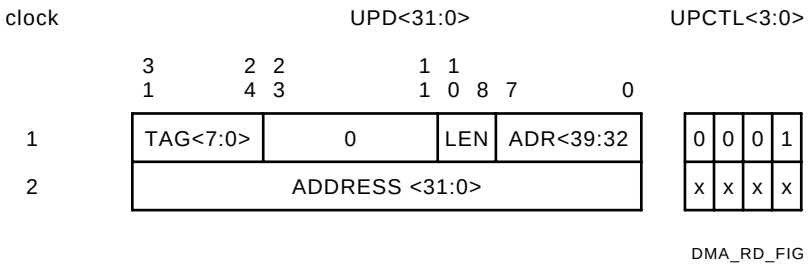
The Return Data fields contain read data in response to MAILBOX Command packets that were reads, and is unpredictable when responding to MAILBOX Command packets that were writes. The Done bit will always be set in a MAILBOX Return Packet. If an error was detected by the I/O adapter module, the Error bit will be set. The Device Specific field contains the operation completion status. The interpretation of this field is defined by the I/O bus adapter module.

The MAILBOX Status Return packet is used by the IOP to flow control MAILBOX Command packets. The IOP must receive a MAILBOX Status Return packet across the UP HOSE before it can issue the next MAILBOX Command packet across a DOWN HOSE.

4.3.2.2 DMA Read

The DMA Read packet is a request on the UP HOSE from the I/O bus adapter module to the IOP for a data read transaction on the Laser System Bus.

Figure 4–8: DMA Read Packet



A DMA Read packet has three possible packet lengths; Octaword, Hexaword, and Double-Hexaword. However, the Length code for the Octaword Packet may indicate that only a Longword or Quadword of data is requested. This length code is looped back through the DMA Read Data Return packet and allows the I/O bus adapter module to use the length code to extract the correct amount of information from the octaword DMA Read Data Return packet.

Table 4–6: DMA Read Packet Sizes

Length Code	Packet Data Length	Significant Address Bits[1]	Wrapped LSB Read	Data Requested
001	Octaword	ADR<39:4>	Yes, if ADR<5>=1	Longword
010	Octaword	ADR<39:4>	Yes, if ADR<5>=1	Quadword
011	Octaword	ADR<39:4>	Yes, if ADR<5>=1	Octaword
000	Hexaword	ADR<39:5>	Yes, if ADR<5>=1	Hexaword
100	Double-HW	ADR<39:6>	No	Double-HW

[1]ADR<3:0> bits are ignored by the IOP

ADR<39:0> is the target address for the Laser memory read and must be naturally aligned to length (LEN) code of the data being requested. ADR<39:5> is used by the IOP as the target address for the LSB Read. If ADR<5> is set, a wrap-around read occurs and the IOP receives the requested Hexaword first. When the length code indicates a Longword, Quadword, or Octaword, ADR<4> is used by the IOP to select which Octaword to return to the I/O Adapter module via a DMA Read Data Return packet.

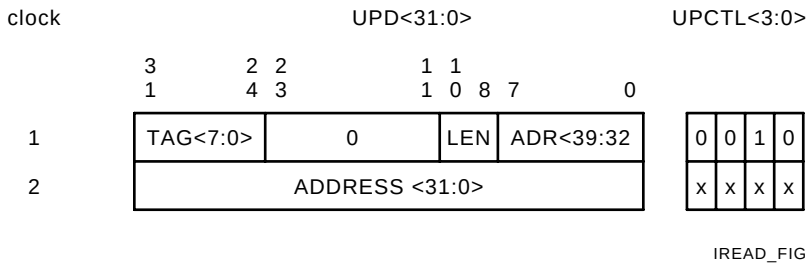
The TAG<7:0> field allows the subsequent DMA Read Data Return packet on the DOWN HOSE to be associated with a DMA Read Data packet on the UP HOSE. The TAG is generated by the I/O bus adapter.

The requested data for a DMA Read Packet is returned on the DOWN HOSE with a DMA Read Data Return packet.

4.3.2.3 IRead

The Interlock Read (IRead) packet is a request on the UP HOSE from the I/O bus adapter module to the IOP for a QW data read transaction on the Laser System Bus. The transaction works similar to a normal DMA Read except the IOP performs an atomic READ/MODIFY /WRITE operation to read the data, set bit 0 of the target QW, and write it back to LSB memory. This hardware assist helps software obtain an interlock on the QW location. The unaltered read data is returned across the DOWN HOSE using a DMA Read Data Return packet just as it would be for a normal read.

Figure 4–9: IRead Packet



The packet data length of an IRead packet will always be an Octaword and the Length code for the Octaword Packet will indicate that only a Quadword of data is requested. This length code is looped back through the DMA Read Data Return packet and allows the I/O bus adapter module to use the length code to extract the correct amount of information from the octaword DMA Read Data Return packet.

Table 4–7: IRead packet size

Length Code	Packet Data Length	Significant Address Bits[1]	Wrapped LSB Read	Data Requested
010	Octaword	ADR<39:3>	Yes, if ADR<5>=1	Quadword

[1]ADR<2:0> bits are ignored by the IOP

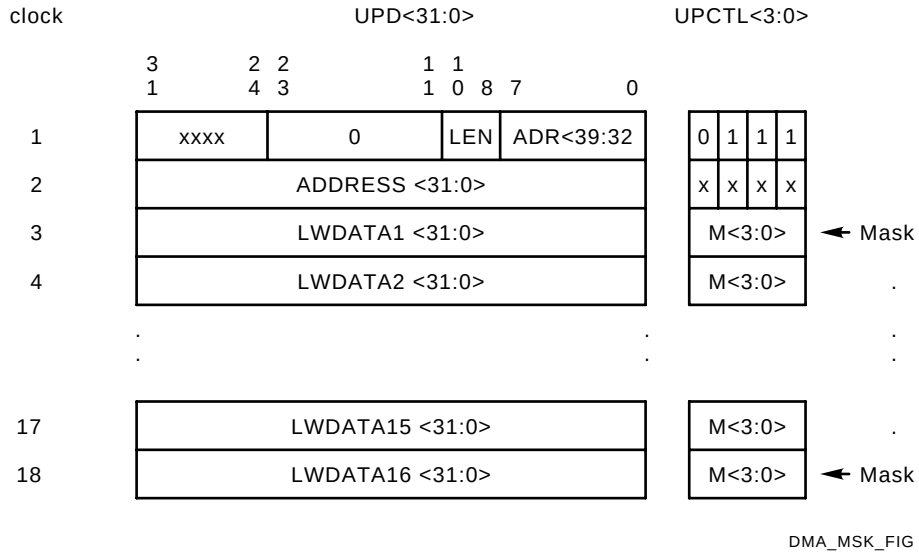
ADR<39:0> is the target address for the Laser memory read and must be naturally aligned to length (LEN) of the data being requested (in this case a Quadword). ADR<39:5> is used by the IOP as the target address for the LSB Read. If ADR<5> is set, a wrap-around read occurs and the IOP receives the requested Hexaword first. ADR<4> is used by the IOP to select which Octaword to return to the I/O Adapter module via a DMA Read Data Return packet. ADR<3> is used to determine in which Quadword to set bit 0 when performing the "Write" portion of the READ/MODIFY/WRITE operation.

The TAG<7:0> field allows the subsequent DMA Read Data Return packet on the DOWN HOSE to be associated with an IRead packet on the UP HOSE. The TAG is generated by the I/O bus adapter.

4.3.2.4 DMA Masked Write With Data

The DMA Masked Write packet is a request on the UP HOSE from the I/O bus adapter module to the IOP for a Laser System Bus data write transaction.

Figure 4–10: DMA Masked Write Packet



A DMA Masked Write packet has three possible packet lengths; Octaword, Hexaword, and Double-Hexaword. However, the Length code for the Octaword Packet may be a Longword (001) or Quadword (010) code. The length code is defined this way to remain consistent with the length code of DMA Read packet. Even though the length code may indicate a Longword or Quadword, the IOP treats the packet as a normal Octaword Masked Write packet and performs a masked write using an Octaword of data. The actual bytes of data in an Octaword Masked Write packet that get written to memory are selected by the byte mask bits and not by the Quadword or Longword length code.

Therefore, when performing a Longword or Quadword transfer using an Octaword Masked Write packet, the I/O bus adapter must set the proper mask bits to select which Longword or Quadword within the Octaword will be written to memory. The I/O bus adapter must clear the remaining mask bits of the Octaword Masked Write packet to prevent the unused bytes of the Octaword packet from being written to memory.

NOTE

Even though the mask bits may "mask out" a particular longword of the packet, the parity across the HOSE must be good for all cycles of the DMA Masked Write packet.

Table 4–8: DMA Masked Write Packet Sizes

Length Code	Packet Data Length	Data Valid in Packet	Number of UP HOSE Cycles Required
001	Octaword	Longword	6
010	Octaword	Quadword	6
011	Octaword	Octaword	6
000	Hexaword	Hexaword	10
100	Double-Hexaword	Double-Hexaword	18

ADR<39:0> is the target address for the DMA Masked Write and must be naturally aligned to length (LEN) of the DMA Masked Write. ADR<39:5> is used by the IOP as the target address. When the length code indicates a Hexaword, ADR<5> is used by the IOP to select which Hexaword in the Read/Merge buffer will be written with the DMA Masked Write data. When the length code indicates a Octaword, Quadword, or Longword, ADR<5:4> are used by the IOP to select which Octaword in the Read/Merge buffer will be written with the DMA Masked Write data.

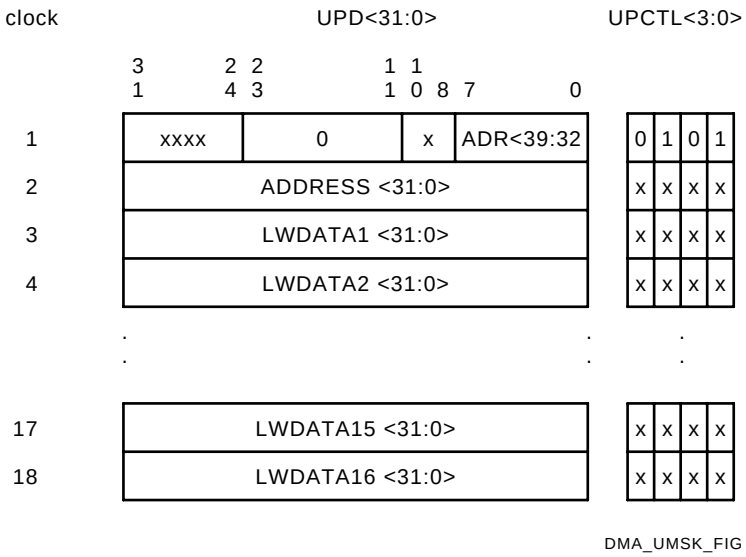
Bits <31:24>, which are normally the TAG field, are "don't care" since DMA Masked Write packets are disconnected and have no corresponding return packet.

Any combination of mask bits is allowed, however the I/O bus adapter may or may not support this capability. A "one" is asserted in the appropriate mask bit to write the corresponding byte of data.

4.3.2.5 DMA Unmasked Write With Data

The DMA Unmasked Write packet is a request on the UP HOSE from the I/O bus adapter module to the IOP for a Laser System Bus data write transaction. The data length of the unmasked write is **always** a Double-Hexaword in length.

Figure 4–11: DMA Unmasked Write Packet



ADR<39:0> is the target address for the Laser memory write and must be naturally aligned to a Double-Hexaword boundary. ADR<39:5> is used by the IOP as the target address for the Double-Hexaword write on the LSB.

Bits <31:24>, which are normally the TAG field, are "don't care" since DMA Unmasked Write packets are disconnected and have no corresponding return packet.

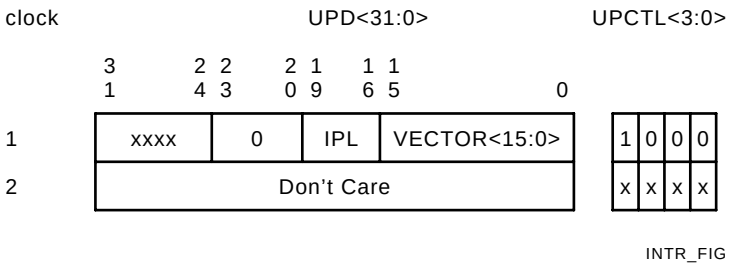
NOTE

The DMA Unmasked Write packet is the most efficient DMA write because it only requires a single write on the Laser bus whereas a DMA Masked Write packet requires a READ/MODIFY/WRITE operation on the Laser bus. Generally speaking, an I/O bus adapter module should be able to exploit the DMA Unmasked Write packet whenever an I/O device is using a "More" type protocol.

4.3.2.6 INTR/IDENT

The INTR/IDENT packet is the combined IDENT vector and IPL for an interrupt on the I/O bus. The status of the Interrupt transaction on the Laser System bus is returned on the DOWN HOSE with a INTR/IDENT Status Return packet.

Figure 4–12: INTR/IDENT Packet



Bits <31:24>, which are normally the TAG field, are "don't care" since only one interrupt per IPL can be pending at a time and the corresponding INTR/IDENT Status Return packet can be easily identified by the IPL field.

The IPL field is the Interrupt Priority Level of the interrupt request. Bits <19:16> correspond to IPL17 to IPL14 respectively. (refer to Figure 4–13). Only one IPL bit should be set per packet. If more than one IPL bit is set or if no IPL bits are set at all, the IOP may not function as expected.

NOTE

If more than one IPL bit is set, the IOP will "AND" the IPL bits with the LCPUMASK and use the result as the data to be written to the LIOINTR register in the CPUs. This could cause interrupts to be simultaneously posted at different levels for each IPL bit that was set. The Vector that was in the packet will be returned for each the the interrupts that are serviced by the CPUs. An INTR/IDENT Status Return packet will be returned across the Down Hose for each Interrupt that is serviced.

If no IPL bits are set in the INTR/IDENT packet, the IOP will still perform a CSR write to the LIOINTR register in the CPUs, but the data will be all zeros (deasserted) so no interrupts will be posted. As a result, no INTR/IDENT Status Return packet will be returned across the DOWN HOSE.

Figure 4–13: INTR/IDENT Packet bits <19:16>

INTR/IDENT Packet bits <19:16>)	
Bit<19>	IPL 17
Bit<18>	IPL 16
Bit<17>	IPL 15
Bit<16>	IPL 14

INTR_IPL_FIG

VECTOR<15:0> is the vector of the interrupt service routine.

NOTE

The second cycle of the IDENT/INTR packet contains NO real data. By making the INTR/IDENT packet 2 HOSE cycles long, all UP HOSE packets are now an even number of cycles. This makes the implementation easier for I/O bus adapters where the adapter UP HOSE controller is running at ½ the UP HOSE frequency. Such is the case with the LAMB module where its controller is running at 64ns and the UP HOSE is cycling at 32ns. Note that UPDATAVAL must still be asserted during this cycle.

4.4 HOSE Errors

There are four types of errors affecting the HOSES:

1. Parity errors on the transmitted data/control information
2. Illegal Packet errors
3. FIFO Overflow errors
4. IOP Internal errors

Parity errors can be detected on any of the four UP HOSES and have corresponding CSR error bits to indicate the failure to the system. Parity errors cause the IOP to generate an error interrupt to the appropriate CPU(s).

Illegal Packet errors indicate that the UPCTL<3:0> field or some other field in the packet did not contain a valid code for that field even though UPDATAVAL was asserted and there were no parity errors. An Illegal Packet error will occur under the following conditions:

1. UPCTL<3:0> contains an Illegal Packet Type code
2. Sequence error. Packet is not the correct number of HOSE cycles expected for the packet type received.
3. Illegal Length code for DMA read packets or DMA Write packets

Overflow errors occur if the IOP receives a packet across the UP HOSE and the IOP's HIC buffers are already full.

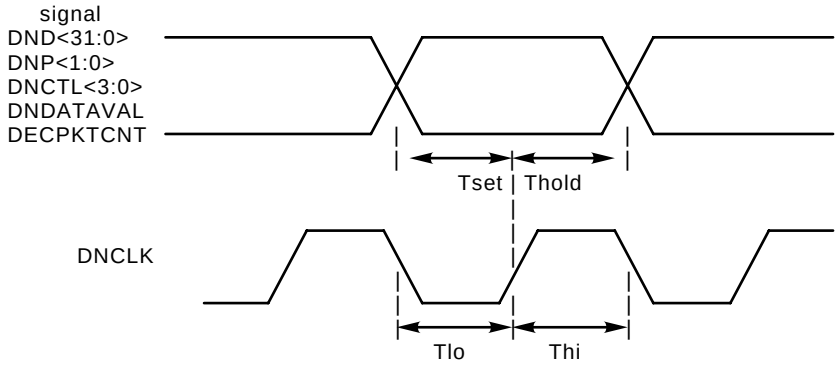
IOP Internal errors occur if the IOP receives a valid packet across the UP HOSE, but for some reason the IOP detects a failure when trying to process the packet. For example, if an IOP controller ends up in an illegal state.

For more information on these four hose errors and how they are handled by the IOP, refer to Chapter 8.

4.5 HOSE AC Specifications

*****TBD*****

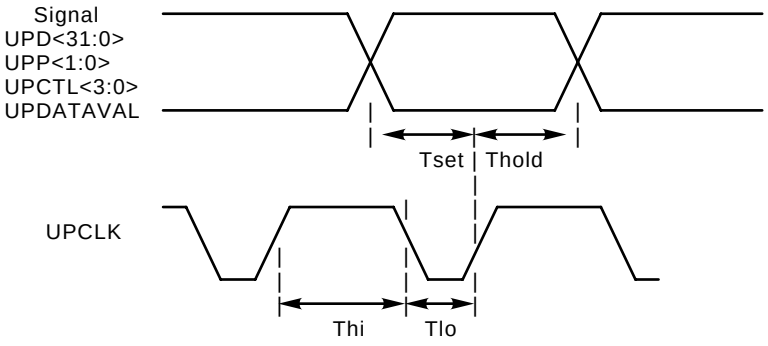
Figure 4–14: DOWN HOSE timing as Driven by IOP



	min	typ	max
Tset	TBDns	17ns	TBDns
Thold	TBDns	17ns	TBDns
Tlo	TBDns	17ns	TBDns
Thi	TBDns	17ns	TBDns

HOSE_AC_FIG1

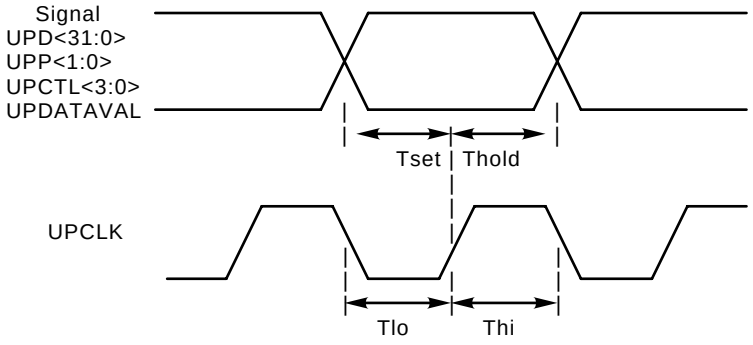
Figure 4–15: UP HOSE timing as Driven by LAMB



	min	typ	max
Tset	TBDns	16.0ns	TBDns
Thold	TBDns	16.0ns	TBDns
Tlo	TBDns	10.7ns	TBDns
Thi	TBDns	21.3ns	TBDns

HOSE_AC_FIG2

Figure 4–16: UP HOSE timing as Driven by FLAG



	min	typ	max
Tset	3.2ns	12.5ns	23.7ns
Thold	1.3ns	12.5ns	21.8ns
Tlo	TBDns	12.5ns	TBDns
Thi	TBDns	12.5ns	TBDns

HOSE_AC_FIG3

4.6 HOSE DC Specifications

*****TBD*****

Will probably copy the FCT (or some other common logic family) DC specs and specify specific parts as the mandatory driver and receiver.

CHAPTER 5

FUNCTIONAL DESCRIPTION

The intent of this chapter is to describe the functionality of the IOP. It is not the intent to describe gate level implementation details. However, the basic transaction flow paths through the IOP are discussed in moderate detail.

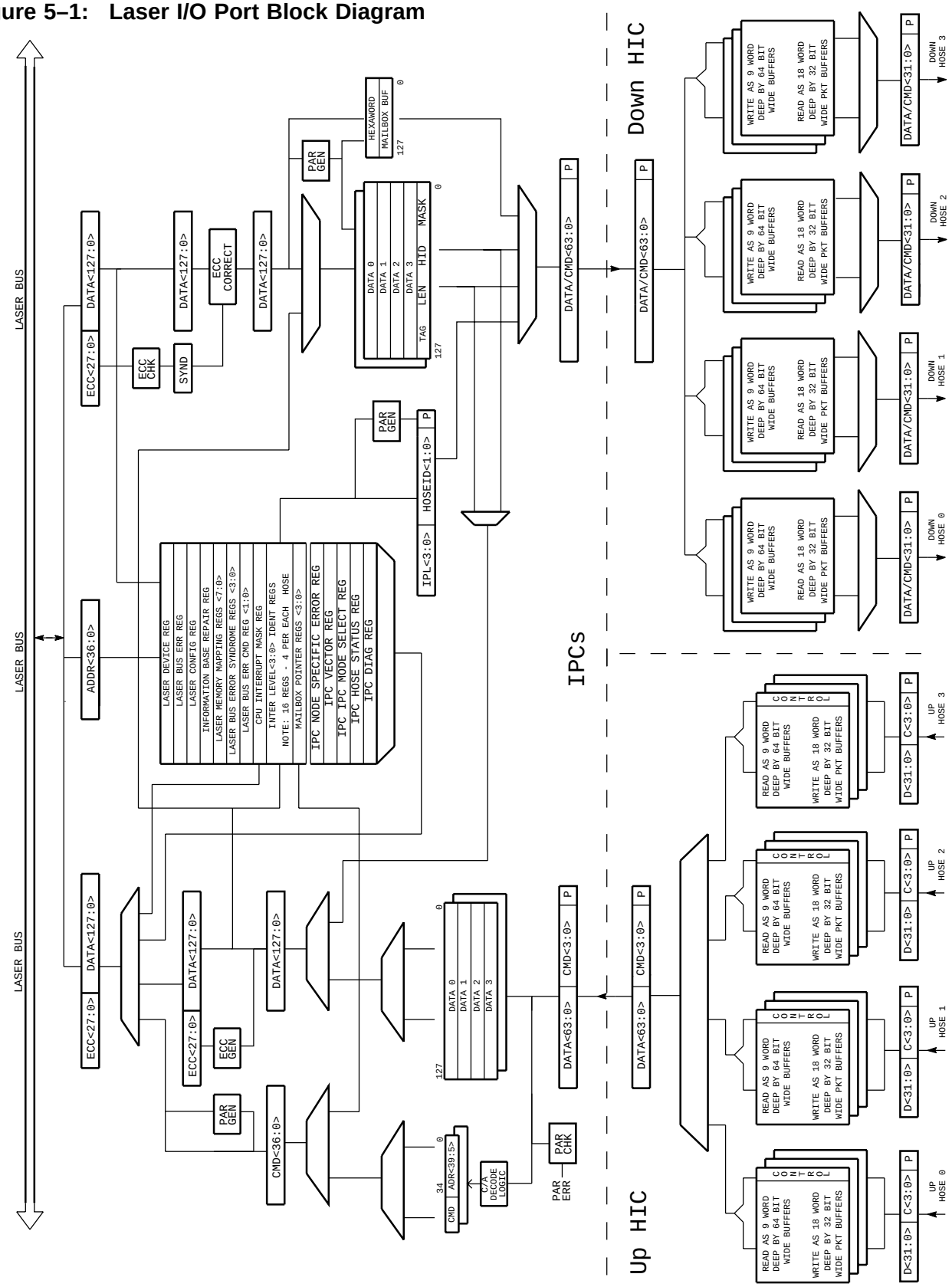
5.1 IOP Block Diagram and Description

Figure 5–1 shows the various blocks and data paths of the IOP and should be used as reference when reading this chapter. The various sections in this chapter will describe the data flow for each of the transactions supported by the IOP, and the operations required by the IOP gate array logic in order to move, manipulate, and check the data for errors.

There are six types of transactions which the IOP is capable of processing. They are as follows:

1. DMA Unmasked Write transaction
2. DMA Masked Write transaction
3. DMA Read transaction
4. Interrupt transaction
5. DMA Interlock Read transaction
6. Mailbox transaction

Figure 5–1: Laser I/O Port Block Diagram



5.1.1 IOP Functional Operation

The IOP houses four large gate arrays. Two of these arrays are called HICs (Hose Interconnect Chips) consisting of an Up HIC and a Down HIC. See Figure 5–1 of this chapter and Figure 11–1 in Chapter 11. The other two arrays are called IPCs (I/O Port Chips).

The IPCs interface the IOP to the LSB. They are identical gate arrays. One array interfaces the high order quadword of the LSB to the IOP, and the other array interfaces the low order quadword and all of the LSB control signals to the IOP.

Even though the IPCs are identical arrays they do not both function identically on the IOP. The lower quadword IPC, which receives the control signals from the LSB, houses the primary IOP control logic and most of the IOP CSR registers. Since these arrays perform different functions, some of the logic in each array is not used (i.e. the control logic and most of the CSR registers in the lower quadword IPC are not used in the upper quadword IPC).

The HICs are also identical gate arrays. However, each array also functions differently on the IOP. One HIC (the Up HIC) receives packets from the four UP HOSES and transmits them to the IPCs. The other HIC (Down HIC) receives packets from the IPCs and transmits them to the four DOWN HOSES. Consequently, as with the IPCs, there is also logic in each HIC which is used in one, but not the other.

5.1.1.1 HIC Functional Operation

The HIC Arrays consist primarily of large double hexaword storage buffers, along with the necessary control logic to transfer the buffer contents between the IPCs and the HOSES.

The Up HIC houses three double hexaword receive buffers for each of the four UP HOSES. Likewise, the Down HIC houses three double hexaword transmit buffers for each of the four DOWN HOSES.

The Up HIC receives packets from the I/O Adapter Module on the UP HOSE in one or more longword cycles, checks each longword for correct parity, and stores the packets in an available Up HIC double hexaword receive buffer assigned to that specific UP HOSE. The Up HIC then determines whether or not the IPCs have any empty Transaction Buffers which can be used to receive the packet. If so, the Up HIC transfers the packet to the IPCs in 17 ns (LSB clock cycle time) quadword cycles via the Up Vortex Bus.

The Up HIC houses synchronizers which synchronize the UP HOSE clock with the 17 ns LSB clock.

The Up HIC generates a "Decrement Packet Count" signal to be asserted to the I/O Adapter Module for one Down Hose cycle whenever the Up HIC removes a packet from its buffer and transmits it across the Up Vortex Bus. The Up HIC also asserts "Decrement Packet Count" for one Down Hose cycle whenever it detects (and discards) an Up Hose packet containing an error.

After 3 packets are sent across the Up Hose, the I/O Adapter Module must wait until it receives the "Decrement Packet Count" signal before it can transmit more Up Hose Packets. This prevents the I/O Adapter Module from overflowing the Up HIC's buffers.

The Down HIC receives packets from the IPCs in multiple 17 ns quadword cycles via the Down Vortex bus, checks each quadword for correct parity, and stores the packets in an available Down HIC double hexaword transmit buffer. The Down HIC then transmits the packet on the DOWN HOSE, with correct parity in one or more 34 ns (½ LSB clock speed) longword cycles.

The Down HIC can always transmit a packet from its FIFO across the DOWN HOSE. DOWN HOSE flow control is maintained solely by the I/O Adapter module. The I/O Adapter Module will never request more data (via UP HOSE packets) than it is capable of receiving across the DOWN HOSE.

Both the Up HIC and the Down HIC process packets on a first in, first out basis. Packets across the same HOSE will never be processed out of order. When packets are received across multiple UP HOSES simultaneously, they are processed (transmitted to the IPCs) in a round-robin fashion.

5.1.1.2 IPC Functional Operation

The IPCs receive packets from the Up HIC via the Up Vortex Bus, in one or more 17 ns quadword cycles and store them in one of two available double hexaword Transaction Buffers.

The IPCs check each quadword of a packet received across the Up Vortex Bus for correct parity. If there is no parity error, the IPCs load the received quadword into the correct location of the selected double hexaword Transaction Buffer.

The IPCs may pipeline two DMA transactions at a time on the LSB such that two IPC initiated transactions may be in various stages of completion on the LSB at any given time. However, transactions will never be processed out of the order in which they are received. Transactions are **always** serviced on a first in, first out basis regardless from which HOSE they originated or are destined.

The IPCs transmit packets from their double hexaword Read/Merge Buffers to the Down Vortex Bus in multiple 17 ns quadword cycles. The IPCs generate correct parity for each quadword of a packet transmitted across the Down Vortex Bus.

These packets are loaded into one of the available Down HIC double hexaword Transmit buffers for the specific DOWN HOSE to which it is addressed.

The IPCs pass transactions to/from the LSB in conformance with the LSB protocol. IPCs check the targeted LSB Memory address of the packet for memory bank conflicts. If the address does not conflict with a memory bank that is currently in use (i.e. does not match any address of the two previous memory banks that were addressed on the LSB) the IPCs arbitrate for the LSB.

Arbitration occurs either on Request Level 5 (the highest priority request) or Request Level 0 (the lowest priority request). The default arbitration algorithm used by the IOP is to arbitrate 6 times using REQ 5 (high) and 2 times using REQ 0 (low). This prevents the IOP from locking out other devices from using the LSB Bus. When the IOP wins arbitration for the LSB, the IPCs are allowed to drive the LSB on the following LSB C/A cycle. At this time

the IPCs transmit the C/A onto the LSB. Eleven LSB cycles later the IPCs either transmit or receive a double hexaword data block on the LSB via four consecutive LSB data cycles. The direction of transmission depends on the type of transaction being executed (i.e. READ or WRITE).

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted.

If the IPCs detect an error while executing a transaction they take the appropriate error action. Usually this consists of setting the appropriate error bit in a CSR and generating an interrupt (see Section 5.1.5) to the LSB. For some transactions such as DMA READS the IPCs will return a DMA Read Return Packet with the Error bit set to the Down HIC via the Down Vortex Bus. If the IPCs detect an error on the LSB, the IPCs will assert the ERR signal on the LSB.

5.1.2 DMA Unmasked Write transaction

Figure 4–11 in Chapter 4 illustrates the UP HOSE format of a DMA Unmasked Write Command packet. DMA Unmasked Write Transactions are always double hexaword in length. Each packet requires 18 longword cycles on the UP HOSE (2 for the C/A and 16 for the data).

When the I/O Adapter Module transmits a DMA Unmasked Write Command packet on the UP HOSE, the Up HIC receives the packet, checks each longword for correct parity, and stores the packet in an available Up HIC double hexaword Receive buffer assigned to that specific UP HOSE.

The Up HIC then determines whether or not the IPCs have any empty Transaction Buffers which can be used to receive the packet. If so, the Up HIC transfers the packet to the IPCs in ten consecutive quadword cycles across the Up Vortex Bus (refer to Appendix B for more detail on Vortex packet format).

After the Up HIC finishes transferring the packet across the Up Vortex Bus, it asserts a "Decrement Packet Count" signal to the I/O Adapter Module for one Down Hose cycle indicating to the I/O Adapter Module that a DBL-HW buffer has become available in the Up HIC. This allows the I/O Adapter Module to keep a running count of available buffers in the Up HIC at any given time.

The IPCs check each quadword of the DMA Unmasked Write Command packet received from the Up HIC for correct parity, and if there are no errors, stores the entire packet in one of two available double hexaword Transaction Buffers.

The IPCs also load a decoded version of the C/A cycle of the packet into the corresponding C /A register of the Transaction Buffer. This version of the C/A is decoded in a format suitable for transmission onto the LSB as a WRITE to the specified double hexaword in memory.

Once the Transaction Buffer has been loaded, the IPCs check the address in the C/A register. If the address does not conflict with a memory bank that is currently in use (i.e. does not match any address of the two previous memory banks that were addressed on the LSB) the IPCs arbitrate for the LSB.

After winning arbitration, the IPCs are allowed to drive the LSB on the following LSB C/A cycle. The IPCs assert the CA signal on the LSB, which indicates the start of an LSB C/A cycle, and transmit the value of the C/A register, which is an LSB WRITE Command, onto the LSB. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted.

Eleven LSB cycles later the IPCs transmit the double hexaword data block in the Transaction Buffer onto the LSB via four consecutive LSB octaword data cycles. ECC is generated on each of the four data cycles via an ECC/Parity Generator.

Whenever the IPCs drive a Data cycle through the LSB transmitters, the data is looped back through the LSB receivers, and compared with the data being transmitted to make sure the IPCs are transmitting good data. If no loopback comparison differences are detected, the Data cycle is considered to have been successfully transmitted.

After transmitting the four Data cycles onto the LSB, the IPCs deallocate the double hexaword Transaction Buffer and corresponding C/A register so they can be used to receive another Up Vortex packet.

If the IPCs detect an error while executing a DMA Unmasked Write Transaction they set the appropriate error bit in a CSR, assert the LSB ERR signal onto the LSB and generate an interrupt (see Section 5.1.5).

A DMA Unmasked Write Command packet is performed as a disconnected write operation and therefore has no status return packet associated with it. Therefore, after the last Data cycle is transmitted on the LSB, the DMA Unmasked Write transaction is considered completed.

5.1.3 DMA Masked Write transaction

Figure 4–10 in Chapter 4 illustrates the UP HOSE format of a DMA Masked Write Command packet. DMA Masked Write Transactions are either octaword, hexaword, or double hexaword in length. Packets are transmitted on the UP HOSE in three lengths, depending on the amount of data being written to memory.

- Octaword Packets - 6 longword cycles
- Hexaword Packets - 10 longword cycles

- Double Hexaword Packets - 18 longword cycles

When the I/O Adapter Module transmits a DMA Masked Write Command packet on the UP HOSE, the Up HIC receives the packet, checks each longword for correct parity, and stores the packet in an available Up HIC double hexaword Receive buffer assigned to that specific UP HOSE.

The Up HIC then determines whether or not the IPCs have any empty Transaction Buffers which can be used to receive the packet. If so, the Up HIC transfers the packet to the IPCs in consecutive quadword cycles across the Up Vortex Bus.

After the Up HIC finishes transferring the packet across the Up Vortex Bus, it asserts a "Decrement Packet Count" signal to the I/O Adapter Module for one Down Hose cycle indicating to the I/O Adapter Module that a DBL-HW buffer has become available in the Up HIC.

The IPCs check each quadword of the DMA Masked Write Command packet received from the Up HIC for correct parity and if there are no errors, stores the entire packet in one of two available double hexaword Transaction Buffers.

The IPCs also load a decoded version of the C/A cycle of the packet into the corresponding C /A register of the Transaction Buffer. This version of the C/A is decoded in a format suitable for transmission onto the LSB as a READ to the specified double hexaword in memory. The reason that the DMA Masked Write packet is decoded into an LSB READ, is because masked writes require an atomic READ/MODIFY/WRITE operation on the LSB, in order to READ the targeted double hexaword, MERGE the DMA write data, and WRITE the updated double hexaword back to memory.

Before the READ can be issued onto the LSB, the IPCs must allocate a double hexaword Read/Merge buffer so there is a place to merge the DMA Masked Write data with the double hexaword from Laser memory.

Once this is done, the IPCs check the address in the C/A register. If the address does not conflict with a memory bank that is currently in use (i.e. does not match any address of the two previous memory banks that were addressed on the LSB) the IPCs arbitrate for the LSB.

After winning arbitration, the IPCs are allowed to drive the LSB on the following LSB C/A cycle. The IPCs assert the CA signal on the LSB, which indicates the start of an LSB C/A cycle, and transmit the value of the C/A register, which is an LSB READ Command, onto the LSB. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted.

Once the READ command is transmitted onto the LSB, the masked data from the DMA Masked Write packet stored in the Transaction Buffer is moved to the corresponding byte locations of the previously allocated Read/Merge buffer. Mask bits are then set on these byte locations to prevent the data from being over-written when the return data for the READ is returned on the LSB.

Eleven LSB cycles after the READ C/A was issued on the LSB, the IPCs receive the double hexaword data block from LSB Memory via four consecutive LSB octaword data cycles. The IPCs check each octaword for correct ECC, and if there are no errors, merge the double hexaword Data Block in the double hexaword Read/Merge Buffer causing the unmasked byte locations of the Read/Merge buffer to get updated.

As the Read/Merge Buffer is being updated with the return data, the IPCs start to arbitrate for the LSB at Request Level 5 (the highest priority request) in order to write the updated information back to Laser memory. Since the arbitration occurs on REQ 5 the IPCs are guaranteed to win the LSB on the following LSB C/A cycle thus keeping the READ/MODIFY /WRITE operation atomic.

The COMMAND field in the C/A register is changed from a READ to a WRITE command. After winning the arbitration, the IPCs assert the CA signal on the LSB, which indicates the start of an LSB C/A cycle. The value of the C/A register which is now a WRITE Command to the DMA target address, is driven onto the LSB. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted. The IPCs then deallocate the double hexaword Transaction Buffer and corresponding C/A register so they can be used to receive another Up Vortex packet.

Eleven LSB cycles later the IPCs transmit the double hexaword data block in the Read /Merge Buffer onto the LSB via four consecutive LSB octaword data cycles. ECC is generated for each of the four data cycles via an ECC/Parity Generator.

Whenever the IPCs drive a Data cycle through the LSB transmitters, the data is looped back through the LSB receivers and compared with the data being transmitted to make sure the IPCs are transmitting good data. If no loopback comparison differences are detected, the Data cycle is considered to have been successfully transmitted.

If the IPCs detect an error while executing a DMA Masked Write Transaction they set the appropriate error bit in a CSR, assert the LSB ERR signal onto the LSB if appropriate, and generate an interrupt (see Section 5.1.5).

A DMA Masked Write Command packet is performed as a disconnected write operation and therefore has no status return packet associated with it. Therefore, after the last Data cycle is transmitted on the LSB, the IPCs deallocate the Read/Merge Buffer and the DMA Masked Write transaction is considered completed.

5.1.4 DMA Read transaction

Figure 4–8 in Chapter 4 illustrates the UP HOSE format of a DMA Read Command Packet. DMA Read Command Packets can specify three possible packet lengths; octaword, hexaword, and double hexaword. Each DMA Read Command packet requires 2 longword cycles on the UP HOSE.

For each UP HOSE DMA Read Command Packet there is an associated DOWN HOSE DMA Read Data Return Packet, Figure 4–3 in Chapter 4 which is used to return the read data to the I/O Adapter Module. DMA Read Data Return Packets are transmitted on the DOWN HOSE in varying lengths, depending on the amount of data being returned. The lengths are as follows:

- Octaword Packets - 5 longword cycles
- Hexaword Packets - 9 longword cycles
- Double Hexaword Packets - 17 longword cycles

When the I/O Adapter Module transmits a DMA Read Command packet on the UP HOSE, the Up HIC receives the packet, checks each longword for correct parity, and stores the packet in an available Up HIC double hexaword Receive buffer assigned to that specific UP HOSE.

The Up HIC then determines whether or not the IPCs have any empty Transaction Buffers which can be used to receive the packet. If so, the Up HIC transfers the packet to the IPCs in two consecutive quadword cycles across the Up Vortex Bus.

After the Up HIC finishes transferring the packet across the Up Vortex Bus, it asserts a "Decrement Packet Count" signal to the I/O Adapter Module for one Down Hose cycle indicating to the I/O Adapter Module that a DBL-HW buffer has become available in the Up HIC.

The IPCs check the quadword of the DMA Read Command Packet received from the Up HIC for correct parity. If there are no errors, the IPCs store the quadword in the C/A location of one of two available double hexaword Transaction Buffers.

The IPCs also load a decoded version of the quadword into the corresponding C/A register of the Transaction Buffer. This version of the C/A is decoded in a format suitable for transmission onto the LSB as a READ to the specified double hexaword in memory which contains the requested data.

Before the READ can be issued on the LSB, the IPCs must allocate one of two available Read /Merge Buffers for the return READ Data. Once this is done, the IPCs check the address in the C/A register. If the address does not conflict with a memory bank that is currently in use (i.e. does not match any address of the two previous memory banks that were addressed on the LSB) the IPCs arbitrate for the LSB.

After winning arbitration, the IPCs are allowed to drive the LSB on the following LSB C/A cycle. The IPCs assert the CA signal on the LSB, which indicates the start of an LSB C/A cycle, and transmit the value of the C/A register, which is an LSB READ Command, onto the LSB. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted.

While the IPCs are waiting for the Read data to be returned from LSB Memory they transfer the TAG, LEN and HOSE ID fields of the C/A location in the Transaction Buffer to the TAG, LEN and HOSE ID fields of the C/A location in the allocated Read/Merge Buffer.

Once the TAG, LEN and HOSE ID fields of the double hexaword Transaction Buffer are moved to the Read/Merge buffer, the IPCs deallocate the double hexaword Transaction Buffer and corresponding C/A register so they can be used to receive another Up Vortex packet.

Eleven LSB cycles after the C/A was issued on the LSB, the IPCs receive the double hexaword data block from LSB Memory via four consecutive LSB octaword data cycles. The IPCs check each octaword for correct ECC, and if there are no errors, stores the double hexaword Data Block in the appropriate data locations of the previously allocated double hexaword Read/Merge Buffer.

Once the Read/Merge Buffer has been loaded the IPCs determine whether or not the Down HIC has any empty double hexaword Transmit Buffers for the addressed DOWN HOSE which can be used to receive the packet. If so, the IPCs transmit the DMA Read Data Return packet to the Down HIC in consecutive quadword cycles across the Down Vortex Bus. The IPCs generate correct parity for each quadword transmitted across the Down Vortex Bus.

If the transaction was an octaword Read Command the IPCs transfer only the specific octaword within the double hexaword which was requested. This requires three quadword cycles on the Down Vortex Bus (1 for the TAG and LEN information, and 2 for the return data).

If the transaction was a hexaword Read Command the IPCs transfer the specific hexaword within the double hexaword which was requested. This requires five quadword cycles on the Down Vortex Bus.

If the transaction was a double hexaword Read Command the IPCs transfer the entire double hexaword which was requested. This requires nine quadword cycles on the Down Vortex Bus.

Once the DMA Read Data Return packet is transmitted across the Down Vortex to the Down HIC, the IPCs deallocate the Read/Merge Buffer free it up for other transactions.

The Down HIC then transmits the DMA Read Data Return packet from the double hexaword Transmit Buffers across the DOWN HOSE in consecutive longword cycles to the I/O Adapter Module.

If the IPCs detect an error while executing a DMA Read Transaction they set the appropriate error bit in a CSR, assert the LSB ERR signal onto the LSB if appropriate, and generate an interrupt (see Section 5.1.5). If possible, the IPCs will also return a DMA Read Data Return Packet with the Error bit set (see Figure 4–4 in Chapter 4) to the I/O Adapter Module.

The IPCs set the error bit (bit 23 of 1st cycle) in the DMA Read Data Return Packet containing the error. The IPCs then determine whether or not the Down HIC has any empty double hexaword Transmit Buffers for the addressed DOWN HOSE which can be used to receive the packet. If so, the IPCs transmit the DMA Read Data Return With Error packet to the Down HIC using consecutive quadword cycles across the Down Vortex Bus. The IPCs generate correct parity for the quadword cycles transmitted across the Down Vortex Bus.

The Down HIC then transmits the DMA Read Data Return With Error Packet from the double hexaword Transmit Buffers across the DOWN HOSE as a single longword cycle to the I/O Adapter Module.

5.1.5 Interrupt transactions

There are two basic types of interrupts handled by the IOP module; those generated across the UP HOSE by the I/O Adapter module via an INTR/IDENT packet and those initiated internally by the IOP to flag the CPU that the IOP module detected an error. The following two sections describe each of these interrupt types.

5.1.5.1 I/O Adapter Initiated Interrupts

Figure 4–12 in Chapter 4 illustrates the UP HOSE format of an INTR/IDENT packet. The IPL and Vector are are sent across the UP HOSE using a single packet. Therefore, the I/O Adapter module must first acquire the Vector from the interrupting I/O device before issuing the INTR/IDENT packet to the IOP.

Once the I/O Adapter module has obtained the interrupt vector, it will construct an INTR /IDENT packet for the given IPL. If the I/O Adapter module *doesn't* already have an INTR /IDENT pending at that IPL, and if there is a buffer available in the Up HIC, the I/O Adapter module will transmit the INTR/IDENT packet across the UP HOSE into the Up HIC double hexaword Receive buffer. The INTR/IDENT packet requires two longword cycles.

As the Up HIC receives the packet, it checks the longword for correct parity, and stores it in an available Up HIC double hexaword Receive buffer assigned to that specific UP HOSE.

The Up HIC then checks to determine whether or not the IPCs have any empty Transaction Buffers which can be used to receive the packet. If so, the Up HIC transfers the packet to the IPCs in two quadword cycles across the Up Vortex Bus. Note that the TAG, IPL, and Vector from the single longword INTR/IDENT packet are sent to both halves of the IPC chips by replicating it on both the upper and lower longwords of the Up Vortex Bus. This allows both IPCs to receive the information needed to service the packet. (refer to Appendix B for more detail on Vortex packet format).

After the Up HIC finishes transferring the packet across the Up Vortex Bus, it asserts a "Decrement Packet Count" signal to the I/O Adapter Module for one Down Hose cycle indicating to the I/O Adapter Module that a DBL-HW buffer has become available in the Up HIC.

When the IPCs receive the packet, they check it for correct parity and store it in one of the two double hexaword Transaction Buffers.

NOTE

For IPLs 14, 15, and 16, as many as 4 Interrupts at a given level may be pending on the LSB at the same time (one device interrupt at a given level per HOSE). For IPL17 it is possible for 5 interrupts to be pending at the same time (1 device interrupt across each of the four UP HOSES and 1 IOP generated error interrupt). Since the CPUs can queue up this many interrupts, the IOP may immediately post to the LSB each interrupt it receives at any IPL. The only special consideration is that the IOP prioritizes error interrupts over any device interrupts.

The interrupt request is posted by performing a CSR WRITE to the Laser I/O Interrupt register (LIOINTR) in broadcast space.

The arbitration is performed in the same manner as a DMA Read transaction (refer to Section 5.1.4). All CSR WRITEs are considered to be accesses to Bank 0. Therefore, the last two preceding C/A cycles could not be to Bank 0. When the IOP wins the arbitration request, the IPCs assert the CA signal on the LSB and drive a hardwired C/A onto D<37:0> of the LSB. This hardwired version of the C/A is in a format suitable for transmission onto the LSB as a CSR WRITE to the Laser I/O Interrupt register (LIOINTR) in broadcast space. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted.

While the IPCs are waiting to transmit the write data for the CSR WRITE, the IPL, Vector, and HOSE NUMBER are moved from the Transaction Buffer to the CSR registers so that the Interrupt Levelx Ident register (LILIDX) can be loaded with the Vector, the IPL can be AND'ed with the CPU Interrupt Mask register (LCPUMASK), and the HOSE NUMBER can be saved to return in the INTR/IDENT Status Return Packet.

As the IPL, Vector, and HOSE NUMBER are moved to the CSR block, IPL<3:0> is AND'ed with each of the four lower nibbles of the LCPUMASK register and stored in a temporary register to be used as the write data for the CSR WRITE. IPL<3:0> is also used to select which of the four Interrupt Levelx Ident registers will be loaded with the Vector. IPL<0> corresponds to LILID0 and IPL<3> corresponds to LILID3. The Vector is loaded into the lower 16 bits of the LILIDx register.

Each LILIDx register actually consists of a 4 deep FIFO. This enables all four possible Interrupts from 4 separate HOSES at a given Level to be temporarily stored in the IOP. However, to the LSB this 4 deep LILIDx FIFO appears as one register. The IOP always transmits on the LSB the least recently loaded Interrupt Vector from a specific LILIDx FIFO Register.

Once the IPL, Vector, and HOSE NUMBER fields have been removed from the Transaction Buffer the IPCs deallocate the buffer so it can be used to receive another transaction.

Eleven LSB cycles after the CSR WRITE to the LIOINTR was issued on the LSB, the IPCs drive the data stored in the temporary register (IPL AND'ed with LCPUMASK) onto the LSB for the first of four consecutive LSB octaword data cycles. The IPCs will "default" the Laser bus for the last three LSB data cycles causing bad ECC for these unused cycles.

Some time after the CSR WRITE data is transmitted to the LIOINTR register, one of the CPUs will issue a CSR READ to the IOP's Interrupt Levelx Ident FIFO register (LILIDx) corresponding to the requested IPL.

Eleven LSB cycles after this CSR READ, the IPCs drive the least recently stored Vector in the LILIDx FIFO register onto the LSB for the first of four consecutive LSB octaword data cycles. The IPCs will "default" the Laser bus for the last three LSB data cycles causing bad ECC for these unused cycles.

As soon as the LILIDx Vector is returned to the requesting CPU, the IPCs will clear the valid flag associated with that location of the LILIDx FIFO register, and generate an INTR /IDENT Status Return packet across the DOWN HOSE to the I/O Adapter module. To generate an INTR/IDENT Status Return packet, the IOP will use the HOSE NUMBER information previously saved in the CSR block.

First, the Interrupt Status buffer is checked to make sure it is available. Then the HOSE NUMBER and IPL are moved to this buffer.

The IPCs then check to determine whether or not the Down HIC has any empty double hexaword Transmit Buffers for the addressed DOWN HOSE, which can be used to receive the INTR/IDENT Status data. If so, the IPCs transmit the INTR/IDENT Status Return packet to the Down HIC in one quadword cycle across the Down Vortex Bus. The IPCs generate correct parity for the quadword transmitted across the Down Vortex Bus.

After the INTR/IDENT Status Return packet is transmitted to the Down HIC, the Interrupt Status buffer is deallocated to free it up for other INTR/IDENT Status Return packets.

Once a double hexaword Transmit Buffer has been loaded with an INTR/IDENT Status Return packet, the the Down HIC will unload packet across the DOWN HOSE. The INTR /IDENT Status Return packet requires 1 longword cycle on the DOWN HOSE (refer to Figure 4–5.

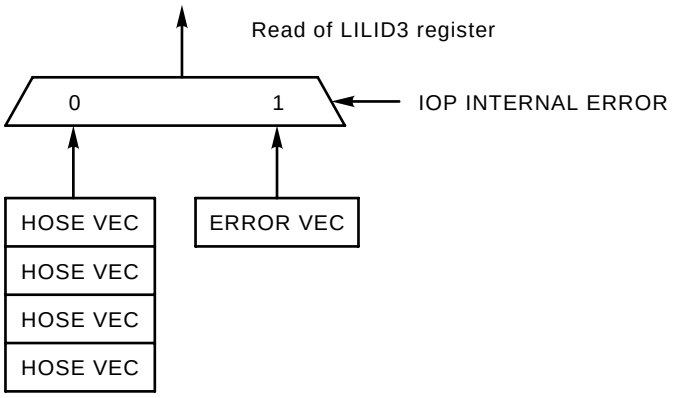
After the the I/O Adapter Module receives the INTR/IDENT Status Return packet, it can generate another INTR/IDENT packet at the same IPL.

5.1.5.2 IOP Initiated Interrupts

If, during the course of processing transactions, the IOP detects an error and the corresponding interrupt enable bit is set, the IOP will generate an error interrupt to the CPU(s). IOP error interrupts are posted at IPL17 and are prioritized over HOSE initiated interrupts.

If there are already HOSE interrupts pending at IPL17 when an error occurs, the IOP will post another IPL17 interrupt (via CSR WRITE to LIOINTR) and return the error vector (ahead of any pending HOSE interrupts) the next time a CPU reads the LILID3 register (see Figure 5–2).

Figure 5–2: IPL17 error vector select logic



The error interrupt is posted on the LSB by performing a CSR WRITE to the Laser I/O Interrupt register (LIOINTR) in broadcast space.

The arbitration is performed in the same manner as a DMA Read transaction (refer to Section 5.1.4). All CSR WRITEs are considered to be accesses to Bank 0. Therefore, the last two preceding C/A cycles could not be to Bank 0. When the IOP wins the arbitration request, the IPCs assert the CA signal on the LSB and drive a hardwired C/A onto D<37:0> of the LSB. This hardwired version of the C/A is in a format suitable for transmission onto the LSB as a CSR WRITE to the Laser I/O Interrupt register (LIOINTR) in broadcast space. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted.

While the IPCs are waiting to transmit the write data for the CSR WRITE, the error vector from the Interrupt Vector Register (IPCVR) register is loaded into the Interrupt Level3 Ident register (LILID3) FIFO. 1000100010001000#2 is AND'ed with the CPU Interrupt Mask register (LCPUMASK) to determine the target CPU(s) and stored in a temporary register.

Eleven LSB cycles after the CSR WRITE to the LIOINTR was issued on the LSB, the IPCs drive the data stored in the temporary register onto the LSB for the first of four consecutive LSB octaword data cycles. The IPCs will "default" the Laser bus for the last three LSB data cycles causing bad ECC for these unused cycles.

Some time after the CSR WRITE data is transmitted to the LIOINTR register, one of the CPUs will issue a CSR READ to the IOP's LILID3 FIFO register to retrieve the vector.

Eleven LSB cycles after this CSR READ, the IPCs drive the error vector stored in the LILID3 FIFO register onto the LSB for the first of four consecutive LSB octaword data cycles. The IPCs will "default" the Laser bus for the last three LSB data cycles causing bad ECC for these unused cycles.

As soon as the error vector is returned to the requesting CPU, the IOP will clear that location of the LILID3 FIFO register (set it to all zeros), and the error interrupt transaction is considered complete for the IOP.

5.1.6 DMA Interlock transactions

5.1.6.1 DMA Interlock Read transaction

Support for an interlocked operation on a memory data structure is necessary to support primitive VAX CI-port style devices such as XCD and DASH. Since LASER memory has no primary hardware interlock capabilities, the equivalent functionality has been provided by the IOP for XMI devices. Interlocked memory operations are not supported for the Futurebus+.

When a node on the XMI wants to lock a memory data structure, it issues an IREAD command. This command is seen by the LAMB module which sends it up the UP hose. The format of this packet on the UP hose is shown in Figure B–3. This packet can only be an octaword in length, but the LENGTH field in the packet will indicate a quadword length.

When the LAMB transmits a DMA Interlock Read Command packet on the UP HOSE, the Up HIC receives the packet, checks each longword for correct parity, and stores the packet in an available Up HIC double hexaword Receive buffer assigned to that specific UP HOSE.

The Up HIC then determines whether or not the IPCs have any empty Transaction Buffers which can be used to receive the packet. If so, the Up HIC transfers the packet to the IPCs in two quadword cycles across the Up Vortex Bus.

The IPCs check the octaword of the DMA IREAD Command Packet received from the Up HIC for correct parity. If there are no errors, the IPCs store the octaword in the C/A location of one of two available double hexaword Transaction Buffers.

The IPCs also load a decoded version of the octaword into the corresponding C/A register of the Transaction Buffer. This version of the C/A is decoded in a format suitable for transmission onto the LSB as a READ to the specified double hexaword in memory which contains the requested data.

Before the READ can be issued on the LSB, the IPCs must allocate one of two available Read/Merge Buffers for the return read data. Once this is done, the IPCs check the address in the C/A register. If the address does not conflict with a memory bank that is currently in use (i.e. does not match any address of the two previous memory banks that were addressed on the LSB) the IPCs arbitrate for the LSB.

After winning arbitration, the IPCs are allowed to drive the LSB on the following LSB C/A cycle. The IPCs assert the CA signal on the LSB, which indicates the start of an LSB C/A cycle, and transmit the value of the C/A register, which is an LSB READ Command, onto the LSB. (Note that an IREAD command is not defined on the LSB). Parity is generated on the C/A cycle via an ECC/Parity Generator.

Since IREADs are not defined on the LSB, an IREAD coming up from an I/O node is turned into a READ and a WRITE on the LSB (effectively a READ/MODIFY/WRITE). The IOP MUST arbitrate for the bus using REQ 5 for at least the WRITE. After the READ, this guarantees that the WRITE will occur before another entity on the bus can touch the memory data structure in question, thereby guaranteeing atomicity in the setting of the lock bit in the data structure.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted.

While the IPCs are waiting for the Read data to be returned from LSB Memory they transfer the TAG, LEN and HOSE ID fields of the C/A location in the Transaction Buffer to the TAG LEN and HOSE ID fields of the C/A location in the allocated Read/Merge Buffer.

Eleven LSB cycles after the C/A was issued on the LSB, the IPCs receive the double hexaword data block from LSB Memory via four consecutive LSB octaword data cycles. The IPCs check each octaword for correct ECC, and if there are no errors, stores the double hexaword data block in the appropriate data locations of the previously allocated double hexaword Read/Merge Buffer.

Once the Read/Merge Buffer has been loaded, the IPCs determine whether or not the Down HIC has any empty double hexaword Transmit Buffers for the addressed DOWN HOSE which can be used to receive the packet. If so, the IPCs transmit the DMA Read Data Return packet to the Down HIC in consecutive quadword cycles across the Down Vortex Bus. The IPCs generate correct parity for each quadword transmitted across the Down Vortex Bus.

The I/O node has now received the requested read data, but the lock bit must still be set. Bit 0 in the appropriate quadword is set to a ONE. The COMMAND field in the C/A register is changed from a READ to a WRITE command. After winning arbitration, the IPCs assert the CA signal on the LSB, which indicates the start of an LSB C/A cycle. The value of the C/A register, which is now a WRITE Command to the DMA target address, is driven onto the LSB. The WRITE data is taken from the READ/MERGE buffer and is sent onto the LSB.

5.1.6.2 Unlock Write transaction

When an XMI I/O node wishes to unlock a memory data structure, it will generally issue an unlock write transaction on the XMI. (In a special case for LASER, these devices may be changed to issue a masked write instead of an unlock write). When this transaction is seen by the LAMB, the LAMB converts this into a generic masked write operation. The transaction flow is then the same as for the masked write.

5.1.7 Mailbox transaction

Mailbox transactions allow CPU nodes to perform reads and writes to CSRs that exist on external I/O busses (i.e. LAMB and FLAG). First, the CPU will set up a Mailbox structure in Laser memory and then write the Laser address value (where the Mailbox structure resides) into the Mailbox Pointer register (LMBPR) of the IPCs.

There are eight LMBPRs (two for each possible CPU node) allowing each CPU node to have two outstanding Mailbox transactions at a time. This prevents one CPU from getting starved by the other CPU nodes in the system. Once a CPU has two Mailbox Transactions pending, further writes to the LMBPR by the same CPU will be NOAcked (CNF not asserted) to indicate a busy status until one of the two pending Mailbox transactions (for that CPU) is completed.

NOTE

Even though there can be eight outstanding Mailbox transactions queued up in the IPCs at the same time (two for each possible CPU), the IPC's cannot service more than 1 Mailbox Command transaction at a time. Therefore, if two or more Mailbox transactions are queued up the the IPCs, the second Mailbox command packet cannot be generated across the DOWN HOSE until the Mailbox Status Return packet for the first Mailbox Command Packet is returned by the I/O Adapter Module across the UP HOSE.

Strict ordering is maintained on Mailbox transactions, since the IPC's Mailbox transaction queue is a FIFO. The order that Mailbox transactions are received off the LSB is the order that the Mailbox Command packets are transmitted across the Down Vortex Bus.

Once the LMBPR is written by a CPU, the IOP must fetch the Mailbox structure for transmission across the DOWN HOSE. To do this, the address in the LMBPR is concatenated with a hardwired READ command field to form a Mailbox C/A in a format suitable for transmission onto the LSB as a READ to the specified double hexaword in memory which contains the Mailbox structure.

Arbitration is performed in the same manner as a DMA READ transaction (refer to Section 5.1.4). If the Mailbox address does not conflict with a memory bank that is currently in use and the IOP wins the arbitration request, the IPCs assert the CA signal on the LSB and drive the Mailbox C/A onto D<37:0> of the LSB. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the C/A cycle is considered to have been successfully transmitted.

Eleven LSB cycles later the IPCs receive the double hexaword Data Block (Mailbox Data) from LSB Memory via four consecutive LSB octaword data cycles. The IPCs check each octaword for correct ECC, and if there are no errors, store the first hexaword of the double hexaword Data Block in the hexaword Mailbox Buffer. The second hexaword of the double hexaword Data Block is not saved because it is only used when the IPC is returning Mailbox status information back to the CPU.

Once the hexaword Mailbox Buffer has been loaded, the IPCs check to determine whether or not the Down HIC has any empty double hexaword Transmit Buffers. for the addressed DOWN HOSE, which can be used to receive the Mailbox data. If so, the IPCs transmit the Mailbox Command packet to the Down HIC in consecutive quadword cycles across the Down Vortex Bus. The IPCs generate correct parity for each quadword transmitted across the Down Vortex Bus.

Once a double hexaword Transmit Buffer has been loaded with a Mailbox Command packet, the Unload controller in the Down HIC will transmit the Mailbox Command packet across the DOWN HOSE. The Mailbox Command packet requires 8 longword cycles on the DOWN HOSE (refer to Figure 4–2 for exact Mailbox Command Packet format).

After the the I/O Adapter Module decodes and performs the Mailbox Command, it transmits a Mailbox Status Return packet on the UP HOSE. The Up HIC receives the packet, checks each longword for correct parity, and stores the packet in an available Up HIC double hexaword Receive buffer assigned to that specific UP HOSE.

The Up HIC then checks to determine whether or not the IPCs have any empty Transaction Buffers which can be used to receive the Mailbox packet. If so, the Up HIC transfers the packet to the IPCs in consecutive quadword cycles across Up Vortex Bus.

After the Up HIC finishes transferring the packet across the Up Vortex Bus, it asserts a "Decrement Packet Count" signal to the I/O Adapter Module for one Down Hose cycle indicating to the I/O Adapter Module that a DBL-HW buffer has become available in the Up HIC.

The IPCs check each quadword of the Mailbox Status Return Packet received from the Up HIC for correct parity, and if there are no errors, stores the packet in an available double hexaword Transaction Buffer.

The IPCs also load a decoded version of the UPCMD<3:0> field into the corresponding C /A register of the Transaction Buffer. For Mailbox packets, the only portion of the C/A register that is valid is the CMD field (bits <37:35>). The address bits (ADR<39:5>) are not important since the address from the Mailbox Pointer register (LMBPR) will be substituted instead.

Once the IPCs decode that the Up Vortex packet is a Mailbox Status Return Packet, they must perform an atomic READ/MODIFY/WRITE operation on the LSB in order to READ the Mailbox Structure, MERGE the information from Mailbox Status Return Packet, and WRITE the updated Mailbox data back to memory.

To do this, the address in the Mailbox Pointer register is concatenated with a hardwired READ command field to form a Mailbox C/A in a format suitable for transmission onto the LSB as a READ to the specified double hexaword in memory which contains the Mailbox structure.

Before the Mailbox READ can be issued onto the LSB the IPCs must allocate a double hexaword Read/Merge buffer so there is a place to merge the Mailbox Status Return data with the data from the Mailbox READ. Once this is done, the arbitration is performed in the same manner as a DMA Read transaction (refer to Section 5.1.4). If the Mailbox address does not conflict with a memory bank that is currently in use and the IOP wins the arbitration request, the IPCs assert the CA signal on the LSB and drive the Mailbox C/A onto D<37:0> of the LSB. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the Mailbox READ C/A cycle is considered to have been successfully transmitted.

Once the Mailbox READ command is transmitted onto the LSB, the data from the Mailbox Status Packet stored in the Transaction Buffer is moved to the previously allocated Read /Merge buffer. Mask bits are then set to prevent the upper hexaword from being over-written when the return data for the Mailbox READ is returned on the LSB.

After the Mailbox Status packet data is removed from the double hexaword Transaction Buffer, the IPCs deallocate the buffer and corresponding C/A register so they can be used to receive another Up Vortex packet.

Eleven LSB cycles after the Mailbox READ C/A was issued on the LSB, the IPCs receive the double hexaword data block from LSB Memory via four consecutive LSB octaword data cycles. The IPCs check each octaword for correct ECC, and if there are no errors, merge the double hexaword Data Block in the double hexaword Read/Merge Buffer causing the lower hexaword word of the Read/Merge buffer to get updated.

As the Read/Merge Buffer is being updated with the return data, the IPCs start to arbitrate for the LSB at Request Level 5 (the highest priority request) in order to write the updated mailbox information back to Laser memory. Since the arbitration occurs on REQ 5 the IPCs are guaranteed to win the LSB on the following LSB C/A cycle thus keeping the READ /MODIFY/WRITE operation atomic.

The C/A data is formed by concatenating the address in the Mailbox Pointer register (LMBPR) with a hardwired WRITE command field to form a Mailbox C/A in a format suitable for transmission onto the LSB as a WRITE to the specified double hexaword in memory which contains the Mailbox structure. Parity is generated on the C/A cycle via an ECC/Parity Generator.

Whenever the IPCs drive the C/A through the LSB transmitters, the data is looped back through the LSB receivers and compared with the C/A being transmitted to make sure the IPCs are transmitting good data. The IPCs also monitor the LSB Responder Confirmation Line (CNF) for each C/A cycle they transmit onto the LSB. If no loopback comparison differences are detected and the IPCs receive good confirmation (CNF asserted) for the C/A, the Mailbox WRITE C/A cycle is considered to have been successfully transmitted.

Eleven LSB cycles after the Mailbox WRITE C/A was issued on the LSB, the IPCs drive the updated Mailbox data stored in the Read/Merge buffer onto the LSB using four consecutive LSB octaword data cycles. ECC is generated on each data cycle via an ECC/Parity Generator.

Once the double hexaword of data is transferred to main memory, the IPCs deallocate the Read/Merge Buffer and the entire Mailbox Transaction is considered complete. The CPU that initiated the Mailbox Transaction may now write to the Mailbox Pointer register (without fear of getting NOAcked) to cause the IOP to service another Mailbox transaction.

CHAPTER 6

IOP REGISTERS

All IOP control/status registers are accessed using the read and write CSR commands to IOP node space. The IOP implements all the LSB required registers for I/O Ports, as specified by the Laser System Bus Specification. In addition to the required registers, the IOP implements error/status and diagnostic registers specific to the IOP itself. IOP specific registers are located within IPC-A.

6.1 LSB CSR Space

Table 6–1 is a list of the LSB nodespace base addresses.

Table 6–1: LSB Node Base Addresses

Node Number	Module	C/A Cycle D<22:0> (hex)	Software Byte Address (hex)
0	CPU/MEM	40 0000	800 0000
1	CPU/MEM	42 0000	840 0000
2	CPU/MEM	44 0000	880 0000
3	CPU/MEM	46 0000	8C0 0000
4	CPU/MEM	48 0000	900 0000
5	CPU/MEM	4A 0000	940 0000
6	CPU/MEM	4C 0000	980 0000
7	CPU/MEM	4E 0000	9C0 0000
8	**I/O**	**50 0000**	**A00 0000**
Broadcast Space Base	BSB	70 0000	E00 0000

Table 6–2 is a list of the defined LSB required CSRs which the IOP implements. Table 6–3 is a list of the IOP specific CSRs.

NOTE

Note that the addresses shown in the Hexaword Address column are from a hardware viewpoint (as they would appear on the LSB) with the lower 5 address bits stripped off. Accesses made by software to these CSRs will use the full double Hexaword aligned address shown in the Byte Address column.

Table 6–2: LSB Required CSRs

Hexaword Address (hex)	Register	Software Byte Address (hex)
50 0000	LDEV Laser Device Reg	A00 0000
50 0002	LBER Laser Bus Error Reg	A00 0040
50 0004	LCNR Laser Configuration Reg	A00 0080
50 0006	IBR Information Base Repair Reg	A00 00C0
50 0010	LMMR0 Laser Memory Mapping Reg 0	A00 0200
50 0012	LMMR1 Laser Memory Mapping Reg 1	A00 0240
50 0014	LMMR2 Laser Memory Mapping Reg 2	A00 0280
50 0016	LMMR3 Laser Memory Mapping Reg 3	A00 02C0
50 0018	LMMR4 Laser Memory Mapping Reg 4	A00 0300
50 001A	LMMR5 Laser Memory Mapping Reg 5	A00 0340
50 001C	LMMR6 Laser Memory Mapping Reg 6	A00 0380
50 001E	LMMR7 Laser Memory Mapping Reg 7	A00 03C0
50 0030	LBESR0 Laser Bus Error Syndrome Reg 0	A00 0600
50 0032	LBESR1 Laser Bus Error Syndrome Reg 1	A00 0640
50 0034	LBESR2 Laser Bus Error Syndrome Reg 2	A00 0680
50 0036	LBESR3 Laser Bus Error Syndrome Reg 3	A00 06C0
50 0038	LBECR0 Laser Bus Error Command Reg 0	A00 0700
50 003A	LBECR1 Laser Bus Error Command Reg 1	A00 0740
50 0050	LILID0 Interrupt Level0 Ident Reg	A00 0A00
50 0052	LILID1 Interrupt Level1 Ident Reg	A00 0A40
50 0054	LILID2 Interrupt Level2 Ident Reg	A00 0A80
50 0056	LILID3 Interrupt Level3 Ident Reg	A00 0AC0
50 0058	LCPUMASK CPU Interrupt Mask Reg	A00 0B00
50 0060	LMBPR0 Mailbox Pointer Register 0	A00 0C00
50 0062	LMBPR1 Mailbox Pointer Register 1	A00 0C00
50 0064	LMBPR2 Mailbox Pointer Register 2	A00 0C00
50 0066	LMBPR3 Mailbox Pointer Register 3	A00 0C00

Table 6–3: IOP Specific CSRs

Hexaword Address (hex)	Register	Software Byte Address (hex)
50 0100	IPCNSE I/O Port Chip Node Specific Error Reg	A00 2000
50 0102	IPCVR I/O Port Chip Vector Reg	A00 2040
50 0104	IPCMSR I/O Port Chip Mode Selection Reg	A00 2080
50 0106	IPCHST I/O Port Chip Hose Status Reg	A00 20C0
50 0108	IPCDR I/O Port Chip Diagnostic Reg	A00 2100

6.2 LSB Required CSRs

REGISTER BIT FIELD MNEMONICS - Bits or fields in figures are shown with a mnemonic which indicates the characteristic of the bit or field, as follows:

- 0: - Indicates that the initialization state of a bit is "ZERO"
- 1: - Indicates that the initialization state of a bit is "ONE"
- R/W: - A read/write bit to the user. It may be read and modified by hardware.
- RO: - A read-only bit to the user. Only hardware can change the value of the bit. Hardware can look at the bit field. User writes to the bit are ignored.
- WO: - A write-only bit to the user. It is always read as a 0.
- W1C: - A write-one-to-clear bit. It may be read by the user and may be cleared by writing a "1" to the bit. The hardware can change the value of the bit, and can look at the bit. A user write of "0" has no effect on the bit.
- NI: - Not Implemented
- MBZ: - Must Be Zero
- RTC: - Read to Clear. Used specifically in the LILIDx Register

6.2.1 LDEV - Laser Device Reg (50 0000)

The Laser Device Register contains information to identify the IOP and its revision.

Figure 6–1: LDEV - Laser Device Reg (50 0000)

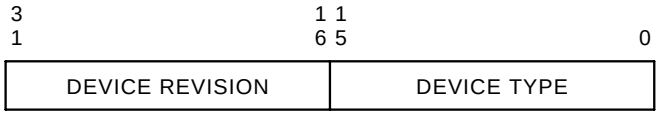


Table 6–4: LDEV - Laser Device Register Format (50 0000)

Name	Bit(s)	Type	Init	Description
DREV	31:16	RO	1	DEVICE REVISION This field identifies the revision level of the IOP. The IOP needs only to implement bits<20:16> to indicate its revision. The remaining bits (<31:21>) will always be read as 0. The list below maps the first 16 revision levels against the module's revision letter that may appear in this field. MODULE REVISION — DREV FIELD A — 0001 B — 0002 C — 0003 D — 0004 E — 0005 F — 0006 not used — not used H — 0008 not used — not used J — 000A K — 000B L — 000C M — 000D N — 000E not used — not used
DTYPE	15:0	RO	2000 (hex)	DEVICE TYPE This field identifies the type of node as follows: • bit <15> CPU node • bit <14> Memory node • bit <13> I/O node • bits<7:0> node type ID **The IOP device type is defined as 2000 (hex)**. All LSB Device Type assignments are shown in Table 6–5

Table 6–5: LSB Device Types

Device	Device Type Value (hex)	Comments
Laser I/O Module	2000	
Laser Memory Module	4000	
Laser EV4 Processor	8000	Ruby
Laser NVAX Scalar/Vector Processor	8001	Neon-V
Laser NVAX Quad Scalar Processor	8002	Neon-Q
Laser EV5 Processor	C000	Assumes Local Memory

6.2.2 LBER - Laser Bus Error Reg (50 0002)

The Laser Bus Error Register contains bits that are set when the IOP detects errors in the LSB environment.

Figure 6–2: LBER - Laser Bus Error Reg (50 0002)

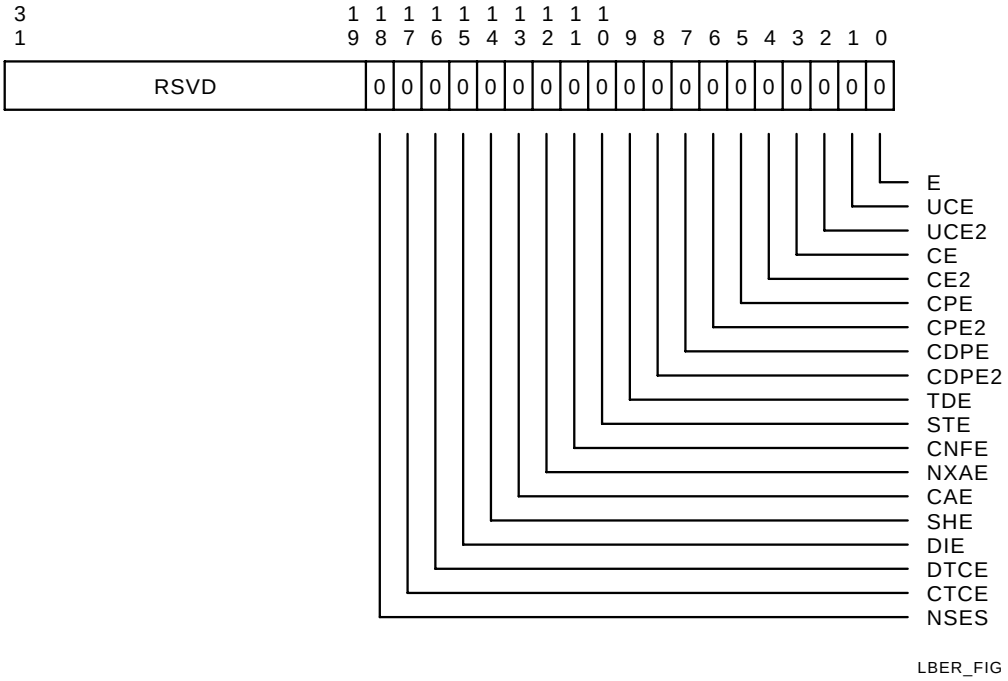


Table 6–6: LBER - Laser Bus Error Register Format (50 0002)

Name	Bit(s)	Type	Init	Description
RSVD	31:19	MBZ	0	RESERVED
NSES	18	RO	0	NODE SPECIFIC ERROR SUMMARY This bit is set when any IOP specific error condition has been recorded within the IPCNSE Register or IPCHST Register. Refer to Section 6.4.1 and Section 6.4.4 for detailed description of IOP specific error conditions. In addition, the IOP will post an interrupt on the LSB if enabled through IPCNES.intr_nses.

Table 6–6 (Cont.): LBER - Laser Bus Error Register Format (50 0002)

Name	Bit(s)	Type	Init	Description
CTCE	17	W1C	0	CONTROL TRANSMIT CHECK ERROR When the IOP drives a LASER control line, it checks the received status on that control line at the end of the cycle to verify that the line was asserted on the bus. If there is a mismatch, the IOP asserts ERR for 1 cycle within 4 cycles, and sets the CTCE bit.
DTCE	16	W1C	0	DATA TRANSMIT CHECK ERROR When the IOP drives a data cycle or command cycle onto the bus, it checks the received data on the D<127:0> and ECC<27:0> lines at the end of the cycle to verify that the received data matches the data that was transmitted. If there is a mismatch, the IOP asserts ERR for 1 cycle within 4 cycles, and sets the DTCE bit. All bits of D<127:0> and ECC<27:0> are checked during memory data cycles. During command cycles and CSR data cycles only bits D<63:0> and ECC<13:0> are checked.
DIE[1]	15	NI	0	Not implemented in the IOP, read as 0.
SHE[1]	14	NI	0	Not implemented in the IOP, read as 0.
CAE	13	W1C	0	CA ERROR When the IOP sees the CA signal asserted during a cycle in which CA is not permitted, the IOP asserts ERR for 1 cycle within 4 cycles, and sets the CAE bit.
NXAE	12	W1C	0	NON-EXISTENT ADDRESS ERROR When the IOP does not receive the CNF signal for a command that it transmitted on the LSB, the IOP asserts ERR for 1 cycle within 4 cycles, and sets the NXAE bit.
CNFE	11	W1C	0	CONFIRM ERROR When the IOP sees the CNF signal asserted during a cycle in which CNF is not permitted, the IOP asserts ERR for 1 cycle within 4 cycles, and sets the CNFE bit.
STE	10	W1C	0	STALL ERROR When the IOP sees the STALL signal asserted during a cycle in which STALL is not permitted, the IOP asserts ERR for 1 cycle within 4 cycles, and sets the STE bit.
TDE	9	W1C	0	TRANSMITTER DURING ERROR The IOP sets the TDE bit if it was driving the D<127:0> lines during the cycle in which a CE, UCE, CPE, or CDPE error occurred on the Laser Bus. Command information will be saved in the Laser Bus Error Command Register.

[1]Because the IOP does not implement an on board cache, there is no need to monitor the SHARED and DIRTY signals on the LSB. Therefore, the DIE and SHE bits in the LBER<15:14> are not implemented. They will always be read as 0.

Table 6–6 (Cont.): LBER - Laser Bus Error Register Format (50 0002)

Name	Bit(s)	Type	Init	Description
CDPE2	8	W1C	0	SECOND CSR DATA PARITY ERROR The IOP sets CDPE2 if CDPE is already set and a second parity error occurs on a CSR data cycle. CDPE2 is set in the same manner as CDPE, with the qualification that CDPE has already be set due to a previous parity error.
CDPE	7	W1C	0	CSR DATA PARITY ERROR The IOP monitors all CSR data cycles for parity errors. If a parity error occurs, the IOP asserts ERR for 1 cycle within 4 cycles after the parity error appears on the Laser Bus. In addition, the IOP captures D<38:0> of the command cycle in LBECR0 and LBECR1, and sets the CDPE bit.
CPE2	6	W1C	0	SECOND COMMAND PARITY ERROR The IOP sets CPE2 if CPE is already set and a second parity error occurs on a command cycle. CPE2 is set in the same manner as CPE, with the qualification that CPE has already been set due to a previous parity error.
CPE	5	W1C	0	COMMAND PARITY ERROR The IOP monitors all command cycles for parity errors. If a parity error occurs, the IOP asserts ERR for 1 cycle within 4 cycles after the parity error appears on the Laser Bus. In addition, the IOP captures D<38:0> of the command cycle in LBECR0 and LBECR1, and sets the CPE bit.
CE2	4	W1C	0	SECOND CORRECTABLE DATA ERROR The IOP sets CE2 if CE is already set and a second correctable ECC error occurs on a data cycle. CE2 is set in the same manner as CE, with the qualification that CE has already been set due to a previous correctable ECC error.
CE	3	W1C	0	CORRECTABLE DATA ERROR The IOP monitors all memory data cycles for correctable ECC errors. If a correctable error occurs, and if the CEEN bit in LCNR is set, the IOP asserts ERR for 1 cycle within 4 cycles after the correctable data appears on the bus. In addition, the IOP captures D<38:0> of the command cycle in LBECR0 and LBECR1, and sets the CE bit.
UCE2	2	W1C	0	SECOND UNCORRECTABLE DATA ERROR The IOP sets UCE2 if UCE is already set and a second correctable ECC error occurs on a data cycle. UCE2 is set in the same manner as UCE, with the qualification that UCE has already be set due to a previous uncorrectable ECC error.

Table 6–6 (Cont.): LBER - Laser Bus Error Register Format (50 0002)

Name	Bit(s)	Type	Init	Description
UCE	1	W1C	0	UNCORRECTABLE DATA ERROR The IOP monitors all memory data cycles for ECC errors. If an uncorrectable error occurs, the IOP asserts ERR for 1 cycle within 4 cycles after the uncorrectable data appears on the bus. In addition, the IOP captures D<38:0> of the command cycle in LBECR0 and LBECR1, and sets the UCE bit.
E	0	W1C	0	ERROR LINE ASSERTED The IOP sets E whenever it detects the ERROR line asserted on the Laser System Bus.

Refer to Chapter 8 for information regarding how each of the errors is handled.

6.2.3 LCNR - Laser Configuration Reg (50 0004)

The Laser Configuration Register contains Laser system-wide configuration setup and status. IOP specific configuration information exists in the I/O Port Chip Mode Selection Register (IPCMSR).

Figure 6–3: LCNR - Laser Configuration Reg (50 0004)

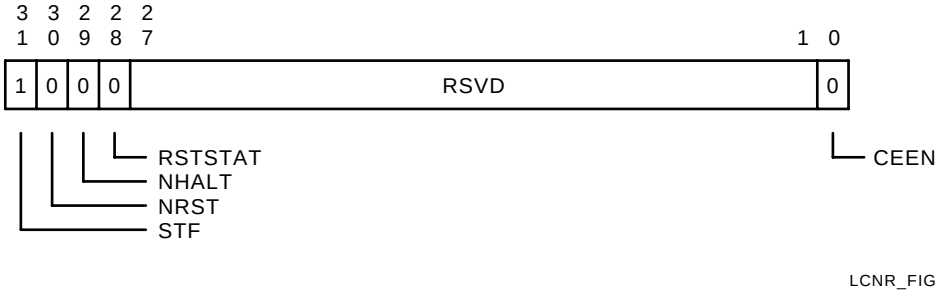


Table 6–7: LCNR - Laser Configuration Register Format (50 0004)

Name	Bit(s)	Type	Init	Description
STF	31	R/W	0	SELF TEST FAIL When clear, indicates that the IOP has not yet passed its power up tests. There is no on-board self test for the IOP, it will be done by the system CPU(s). If the testing is successful then the primary CPU will set this bit. The state of this bit will also affect the IOP Modules *yellow* LED as follows: When this bit is clear, the LED will not be lit. When this bit is set, the LED will be lit.
NRST	30	WO	0	NODE RESET Writing a 1 to this bit initiates a complete reset of the IOP. Transactions in progress are aborted. All internal registers are reset to their default values. All state machines are reset to their IDLE states. Additionally, all attached I/O Bus Adapters will be reset via the Hose signal, DNRST<3:0>.
NHALT	29	NI	0	Not implemented in the IOP, read as 0.
RSTSTAT	28	NI	0	Not implemented in the IOP, read as 0.
RSVD	27:1	MBZ	0	RESERVED
CEEN	0	R/W	0	ENABLE CORRECTABLE ERROR DETECTION When this bit is set and the IOP detects a correctable ECC error, the error will be reported by the IOP asserting the ERR signal and setting the CE bit in LBER. When this bit is not set, the IOP will continue to correct data errors where possible, however, the reporting of the errors will be disabled.

6.2.4 Information Base Repair Register (50 0006)

The Information Base Repair register is used to access the EEPROM located on the IOP module. Access of the EEPROM is made by continual updates of this register by software.

To access the EEPROM, software continually updates the IBR register to transfer command, address, and data to and from the device. Writing an alternating one and zero pattern to the SCLK bit implements the serial data clock used for the EEPROM protocol. Command, address, and write data is serially transferred to the EEPROM by writing to the XMT_SDAT bit in accordance with the SCLK bit. EEPROM read data and response are read serially from the RCV_SDAT bit in accordance with the SCLK bit. When receiving EEPROM read data and responses the XMT_SDAT must be one.

Figure 6–4: IBR

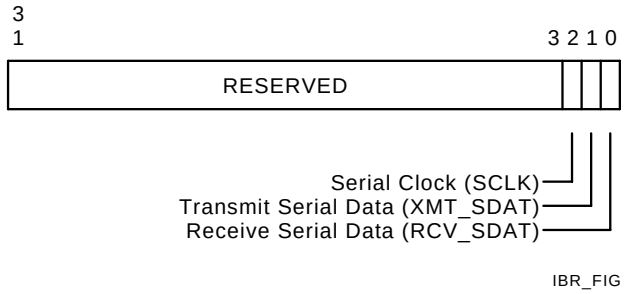


Table 6–8: IBR - Information Base Repair Register Format (50 0006)

Name	Bit(s)	Type	Init	Description
RSVD	31:3	MBZ	0	RESERVED
SCLK	2	R/W	0	SERIAL CLOCK This bit is used to implement the EEPROM serial clock interface by software. When this bit is written with a one, the EEPROM serial clock input is forced to a logic high. When this bit is cleared the serial clock input is forced to low logic level.
XMT_SDAT	1	R/W	1	TRANSMIT SERIAL DATA This bit is used by software to assert the serial data line of the EEPROM to either high or low logic levels. This bit is used with the SCLK bit to transfer command, address, and write data to the EEPROM.
RCV_SDAT	0	RO	1	RECEIVE SERIAL DATA This bit returns the status of the EEPROM serial data line. It is used by software to receive serial read data and EEPROM responses. NOTE: XMT_SDAT must be one to receive an EEPROM response or serial read data.

6.2.5 LMMR0-7 - Laser Memory Mapping Reg 0-7 (50 0010 - 50 001E)

The eight memory mapping registers define the memory configuration for all memory modules installed in the system. These registers are copies of the equivalent Address Mapping Register in each of the memory modules. These registers are loaded by software based on the corresponding memory module register, after the memory modules are initialized and configured. (LMMR0 corresponds to node 0, LMMR1 to node 1, etc.) In most cases, only a subset of these registers are in use, corresponding to the slots which contain memory modules. The E bit in unused registers will remain set to 0 to disable the registers.

Figure 6–5: LMMR0-7 - Laser Memory Mapping Reg 0-7 (50 0010 - 50 001E)

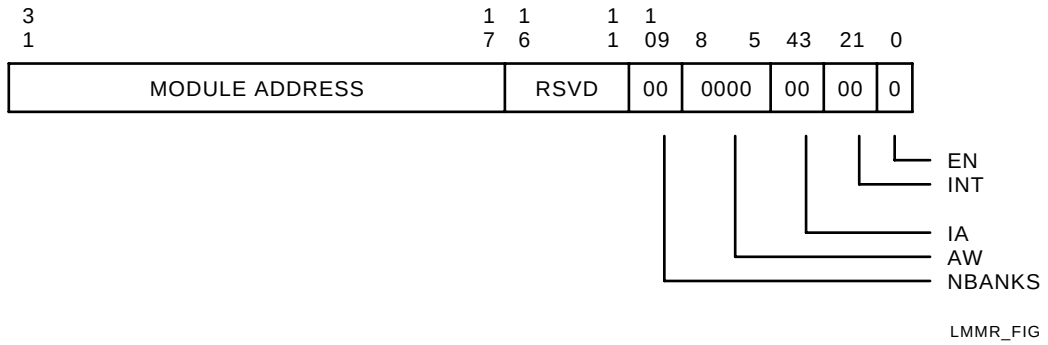


Table 6–9: LMMR0-7 - Laser Memory Mapping Reg 0-7 Format (50 0010 - 50 001E)

Name	Bit(s)	Type	Init	Description
Address	31:17	R/W	0	MODULE ADDRESS This field specifies the most significant bits of the base address of the memory region contained in the memory module associated with this register.
RSVD	16:11	MBZ	0	RESERVED

Table 6–9 (Cont.): LMMR0-7 - Laser Memory Mapping Reg 0-7 Format (50 0010 - 50 001E)

Name	Bit(s)	Type	Init	Description
NBANKS	10:9	R/W	0	NUMBER OF BANKS (per module) This field specifies how many individual memory banks (1, 2, 4, or 8) are contained on the memory module associated with this register. This value determines how many bits of the memory address (0, 1, 2, or 3) are inserted into the bank number.

Value	Banks per Module	Bits in Bank Number
00	1	0
01	2	1
10	4	2
11	8	3

Name	Bit(s)	Type	Init	Description
AW	8:5	R/W	0	ADDRESS WIDTH This field specifies how many bits of the Module Address are valid, starting from the MSB. (The remaining bits of Module Address are ignored.)
IA	4:3	R/W	0	INTERLEAVE ADDRESS This field specifies which interleave within a group of interleaved modules is served by the module associated with this register.
INT	2:1	R/W	0	INTERLEAVE This field specifies how many memory modules are interleaved together with this module (1, 2, or 4). This value determines how many bits of the INT field (0, 1, or 2, starting from the LSB) are compared to the least significant bits of the memory address.

Value	Modules Interleaved	Bits Compared
00	1	0
01	2	1
10	4	2
11	Reserved	Reserved

Table 6–9 (Cont.): LMMR0-7 - Laser Memory Mapping Reg 0-7 Format (50 0010 - 50 001E)

Name	Bit(s)	Type	Init	Description
EN	0	R/W	0	MODULE ENABLE Software sets this bit to a 1 to indicate to the IOP hardware that the contents of this register are valid and must be used to determine memory bank contention. Software must not set this bit unless the corresponding memory module is present and configured in the system.

6.2.6 LBESR0-3 - Laser Bus Error Syndrome Reg 0-3 (50 0030 - 50 0036)

The LSB Bus Error Syndrome Registers contain the syndrome computed from the LSB D and ECC fields received during the cycle that an error was detected. (The syndrome is the bit-wise difference between the ECC check code generated from the received data and the ECC field received over the bus.)

These registers are valid only when CE or UCE in LBER is set.

Figure 6–6: LBESR0-3 - Laser Bus Error Syndrome Reg 0-3 (50 0030 - 50 0036)

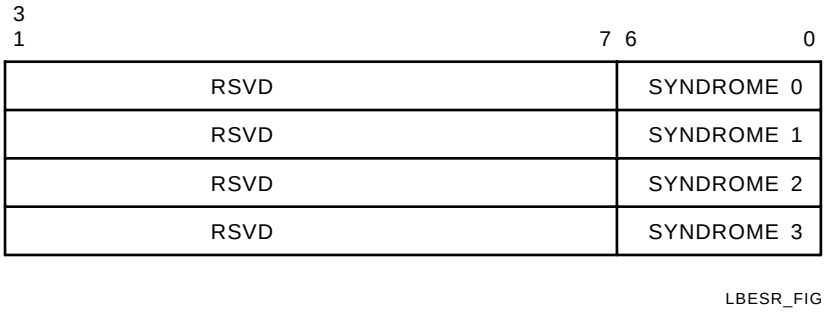


Table 6–10: LBESR0-3 - Laser Bus Error Syndrome Reg 0-3 (50 0030 - 50 0036)

Name	Bit(s)	Type	Init	Description
RSVD	31:7	MBZ	0	RESERVED
Syndrome 0	6:0	RO	0	Syndrome computed from D<31:0> and ECC<6:0> during error cycle.
Syndrome 1	6:0	RO	0	Syndrome computed from D<63:32> and ECC<13:7> during error cycle.
Syndrome 2	6:0	RO	0	Syndrome computed from D<95:33> and ECC<20:14> during error cycle.
Syndrome 3	6:0	RO	0	Syndrome computed from D<127:96> and ECC<27:21> during error cycle.

6.2.7 LBECR - Laser Bus Error Command Reg 0-1 (50 0038 and 50 003A)

The Laser Bus Error Command Registers contain the contents of the LSB command and address fields, CNF, commander ID, and failing data cycle for an LSB command transaction during which an error was detected.

It is important to note that the contents of these registers are valid only when one or more of the following LBER error bits are set: CE, UCE, CPE and CDPE. **If none of these error bits are set when LBECR0 or 1 are accessed, then the data that is returned is unpredictable and not valid.**

Figure 6–7: LBECR - Laser Bus Error Command Reg 0-1 (50 0038 and 50 003A)

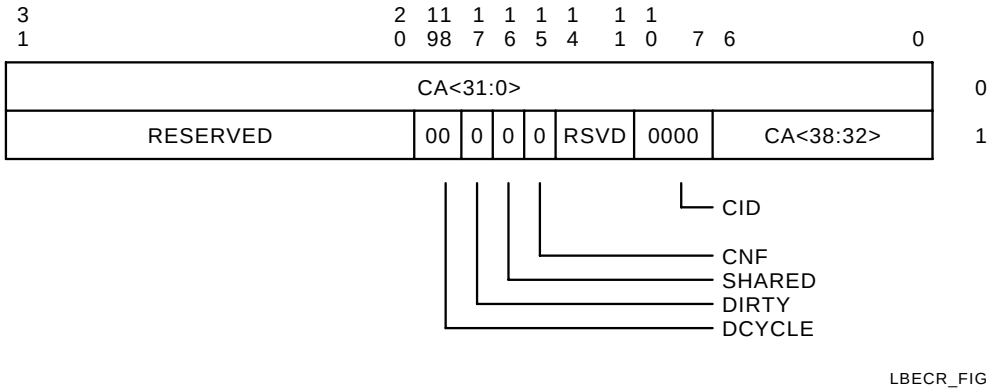


Table 6–11: LBECR - Laser Bus Error Command Register Format (50 0038 and 50 003A)

Name	Bit(s)	Type	Init	Description
	Register 0			
CA<31:0>	31:0	RO	0	COMMAND ADDRESS<31:0> Contents of D<31:0> during an LSB command cycle for which an error was detected. This register is locked by the assertion of CE, UCE, CPE or CDPE and remains locked until software has cleared the associated error bit.
	Register 1			
RSVD	31:20	MBZ	0	RESERVED
DCYCLE	19:18	RO	0	DATA CYCLE Indicates which data cycle had data error. This register is locked by the assertion of CE, UCE, CPE and CDPE. It remains locked until software has cleared the associated error bit(s).

Table 6–11 (Cont.): LBECR - Laser Bus Error Command Register Format (50 0038 and 50 003A)

Name	Bit(s)	Type	Init	Description
DIRTY[1]	17	NI	0	Not implemented in the IOP, read as 0.
SHARED[1]	16	NI	0	Not implemented in the IOP, read as 0.
CNF	15	RO	0	RESPONDER CONFIRMATION When set indicates that CNF was asserted for this command. This register is locked by the assertion of CE, UCE, CPE and CDPE. It remains locked until software has cleared the associated error bit(s).
RSVD	14:11	MBZ	0	RESERVED
CID	10:7	RO	0	COMMANDER ID This field contains the ID of the commander as driven on the REQ lines during transmission of the command address cycle. This register is locked by the assertion of CE, UCE, CPE and CDPE. It remains locked until software has cleared the associated error bit(s).
CA<38:32>	6:0	RO	0	COMMAND ADDRESS<38:32> Contents of D<38:32> during an LSB command cycle for which an error was detected. This register is locked by the assertion of CE, UCE, CPE and CDPE. It remains locked until software has cleared the associated error bit(s).

[1]Because the IOP does not implement an on board cache, there is no need to monitor the SHARED and DIRTY signals on the LSB. Therefore, the DIRTY and SHARED bits in the LBECR<14:13> are not implemented. They will always be read as 0.

6.3 I/O Module CSRs

6.3.1 LILID0-3 - Laser Interrupt Level0-3 Ident Reg (50 0050,52,54,56)

Each of the four registers is the top-most (oldest) entry in a 4 deep queue (one entry per hose) of the interrupts for that IPL. A read to this register sends the "oldest" interrupt IDENT information to the CPU that requests it. The "next oldest" interrupt IDENT information is then read-able at that address. When no active interrupts exist at a given level, reads of the corresponding LILID register will return zeros.

Figure 6–8: LILIDx - Laser Interrupt Levelx Ident Reg (50 0050,52,54,56)

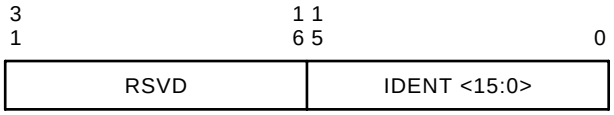


Table 6–12: LILIDx - Laser Interrupt Levelx Ident Register Format (50 0050,52,54,56)

Name	Bit(s)	Type	Init	Description
RSVD	31:16	MBZ	0	RESERVED
IDENT	15:0	RTC	0	IDENT This field is loaded with the vector information supplied by the I/O Bus Adapter (LAMB or FLAG) in the INTR/IDENT Hose command packet. Or if the interrupt was for an IOP error, the vector is loaded from the IPC Vector Register.

6.3.2 LCPUMASK - Laser CPU Interrupt Mask Reg (50 0058)

This register is used to determine which CPU is to service a given level of interrupts. The levels of the interrupts posted from the I/O subsystem are AND'ed with the Laser CPU Interrupt Mask Register to form the write data sent from the IOP to the LIOINTR register.

Figure 6–9: LCPUMASK - Laser CPU Interrupt Mask Reg (50 0058)

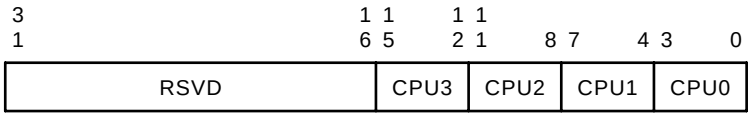


Table 6–13: LCPUMASK - Laser CPU Interrupt Mask Register Format (50 0058)

Name	Bit(s)	Type	Init	Description
RSVD	31:16	MBZ	0	RESERVED
CPU3	15:12	R/W	0	When a bit is set in this field all corresponding interrupts received from the I/O subsystem by the IOP at the corresponding level will be posted to CPU3.
CPU2	11:8	R/W	0	When a bit is set in this field all corresponding interrupts received from the I/O subsystem by the IOP at the corresponding level will be posted to CPU2.
CPU1	7:4	R/W	0	When a bit is set in this field all corresponding interrupts received from the I/O subsystem by the IOP at the corresponding level will be posted to CPU1.
CPU0	3:0	R/W	0	When a bit is set in this field all corresponding interrupts received from the I/O subsystem by the IOP at the corresponding level will be posted to CPU0.

6.3.3 LMBPR0-3 - Laser Mailbox Pointer Reg 0-3 (50 0060, 62, 64, 66)

Software must load the LMBPR with the 64-byte aligned physical address, of the Mailbox Data Structure in main memory, in order to access remote I/O registers. When this register is loaded, the IOP will use the contents of this register as the address to fetch control information (refer to Figure 6–11) required to initiate a remote bus transaction.

IMPLEMENTATION NOTE

Note that the I/O system architecture requires that there be only a single software-visible LMBPR address, regardless of the node id. However, the IOP implements four LMBPR registers. Each CPU is assigned a single LMBPR register with each register representing a two deep queue. This will eliminate the possibility of lock-outs when multiple CPUs desire a Mailbox transaction. It is then possible for a total of eight Mailbox transactions to be outstanding, two each per CPU. However the IOP will never process more then one Mailbox transaction at a time. LMBPR register writes to a full queue will be NoAcked by the IOP until the queue becomes not full. Software will access the LMBPR using address A00 0C00. The CPU's LSB interface logic must insert the two least significant bits of the CPU node id in D<2:1> of the command cycle in order to access its designated LMBPR on the IOP.

Figure 6–10: LMBPR - Laser Mailbox Pointer Reg (50 0060,62,64,66)

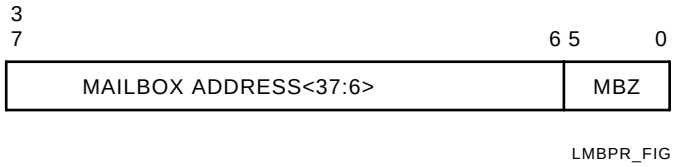


Table 6–14: LMBPR - Laser Mailbox Pointer Register Format (50 0060,62,64,66)

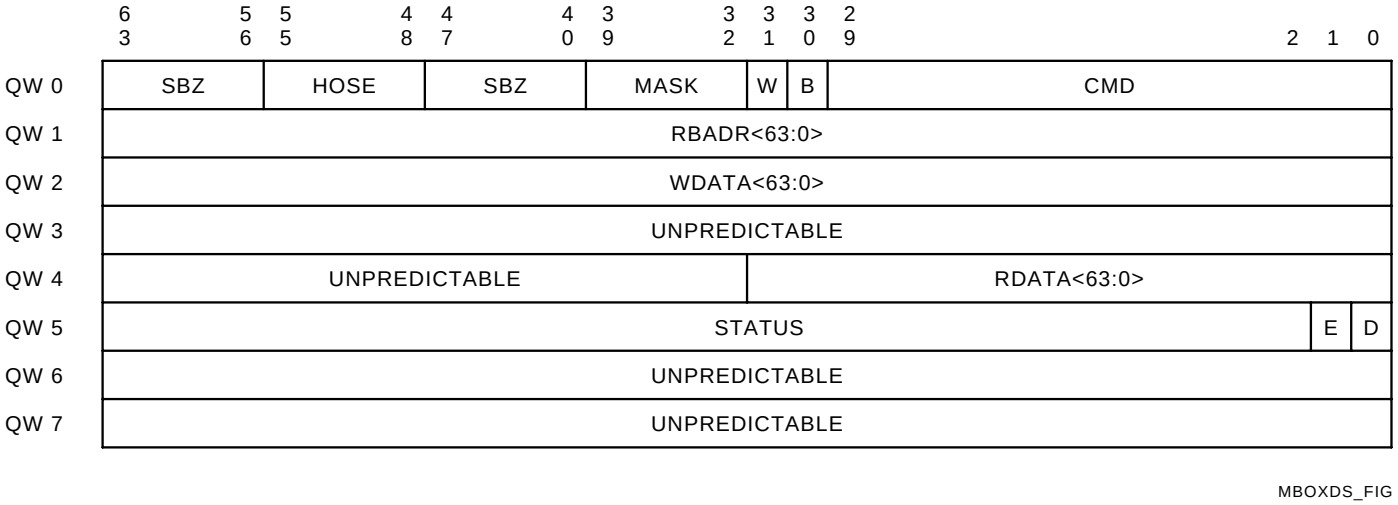
Name	Bit(s)	Type	Init	Description
MBX<37:6>	37:6	WO	0	MAILBOX DATA STRUCTURE ADDRESS This field contains the 64 byte-aligned physical address of the mailbox data structure in memory where the IOP can find information to complete the required operation.
RSVD	5:0	MBZ	0	RESERVED

6.3.3.1 Mailbox Data Structure

Access to remote I/O devices is accomplished using a Mailbox scheme. The Mailbox Data Structure is posted in main memory by software. Software then writes the LMBPR with a pointer to the data structure. The write to the LMBPR signals the IOP to begin the transaction.

Figure 6–11 and Table 6–15 show the Mailbox Data Structure and provide a description of the fields within the structure. For a more detailed description of the Mailbox scheme refer to the Mailbox transaction section of Chapter 5.

Figure 6–11: Mailbox Data Structure



MBOXDS_FIG

Table 6–15: Mailbox Data Structure

Name	Quad word	Bit(s)	Description
CMD	0	29:0	COMMAND This field contains the 30-bit I/O Bus Adapter specific command. For information regarding the encoding of the commands for a specific I/O Bus Adapter, refer to that adapter's specification or the Alpha SRM.
B	0	30	REMOTE BRIDGE ACCESS This bit when set, indicates the command is a special or diagnostic command.

Table 6–15 (Cont.): Mailbox Data Structure

Name	Quad word	Bit(s)	Description
W	0	31	WRITE ACCESS This bit when set, indicates that the remote bus operation is a write.
MASK	0	39:32	DISABLE BYTE MASK These bits when set to a 1 disable bytes within the remote bus address. Mask bit<n> set causes the byte to be disabled; e.g. data byte<n> will NOT be written to the remote address.
SBZ	0	47:40	SHOULD BE ZERO
HOSE	0	55:48	HOSE ID This field contains the ID of the selected Remote Bus or Hose. Because the IOP only supports up to four Hoses, the IOP decodes only bits<49:48> for the Hose ID. The remaining bits in this field are ignored by the IOP. Encoding of bits<49:48> for the four Hoses is as follows: 00 - Hose #0 01 - Hose #1 10 - Hose #2 11 - Hose #3
SBZ	0	63:56	SHOULD BE ZERO
RBADR<63:0>	1	63:0	REMOTE BUS ADDRESS<63:0> This 64-bit field contains the I/O address to be accessed. The I/O device responding to the address can be the I/O Bus Adapter itself or a node residing on the I/O subsystem bus. Refer to the specific I/O Bus Adapter specification regarding supported addresses.
WDATA<63:0>	2	63:0	WRITE DATA<63:0> This field contains data to be written to the selected remote bus address. This field is valid only when the CMD field contains a write-type command code, otherwise the contents of this field are undefined. Refer to the specific adapter's specification for supported data field widths.
UNPRED	3	63:0	UNPREDICTABLE

NOTE: The first Hexaword (Quadword<3:0>) of the Mailbox Data Structure, just described, is READ ONLY for the IOP.

Table 6–15 (Cont.): Mailbox Data Structure

Name	Quad word	Bit(s)	Description
RDATA<31:0>	4	31:0	READ DATA<31:0> This 32-bit field is written, with data from the I/O Bus Adapter, by the IOP at the completion of a Mailbox transaction. If the CMD field had contained a read-type command code and the transaction completed successfully (i.e. the ERR bit is clear and DONE is set), then this data is valid. After the completion of a write-type Mailbox transaction or a failing transaction, this field is unpredictable. Refer to the specific I/O Bus Adapter's specification for supported data field widths.
UNPRED	4	63:32	UNPREDICTABLE
D	5	0	DONE This bit is cleared by software before initiating the Mailbox transaction via a write to the LMBPR. Upon completion of the Mailbox transaction, the I/O Bus Adapter will set this bit to a 1 in the Mailbox Status Packet that it sends across the Hose to the IOP. The IOP will then simply pass the information when it writes the status to memory.
E	5	1	ERROR This bit is cleared by software before initiating the Mailbox transaction via a write to the LMBPR. If the Mailbox transaction fails, the I/O Bus Adapter will set this bit to a 1 in the Mailbox Status Packet it sends across the Hose to the IOP. This bit is valid only if the DONE bit is set.
STATUS	5	63:2	DEVICE SPECIFIC STATUS This field is written, with information from the I/O Bus Adapter, by the IOP at the completion of a Mailbox transaction. Refer to the specific I/O Bus Adapter's specification for support of this field.
UNPREDICTABLE	6	63:0	UNPREDICTABLE Use of this field is currently undefined and the contents are unpredictable.
UNPREDICTABLE	7	63:0	UNPREDICTABLE Use of this field is currently undefined and the contents are unpredictable.

For more detailed information relating to Mailbox transactions, refer to Chapter 5.

6.4 IOP Specific Registers

6.4.1 IPCNSE - I/O Port Chip Node Specific Error Register (50 0100)

This register contains the collective error information relative to the internal operations of the IOP Module and UP Hose transactions.

Although UP Vortex errors may be detected by either of the two IPC chips, the error information from IPC-B will be transferred to IPC-A and recorded within this register. This is done so that the read data for this register is returned on bits<31:0> of the Laser System Bus.

This register also contains error information relating to transactions processed by the Down /Up HIC Chips and UP Hose Bus. Six Vortex error lines are allocated for the purpose of passing Up/Down HIC errors to IPC-A. Five lines are assigned to the UP HIC and one to the DOWN HIC. The IPC-A will interpret these error lines and record the error into the appropriate IPCNSE register bit.

Figure 6–12: IPCNSE - I/O Port Chip Node Specific Error Register (50 0100)

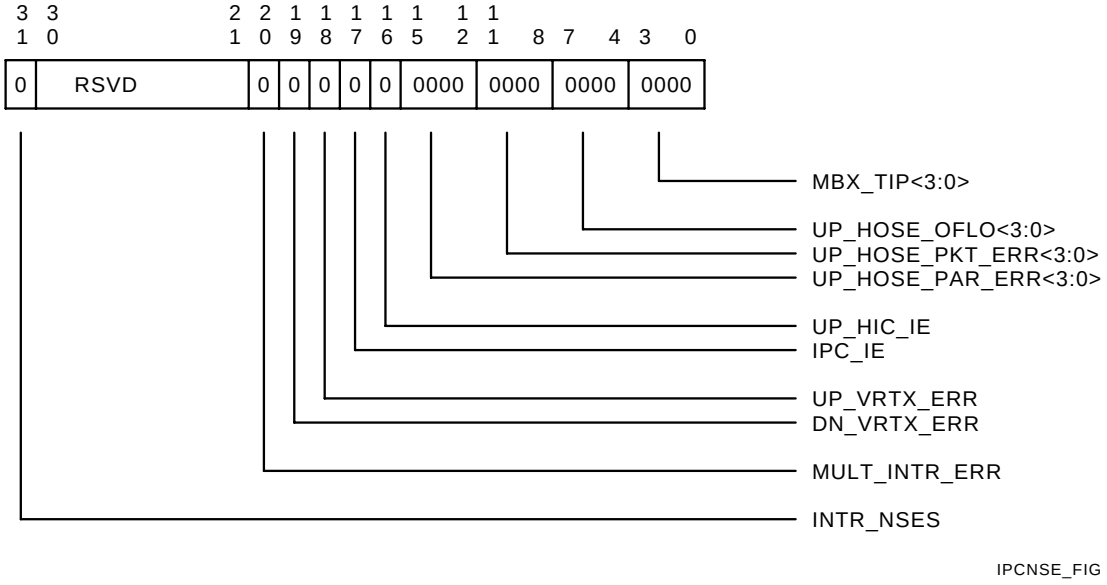


Table 6–16: IPCNSE - I/O Port Chip Node Specific Error Register Format (50 0100)

Name	Bit(s)	Type	Init	Description
INTR_NSES	31	R/W	0	INTERRUPT ON NSES This bit when set to a 1 will globally enable all error interrupt sources on the IOP Module. If an error is detected and this bit is set, the IOP will post a LEVEL 17 interrupt to an LSB CPU(s). The subsequent read of LILID3 will return the vector from the IPC Vector Register (IPCVR). When this bit is clear, no interrupt will be posted as the result of an IOP detected error. (The appropriate error bit will still be set in IPCNSE or IPCHST, however.) Note, all IOP specific error bits must be cleared before a subsequent error interrupt will be posted. LSB detected errors will always cause the IOP Module to post an LSB *interrupt*, via the assertion of LSB<ERROR> and are not maskable.
RSVD	30:21	MBZ	0	RESERVED
MULT_INTR_ERR	20	W1C	0	MULTIPLE INTERRUPT ERROR The IOP can accept up to four (4) pending interrupts from each Hose Port, one each per IPL level. Multiple interrupts at any given IPL will be ignored by the IOP and registered as an error by the setting of this error bit. Setting of this error bit will result in an interrupt posted to the LSB if enabled.
DN_VRTX_ERR	19	W1C	0	DOWN VORTEX ERROR This is a composite error bit of possible DOWN Vortex errors which the DOWN HIC can detect. The Down HIC reports these errors to the IPC via the Down Vortex signal HIC_ERROR_DN. An interrupt will be generated if interrupts are enabled in the IPCNSE.intr_nses. The possible DOWN Vortex errors are as follows: • Parity • Illegal Command • Sequence Error • Buffer Overflow • Internal HIC Error

Table 6–16 (Cont.): IPCNSE - I/O Port Chip Node Specific Error Register Format (50 0100)

Name	Bit(s)	Type	Init	Description
UP_VRTX_ERR	18	W1C	0	UP VORTEX ERROR This is a composite error bit of possible UP Vortex errors which the IPC can detect. An interrupt will be generated if IPCNSE.intr_nses mask bit is set. The possible UP Vortex Errors are as follows: • Parity • Illegal Command • Sequence Error • Buffer Overflow
IPC_IE	17	W1C	0	IPC INTERNAL ERROR This bit when set indicates that the IPC detected an illogical internal error. The internal error generally indicates a hardware problem where control logic encountered an undefined condition or conditions. The setting of this bit will generate an interrupt on the LSB if enabled by IPCNSE.intr_nses. A few of the conditions which cause the IPC_IE bit to set are listed below: • If both the slave port controller and the master port controller try to drive the same LSB data cycle. • If an LMBPR write is CNF'ed but the LMBPR wrt controller thinks the 2-deep queue for that register is full. • If a Mailbox status packet is returned but the LMBPR wrt controller is not expecting it.
UP_HIC_IE	16	W1C	0	UP HIC INTERNAL ERROR This is a composite error bit of possible internal errors which the UP HIC can detect. The UP HIC reports this error to the IPC via the Up Vortex signals HIC_ERROR_UP<4:0>. An interrupt will be generated if interrupts are enabled by IPCNSE.intr_nses. Some possible UP HIC Internal Errors are as follows: • Bad RAM locations • Packet Counter Underflow (Number of "Decrement Packet" assertions is greater than the number of packets sent.)

Table 6–16 (Cont.): IPCNSE - I/O Port Chip Node Specific Error Register Format (50 0100)

Name	Bit(s)	Type	Init	Description
UP_HOSE_PAR_ERR<3:0>	15:12	W1C	0	UP HOSE PARITY ERROR This field indicates the UP HIC detected a parity error on either the command or data cycles for the corresponding UP HOSE. The UP HIC reports this error to the IPC via the Up Vortex signals HIC_ERROR_UP<4:0>.
UP_HOSE_PKT_ERR<3:0>	11:8	W1C	0	UP HOSE PACKET ERROR This field indicates the UP HIC detected either an Illegal Command or Sequence error on the corresponding UP HOSE. The UP HIC reports this error to the IPC via the Up Vortex signals HIC_ERROR_UP<4:0>.
UP_HOSE_OFLO<3:0>	7:4	W1C	0	UP HOSE FIFO OVERFLOW This field indicates the UP HIC detected a FIFO Overflow error on the corresponding UP HOSE. The UP HIC reports this error to the IPC via the Up Vortex signals HIC_ERROR_UP<4:0>.
MBX_TIP<3:0>	3:0	W1C	0	MAILBOX TRANSACTION IN PROGRESS These bits indicate that the IOP has transmitted a Mailbox Command packet, targetting the corresponding HOSE, across the DOWN Vortex. When the corresponding Mailbox Status packet is received and processed by the IPC, the bit will be cleared. If a mailbox transaction does not complete, i.e. no Mailbox Status Packet is returned, software will have to detect the condition with a software timeout. These bits can be examined by software to determine which HOSE the packet was sent down. The LMBPR register can be read, to determine which Mailbox Data Structure was accessed. Software can then write a 1 to clear the MBX_TIP bit, so that the IPC control logic is no longer expecting a Mailbox Status packet to be returned. This will allow subsequent Mailbox Transactions to be processed. These bits may be cleared by writing a 1 directly to the bit or setting the appropriate HOSE_RST<n> bit in the IPCHST register. Note, that setting the HOSE_RST bit will result in the initialization of the entire I/O subsystem attached to the hose.

Refer to Chapter 8 for information regarding how each of the errors is handled.

6.4.2 IPCVR - I/O Port Chip Vector Register (50 0104)

This register should be loaded by software with the vector associated with IOP specific errors.

Figure 6–13: IPCVR - I/O Port Chip Vector Register (50 0104)

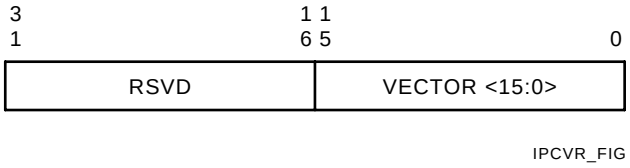


Table 6–17: IPCVR - I/O Port Chip Vector Register Format (50 0104)

Name	Bit(s)	Type	Init	Description
RSVD	31:16	MBZ	0	RESERVED
VECTOR<15:0>	15:0	R/W	0	VECTOR<15:0> These bits contain the vector that will be returned as read data when the CPU servicing an IOP error interrupt reads the LILID3 register.

6.4.3 IPCMSR - I/O Port Chip Mode Selection Register (50 0106)

This register can be used by software to select the desired mode of operation for the I/O Port Module.

This register contains a 2-bit field which controls the back-off arbitration algorithm used to avoid memory bank conflict lockout. The field controls the number of requests the IOP can make at high priority before switching to low priority arbitration for one *losing* request or two *winning* requests. The two requests at low priority are sufficient to insure that any node seeking access to a memory bank will eventually gain access. If the IOP arbs at low priority but *loses* the bus to another node, the IOP will then switch immediately back to high priority arbitration. The fact that the IOP did not gain access to the LSB means that the possiblity of memory bank lockout was resolved.)

For a more detailed explanation of how the algorithm insures access, refer to the Bus Arbitration section in Chapter 3.

There is also a bit, which can be set to a 1, to force the I/O Port Module to always arbitrate at high priority, however, this mode is ***not recommended*** for normal operation.

Figure 6–14: IPCMSR - I/O Port Chip Mode Selection Register (50 0106)

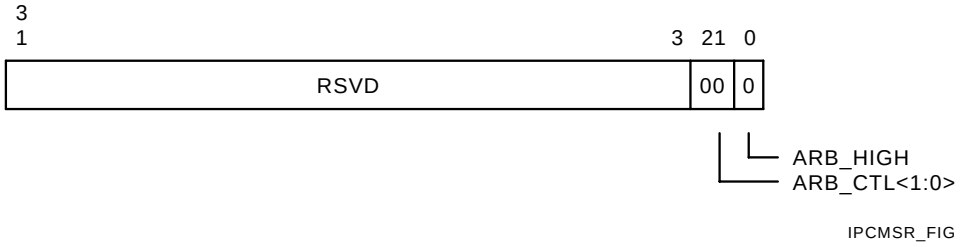


Table 6–18: IPCMSR - I/O Port Chip Mode Selection Register Format (50 0106)

Name	Bit(s)	Type	Init	Description
RSVD	31:3	MBZ	0	RESERVED

Table 6–18 (Cont.): IPCMSR - I/O Port Chip Mode Selection Register Format (50 0106)

Name	Bit(s)	Type	Init	Description
ARB_CTL<1:0>	2:1	R/W	0	ARBITRATION CONTROL These bits can be programmed to select the number of times the I/O Port Module arbitrates at high priority before switching to low priority arbitration for up to 2 requests. Note that only the request for the READ portion of a READ/MODIFY/WRITE transaction is counted as a request in this algorithm. The request to perform the WRITE portion of the READ/MODIFY /WRITE must *ALWAYS* use high priority arbitration (REQ 5) and is not counted in the back off algorithm.

The ARB_CTL field is encoded as follows

00	Arbs six times at high priority, then switches to low priority arbitration for up to 2 requests. This is the DEFAULT at power up or initialization.
01	Arbs four times at high priority, then switches to low priority arbitration for up to 2 requests.
10	Arbs twice at high priority, then switches to low priority arbitration for up to 2 requests.
11	Arbs at low priority always.

It is *STRONGLY* recommended that 11 (arb low always) *NOT* be selected during normal operation as I/O performance will be adversely affected.

ARB_HIGH	0	R/W	0	This bit when set to a 1, forces the I/O Port Module to always arbitrate at high priority (REQ 5). When set, this bit overrides the ARB_CTL field.

It is *STRONGLY* recommended that the ARB_HIGH bit *NOT* be used during normal operation as potential lockouts from memory could occur.

6.4.4 IPCHST - I/O Port Chip Hose Status Register (50 0108)

This register contains status information relating to each HOSE. In the event an error occurs, the HOSE_ERR bit in IPCNSE will be set and status in the IPCHST will be locked for the failing HOSE.

Figure 6–15: IPCHST - I/O Port Hose Status Register (50 0108)

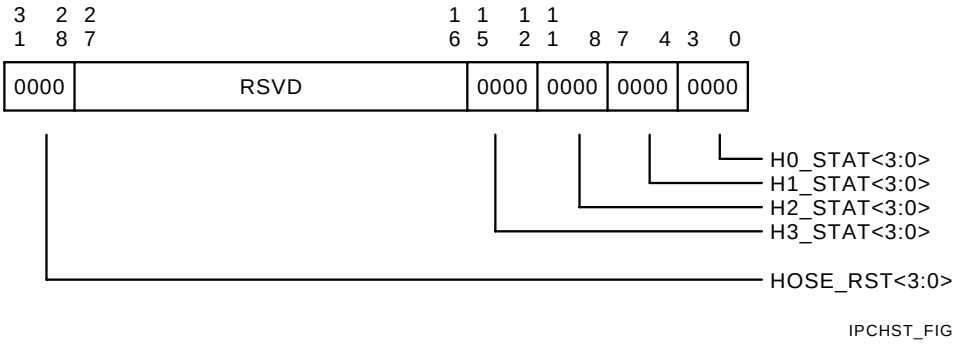


Table 6–19: IPCHST - I/O Port Hose Status Register Format (50 0108)

Name	Bit(s)	Type	Init	Description
HOSE_RST<3:0>	31:28	WO	0	HOSE RESET<3:0> When these bits are written to a 1, the IPC generates the Hose signal DNRST<n> which resets the attached I/O subsystem. Reads to these bit positions always return a 0.
RSVD	27:16	MBZ	0	RESERVED
H3_STAT<3:0>	15:12	*	*	HOSE3 STATUS<3:0> These bits indicate an I/O Adapter's presence and ability to accept transactions, refer to Section 6.4.4.1 for details. The bits are encoded as follows: 1. H3_STAT<3>- HOSE signal *PWROK* transitioned. This bit is W1C. 2. H3_STAT<2>- HOSE signal *CBLOK* current level. This bit is RO. 3. H3_STAT<1>- HOSE signal *PWROK* current level. This bit is RO. 4. H3_STAT<0>- HOSE signal *ERROR* current level. This bit is RO.

Table 6–19 (Cont.): IPCHST - I/O Port Hose Status Register Format (50 0108)

Name	Bit(s)	Type	Init	Description
H2_STAT<3:0>	11:8	*	*	HOSE2 STATUS<3:0> These bits indicate an I/O Adapter's presence and ability to accept transactions. Refer to the table for HOSE3 STATUS above.
H1_STAT<3:0>	7:4	*	*	HOSE1 STATUS<3:0> These bits indicate an I/O Adapter's presence and ability to accept transactions. Refer to the table for HOSE3 STATUS above.
H0_STAT<3:0>	3:0	*	*	HOSE0 STATUS<3:0> These bits indicate an I/O Adapter's presence and ability to accept transactions. Refer to the table for HOSE3 STATUS above.

6.4.4.1 HOSEn STATUS<3:0> - Detailed Description

1. Hx_STAT<3>, HOSE PWROK Transitioned
This bit is latched when ever the associated HOSE signal PWROK transitions. A PWROK transition from 0-1 will indicate a Power-Up, while a PWROK transition of 1-0 indicates a Power-Down. Hx_STAT<1> can then be read to determine the reason why this bit set. The setting of this bit will result in an IOP IPL17 interrupt to be posted on the LSB if enabled.

2. Hx_STAT<2>, HOSE CBLOK
This bit is derived from the HOSE signal CBLOK. If the cable is good and connected properly and the I/O Adapter is plugged into its card cage, then this bit will be a 1. The setting of this bit will *not* result in any LSB interrupt.

3. Hx_STAT<1>, HOSE PWROK
This bit is derived from the HOSE signal PWROK and reflects its current level. If the Hose cable is connected properly to the I/O Adapter and has sufficient power to process commands, then this bit will be a 1. The transition of this bit from either 1-0 or 0-1, will result in Hx_STAT<3> to set. This bit can then be read to determine why Hx_STAT<3> is set. This bit has *no* affect on IOP interrupts.

4. Hx_STAT<0>, HOSE ERROR
This bit is derived from the HOSE signal ERROR. The I/O Adapters will drive this signal when they detect any fatal error which prevents them from using normal Up Hose packet protocols. The I/O Adapters will continue to drive this signal until cleared by an associated HOSE_RST<3:0>. The setting of this bit will result in an IOP IPL17 interrupt to be posted on the LSB if enabled.

6.4.5 IPCDR - I/O Port Chip Diagnostic Register (50 010A)

This register can be programmed by diagnostics, to force errors on the LSB and Vortex buses for the IOP to detect. Hose errors can also be forced but is a function of the Loopback feature. Refer to Chapter 9 for details.

Laser bus errors are transmitted on the LSB, and are detected by the IOP when the signals are received into the IPC. Note that all other nodes that are monitoring the LSB will also detect the errors.

This register can also be used to force errors to be generated across the DOWN and UP Vortex Buses.

Figure 6–16: IPCDR - I/O Port Chip Diagnostic Register (50 010A)

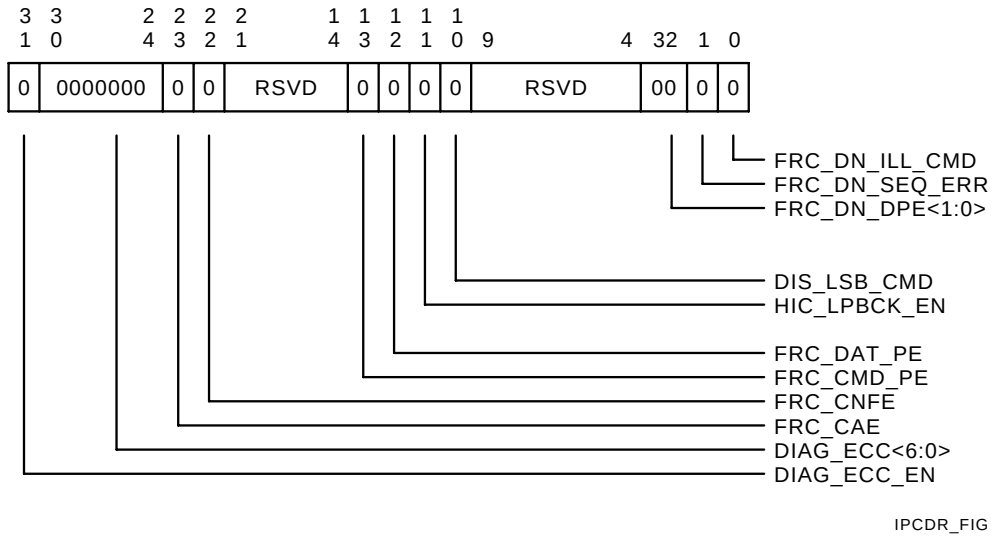


Table 6–20: IPCDR - I/O Port Chip Diagnostic Register Format (50 010A)

Name	Bit(s)	Type	Init	Description
DIAG_ECC_EN	31	R/W	0	DIAGNOSTIC ECC ENABLE When this bit is set to a 1, the ECC generation logic will select DIAG_ECC<6:0> (IPCDR<30:24>) as the ECC bits transmitted on the LSB. This feature can be used to force correctable and uncorrectable ECC errors. Note that the "failing" ECC code is driven on the LSB and all other nodes watching the bus will detect the error.

Table 6–20 (Cont.): IPCDR - I/O Port Chip Diagnostic Register Format (50 010A)

Name	Bit(s)	Type	Init	Description
DIAG_ECC<6:0>	30:24	R/W	0	DIAGNOSTIC ECC<6:0> These bits can be written by diagnostics with a substitute ECC pattern which will allow the forcing of correctable and uncorrectable ECC errors to be transmitted on the LSB. This field is only enabled when DIAG_ECC_EN is set to a 1. Note that the "failing" ECC code is driven on the LSB and all other nodes watching the bus will detect the error.
FRC_CAE	23	WO	0	FORCE CA ERROR When this bit is written by diagnostics, the IOP logic will assert CA on the LSB during an illegal cycle. The assertion of CA in an illegal cycle will cause each node watching the bus, including the IOP, to set the CAE bit in the LBER.
FRC_CNFE	22	WO	0	FORCE CNF ERROR When this bit is written by diagnostics, the IOP logic will assert CNF on the LSB during an illegal cycle. The assertion of CNF in an illegal cycle will cause each node watching the bus, including the IOP, to set the CNFE bit in the LBER. Writing this bit generates the CNF error for one transaction only.
RSVD	21:14	MBZ	0	RESERVED
FRC_CMD_PE	13	R/W	0	FORCE COMMAND PARITY ERROR When set, this bit forces the IPC to assert bad parity on the LSB during a Command Cycle.
FRC_DAT_PE	12	R/W	0	FORCE DATA PARITY ERROR When set, this bit forces the IPC to assert bad parity on the LSB during a CSR Data Cycle.
HIC_LPBCK_EN	11	R/W	0	HIC LOOPBACK ENABLE When set, enables the IOP to internally loopback Mailbox Command packets between the DN HOSE and UP HOSE. When set, the IPC will automatically disable all attached I/O Adapter modules by driving the Hose Error signal on all UP Hoses. Refer to Chapter 9 for loopback programming details.

Table 6–20 (Cont.): IPCDR - I/O Port Chip Diagnostic Register Format (50 010A)

Name	Bit(s)	Type	Init	Description
DIS_LSB_CMD	10	R/W	0	DISABLE LSB COMMAND TRANSMISSION This feature when enabled, allows diagnostics to "fill" all UP Hose data path buffers in the IPC and HIC. Once the bit is cleared, the IOP will process all the transactions in its buffers. Refer to Chapter 9 for recommended guidelines to follow for the use of this bit.
RSVD	9:4	MBZ	0	RESERVED
FRC_DN_DPE<1:0>	3:2	R/W	0	FORCE DOWN DATA PARITY ERROR Either of these bits being set to a 1, forces bad parity on the DOWN Vortex bus as it exits the IPC. FRC_DPE<1> corresponds to the parity bit for longword<1> from IPC-B, FRC_DPE<0> corresponds to the parity bit for longword<0> from IPC-A.
FRC_DN_SEQ_ERR	1	R/W	0	FORCE DOWN SEQUENCE ERROR This bit when set to a 1, causes the IPC to send an incorrect number of valid cycles during the transmission of a packet on the Down Vortex bus.
FRC_DN_ILL_CMD	0	R/W	0	FORCE DOWN ILLEGAL COMMAND This bit when set to a 1, causes the IPC to send an Illegal Command during the transmission of a packet on the Down Vortex bus.

CHAPTER 7

IOP POWER-UP AND INITIALIZATION

7.1 Initialization General

The IOP can be initialized in two ways (collectively these are termed IOP Reset).

1. System Reset - Whenever the LSB RESET signal asserts, the IOP will be initialized. LSB RESET can be asserted by any node or via the front panel key switch.
2. Node Reset - The IOP may be reset, by software or diagnostics, by writing a 1 to the NRST bit (LCNR<30>).

In addition to the IOP being reset, the I/O Port Chip will generate reset signals to be sent down all the Hoses to initialize the attached I/O subsystem. Note, however, that the IOP is expected to power up before the I/O subsystem(s). This means the I/O subsystems must provide a mechanism for initialization on power up.

As a result of an IOP Reset, the IOP aborts any transaction currently in progress and places itself in a defined state ready to receive commands from software. It will not request interrupts, nor read or write memory until requested to do so by software. When either a System Reset or a Node Reset is initiated, the attached I/O subsystem is also reset as defined by the appropriate I/O Bus Adapter specification. Additionally, individual resets of the attached I/O subsystems can be accomplished by writing a 1 to the desired Hose Reset Bit in the I/O Port Chip Hose Status Register.

If the I/O Bus Adapter and I/O subsystem are powered down the IOP will detect the change in state of the HOSE signal PWROK (transition from 1-0). If the bit INTR_NSES (Interrupt on Node Specific Error Summary) is set in the I/O Port Chip Node Specific Error Register, then the IOP will post an interrupt to an LSB CPU(s).

Once the IOP and I/O subsystem(s) have been initialized, software can select the desired operating mode (refer to the IPCMSR description) and select the desired target CPU(s) for servicing interrupts (refer to the LCPUMASK description). The default operating mode for the IOP allows for six bus requests to be made at high priority, then up to two requests must be made at low priority. The number of requests at high priority is programmable via the IPCMSR. The two requests at low priority are sufficient to prevent memory bank lockout and are not programmable.

7.2 Power-Up Testing

After a System Reset, a single (*primary*) CPU will test the IOP and attached I/O subsystem(s). Other CPUs on the system may also be utilized to verify I/O interrupt capability and to perform additional Mailbox transactions.

After completion of the power-up diagnostic, the primary CPU will insure that all error bits are cleared, all interrupts have been serviced, and all writable register bits are set to their default values. Bit<31> (STF) of the IOP's Laser Configuration Register can then be set indicating that the IOP successfully passed the power-up diagnostic.

Note that no diagnostic testing will take place as the result of a Node Reset.

CHAPTER 8

IOP ERROR HANDLING

The primary error handling strategy for Laser is one of error detection, not error correction or hardware error recovery. It has been determined in recent years that the hardware platform seldom has single-bit soft errors. When errors occur, they tend to be hard errors. The I/O Port Controller Module (IOP) follows this same strategy. Error detection at the earliest opportunity will insure system integrity and good fault isolation to a field replaceable unit. The architecture is such that the IOP shall detect problems at the earliest opportunity and send an error interrupt if enabled. It is left up to the operating system software to determine the cause of the error by examining the IOP CSRs.

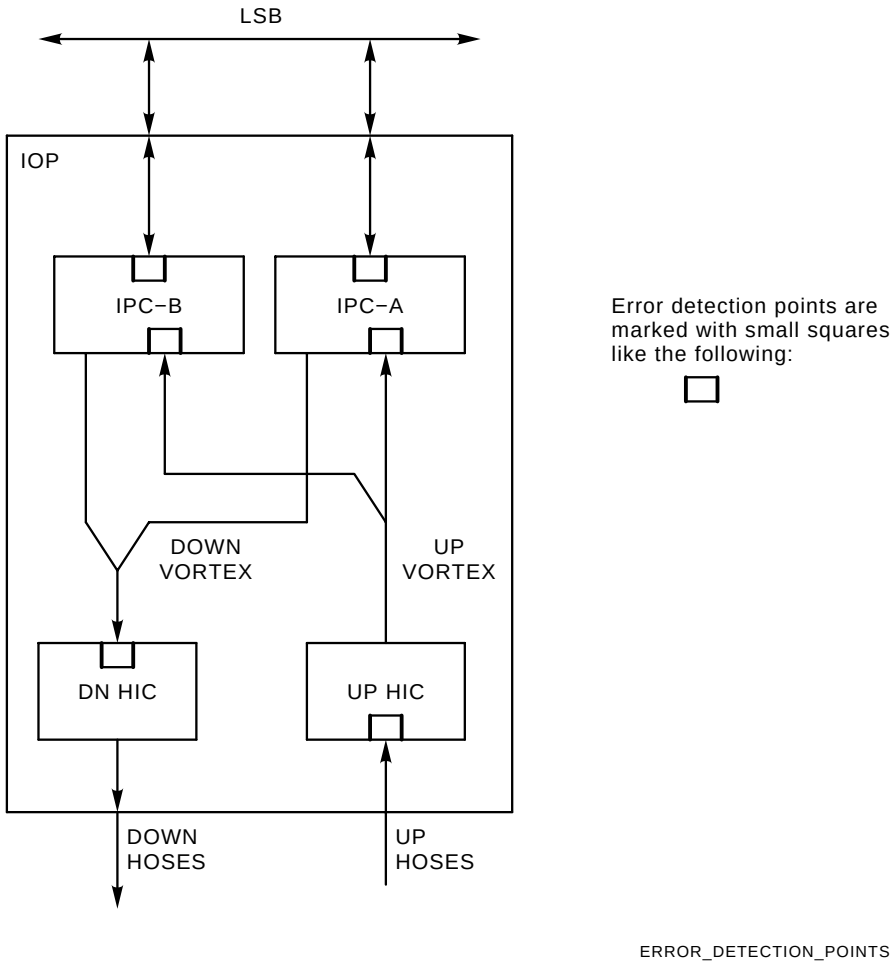
The IOP can detect various types of errors on the LSB, Vortex, and Hoses. There are six major error detection points within the IOP as shown in Figure 8–1. The IPCs detect UP Vortex errors and Laser System Bus errors. The UP HIC detects UP HOSE errors, and the DOWN HIC detects DOWN Vortex errors. The following sections describe each of the error conditions that the IOP can detect on these buses.

8.1 Laser System Bus (LSB) Errors Detectable by IPCs

The Laser System Bus (LSB) has a number of error detection features as well as limited error correction. All command/address cycles and CSR data cycles are parity protected. All memory data cycles are ECC protected. Because the bus arbitration relies upon distributed arbiters, transmit check logic is also an integral part of the LSB interface chips to detect bus collisions.

If the IOP transmits a DMA memory command onto the LSB, but detects an error on that command (CPE, NXAE, CTCE or DTCE), the remainder of the transaction is aborted. This means, if the transaction is a DMA Write, the associated data cycles for that command will not be driven on the LSB. This will result in Uncorrectable ECC Errors being detected by all nodes monitoring the bus. If the transaction is a DMA Read, the DMA READ RETURN DATA packet which gets transmitted across the Vortex will have the ERROR bit set, indicating the data is invalid. Refer to the HOSE chapter and the Vortex appendix for more detail.

Figure 8–1: Error Detection Points



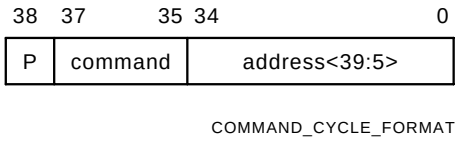
The Laser Bus Error Register (LBER) is the collection point for all Laser System Bus error indicators. Additionally, the LBECR1:0 registers contain Command/Address information pertaining to failing commands, if a Command Parity Error (CPE), CSR Data Parity Error (CDPE), Correctable ECC Error (CE) or Uncorrectable ECC Error (UCE) is detected. If an ECC error is detected, then syndrome information is logged in the LBESRx registers.

8.1.1 LSB Command Cycle Parity Error

Figure 8–2 depicts the command cycle format. A single, odd parity bit is carried across the 35-bit address and 3-bit command fields. A command/address cycle with a parity error is rejected by the lack of confirmation and by assertion of the LASER ERR signal. All LSB nodes that detect a command/address cycle that is in error must report the error by setting the LBER.cpe bit and asserting LASER ERR. If the LBER.cpe bit is already set then the LBER.cpe2 bit will be set.

If the IOP was the transmitter of the failing command, then the LBER.tde bit (Transmitter During Error) will also be set by the IOP.

Figure 8–2: Command Cycle Format



8.1.2 LSB CSR Data Cycle Parity Error

In CSR transactions, the first data cycle contains the CSR Read or Write data. Data bit D<38> is driven with odd parity which is computed across D<37:0>. All LSB nodes that detect a CSR Data Parity Error must report the error by setting the LBER.cdpe bit and asserting LASER ERR. If the LBER.cdpe bit is already set then the LBER.cdpe2 bit will be set.

If the IOP was the transmitter of the failing CSR data, then the LBER.tde bit will also be set by the IOP.

8.1.3 LSB Memory Data Cycle ECC Error

The LSB ECC coding scheme consists of 7 bits of ECC for every longword (32-bits) of data. This ECC algorithm provides double-bit error detection and single-bit error correction. If a data cycle has a correctable ECC error, the ECC logic automatically makes the correction. For more detailed information regarding ECC, please refer to Chapter 2, LSB Operation, of the LSB Specification.

Error logging and reporting of Correctable ECC Errors can be enabled/disabled by the setting or clearing of the LCNR.ceen bit. All LSB nodes that are enabled to report CE errors must set the LBER.ce bit and assert the LASER ERR signal if a CE error is detected. If the LBER.ce bit is already set, then the LBER.ce2 bit will be set.

If the IOP was the transmitter of the failing data cycle(s), then LBER.tde will also be set by the IOP.

Error logging and reporting of Uncorrectable ECC Errors is always enabled. All LSB nodes that detect UCE errors must set the LBER.uce bit and assert the LASER ERR signal. If the LBER.uce bit is already set, then the LBER.uce2 bit will be set.

If the IOP was the transmitter of the failing data cycle(s), then LBER.tde will also be set by the IOP.

NOTE

There are certain error conditions when the IOP will be the source of Uncorrectable ECC Errors being *driven* on the LSB, yet will not set its LBER.tde bit. These instances are: 1) when an UP Vortex Error is detected on a DMA WRITE packet after the command/address is driven onto the LSB, 2) When the IOP detects an Uncorrectable ECC Error for on the Read data return as part of a Read-Modify-Write operation. The reason for this behavior is because the command has already been issued on the LSB and but some portion of the data is incorrect and the data cannot be selectively written.

If the IOP detects a UCE error on DMA read data, a DMA READ RETURN DATA packet with the ERROR bit set will be returned across the Vortex. Refer to the HOSE chapter and Vortex appendix for more detail.

All LSB nodes which report the detection of ECC errors must also retain the failing syndrome information. This is done by loading the syndrome code into the appropriate Laser Bus Error Syndrome Register (LBESRx) for the given data cycle, then *locking* all the LBESRx registers. These registers will remain locked until the LBER.ce and LBER.uce bits are both clear. Please refer to the LSB Specification for more detail regarding ECC error syndromes.

The command/address, commander ID, LSB confirmation state, and failing data cycle number (0 to 3) are locked in the Laser Bus Error Command Registers (LBECR0 and LBECR1). These registers are also *unlocked* when the following LBER bits are all clear: CPE, CDPE, CE and UCE.

8.1.4 LSB Transmit Check Error

During LSB cycles in which the IOP is responsible for driving the data lines, the IOP compares the data received from the bus with the data that it drove onto the bus. In this way, bus collisions with other nodes can be detected.

When the IOP drives a Command/Address onto the LSB or returns CSR Read Data onto the LSB, the received LSB data bits<63:0> and received LSB ECC bits<13:0> are checked for the proper assertion value. If there is a discrepency, indicating a possible bus collision, the Data Transmit Check Error (DTCE) bit in the LBER is set and LASER ERR is asserted. When the IOP transmits memory data, the entire 128-bit data path and 28-bit ECC is checked for proper assertion value.

Anytime the IOP asserts a control line, the IOP checks the receive version of the signal to insure that it was asserted on the bus. If the signal is received *deasserted* then a bus collision may have occurred. The IOP shall set the Control Transmit Check Error (CTCE) bit in the LBER and assert LASER ERR.

8.1.5 LSB Control Line Errors

Bus "state" is very important to system integrity because of the distributed nature of the Laser System Bus arbitration. If any control signals are asserted when they should *not* be asserted or are not asserted when they *should* be, that fact must be logged and reported.

The IOP implements three such bits in the LBER to indicate the detection of these possible bus synchronization errors. They are Command Address Error (CAE), Confirm Error (CNFE), and Stall Error (STE) in the LBER. Any illegal assertion of these signals will result in the IOP setting the corresponding bit and asserting LASER ERR.

The Error Line Asserted (E) bit in the LBER is for error logging purposes only. Any node that sees the Laser Error line asserted shall set the E bit in its bus error register.

If an IOP Command/Address targets an unimplemented address, i.e. no Laser CNF response, the IOP sets the Non-Existent Address Error (NXAE) bit in the LBER and asserts LASER ERR.

8.1.6 Summary of Conditions That Cause the IOP to Assert Laser ERR

Anytime the IOP detects an LSB error condition, the error type will be logged in the LBER register and an LSB CPU(s) will be notified via the IOP's assertion of LASER ERR. LSB error conditions that are detectable by the IOP are summarized in Table 8–1.

There are no IOP *node specific* error conditions under which the IOP will assert LASER ERR. Reporting of IOP specific errors is done via IPL17 interrupts, which can be enabled by setting the IPCNSE.intr_nses (interrupt enable) bit.

Table 8–1: IOP ERR Assertion Summary

Associated Error Bit	Description
CE	The IOP detected a correctable ECC error during a memory data cycle.
UCE	The IOP detected an uncorrectable ECC error during a memory data cycle.
CPE	The IOP detected a parity error during a command cycle.
CDPE	The IOP detected a parity error during the first data cycle of a CSR transaction.
STE	The IOP detected an illegal STALL assertion.
CNFE	The IOP detected an illegal CNF assertion.
NXAE	The IOP was commander when CNF was not received for the command.
CAE	The IOP detected CA asserted when it should not have been asserted.
DTCE	The IOP was responsible for driving data when a transmit check error was detected.
CTCE	The IOP detected a transmit check error on control lines that it was driving on the bus.

8.2 UP Vortex Bus Errors Detectable by IPCs

The IPCs can detect 4 different types of errors on the UP Vortex Bus. When any of the 4 errors occur, the IPCs set the UP_VRTX_ERR (IPCNSE<18>) bit. A summary of the Up Vortex error conditions are listed in Table 8–2 and are explained in more detail in the sections that follow.

Table 8–2: Up Vortex Bus Errors Detectable by IPCs

Up Vortex Error	Description
Parity Error	IPC-A or IPC-B detects a parity error during a valid cycle (i.e. HIC_UP_VTX_VALID_n asserted)
Illegal Command	The IPCs detect an illegal command code in the command field of the Up Vortex during a valid cycle
Sequence Error	The IPCs detect an UP Vortex packet that has good parity but contains the wrong number of cycles, or the LENGTH code in the 1st cycle of the packet is illegal
Buffer Overflow	The IPCs receive an Up Vortex Packet and the IPC buffers are already full

8.2.1 UP Vortex Parity Error

There are two "odd parity" bits protecting the UP Vortex Bus. One parity bit (HIC_UP_PAR<0>) protects the 36 bits of data (HIC_UP_DATA<31:0> plus HIC_UP_CMD_A<3:0>) that go to IPC-A. The other parity bit (HIC_UP_PAR<1>) protects the 36 bits of data (HIC_UP_DATA<63:32> plus HIC_UP_CMD_B<3:0>) that go to IPC-B.

If an UP Vortex Parity error occurs on the 1st or 2nd cycle of an UP Vortex packet, the IPCs will never issue an LSB transaction for that packet. However, if the UP Vortex Parity error occurs on the 3rd or later UP Vortex cycle, the IPCs will still issue the LSB command cycle, but will default the LSB (i.e. force bad parity/ECC) during the data cycles.

In all cases, if a Parity Error is detected, the UP_VRTX_ERR in the IPCNSE register shall be set and an error interrupt generated if enabled.

8.2.2 UP Vortex Illegal Command Error

An Illegal Command Error occurs when the command field (HIC_CMD_UP_n<3:0>) does NOT contain a valid command code during the 1st cycle of an UP Vortex Packet, even though the parity was good.

If an Illegal Command Error is detected by the IPCs, the IPCs will never issue an LSB transaction for that packet. The IPCs will, however, set the UP_VRTX_ERR in the IPCNSE register and generate an error interrupt if interrupts are enabled.

8.2.3 UP Vortex Sequence Error

A Sequence Error occurs on the UP Vortex Bus whenever an UP Vortex packet contains the wrong number of UP Vortex cycles, or whenever the LEN code field of the 1st cycle contains and illegal length code.

There are two ways that the IPCs check for Sequence Errors on the UP Vortex. The first is to count the number of UP Vortex cycles in a packet and if it doesn't match the number of expected cycles for that particular packet type, the packet has a Sequence Error. The second way is to read the LEN code field during the 1st cycle of a packet. If it is an illegal LEN code, the packet has a Sequence Error.

If a Sequence Error occurs on the 1st or 2nd cycle of an UP Vortex packet, the IPCs will never issue an LSB transaction for that packet. However, if the Sequence Error occurs on the 3rd or later UP Vortex cycle, the IPCs will still issue the LSB command cycle, but will default the LSB (i.e. force bad parity/ECC) during the data cycles.

In all cases, if a Sequence Error is detected, the UP_VRTX_ERR in the IPCNSE register shall be set and an error interrupt generated if enabled.

8.2.4 UP Vortex Buffer Overflow

The IPCs have two buffers that are be used to store UP Vortex packets. If both these buffers are full and another packet is sent across the UP Vortex Bus, the IPCs will detect a Buffer Overflow condition.

If a Buffer Overflow occurs, the UP Vortex packet that caused the overflow will never be issued on the LSB. The IPCs will, however, set the UP_VRTX_ERR in the IPCNSE register and generate an error interrupt if interrupts are enabled.

8.3 DOWN Vortex Bus Errors Detectable by DOWN HIC

The DOWN HIC can detect 4 different types of errors on the DOWN Vortex Bus, When any of these 4 errors occur, the DOWN HIC sets the DN_VRTX_ERR (IPCNSE<19>) bit in IPC-A. A summary of the DOWN Vortex error conditions are listed in Table 8–3 and are explained in more detail in the sections that follow.

NOTE

Besides setting the DN_VRTX_ERR bit whenever one of the 4 DOWN Vortex Bus errors is detected, the DOWN HIC will also set the DN_VRTX_ERR bit whenever it detects an internal error condition that occurs in the DOWN HIC gate array itself.

Table 8–3: DOWN Vortex Bus Errors Detectable by DOWN HIC

DOWN Vortex Error	Description
Parity Error	The DOWN HIC detects a parity error during a valid cycle (i.e. IPC_DN_VTX_VALID asserted)
Illegal Command	The DOWN HIC detect an illegal command code in IPC_DN_DATA<13:12> of the DOWN Vortex during the 1st cycle of a packet
Sequence Error	The DOWN HIC detects an DOWN Vortex packet that has good parity but contains the wrong number of cycles, or the LENGTH code in the 1st cycle of the packet is illegal
Buffer Overflow	The DOWN HIC receives a DOWN Vortex Packet and the DOWN HIC buffers for that hose are already full

8.3.1 DOWN Vortex Parity Error

There are two "odd parity" bits protecting the DOWN Vortex Bus. One parity bit (IPC_DN_PAR<0>) protects the 32 bits of data (IPC_DN_DATA<31:0>) that come from IPC-A. The other parity bit (IPC_DN_PAR<1>) protects the 32 bits of data (IPC_DN_DATA<63:32>) that come from IPC-B.

If a DOWN Vortex Parity error is detected by the DOWN HIC, the bad packet is never sent across the DOWN HOSE. The DOWN HIC will set the DN_VRTX_ERR in the IPCNSE register causing an error interrupt if interrupts are enabled. The IOP must then be reset (eg NODE_RESET) in order to get the DOWN HIC back in sync and functioning properly.

8.3.2 DOWN Vortex Illegal Command Error

An Illegal Command Error occurs when the command field (IPC_DN_DATA<13:12> during the 1st cycle of a DOWN Vortex packet) does NOT contain a valid command code, even though the parity was good.

If an Illegal Command Error is detected by the DOWN HIC, the packet will never be issued across the DOWN HOSE. The DOWN HIC will set the DN_VRTX_ERR in the IPCNSE register causing an error interrupt if interrupts are enabled. The IOP must then be reset (eg NODE_RESET) in order to get the DOWN HIC back in sync and functioning properly.

8.3.3 DOWN Vortex Sequence Error

A Sequence Error occurs on the DOWN Vortex Bus whenever a DOWN Vortex packet contains the wrong number of DOWN Vortex cycles, or whenever the LEN code field of the 1st cycle contains and illegal length code.

There are two ways that the DOWN HIC checks for Sequence Errors on the DOWN Vortex. The first is to count the number of DOWN Vortex cycles in a packet and if it doesn't match the number of expected cycles for that particular packet type, the packet has a Sequence Error. The second way is to read the LEN code field during the 1st cycle of a packet. If it is an illegal LEN code, the packet has a Sequence Error.

If a Sequence Error is detected by the DOWN HIC, the packet will never be issued across the DOWN HOSE. The DOWN HIC will set the DN_VRTX_ERR in the IPCNSE register causing an error interrupt if interrupts are enabled. The IOP must then be reset (eg NODE_ RESET) in order to get the DOWN HIC back in sync and functioning properly.

8.3.4 DOWN Vortex Buffer Overflow

The DOWN HIC has four buffers per DOWN HOSE that are be used to store DOWN Vortex packets. If all four buffers for a particular HOSE are full and another packet destined for that HOSE is sent across the DOWN Vortex Bus, the DOWN HIC will detect a Buffer Overflow condition.

If a Buffer Overflow occurs, the DOWN Vortex packet that caused the overflow will never be issued on the DOWN HOSE. The DOWN HIC will, however, set the DN_VRTX_ERR in the IPCNSE register and generate an error interrupt if interrupts are enabled. The IOP must then be reset (eg NODE_RESET) in order to get the DOWN HIC back in sync and functioning properly.

8.4 UP HOSE Errors Detectable by UP HIC

The UP HIC can detect 3 different types of errors on the UP HOSE. When any of these 3 errors occur, the UP HIC sets the appropriate bit in the IPCNSE register in IPC-A. A summary of the UP HOSE error conditions is listed in Table 8–4 (along with the IPCNSE bits that they affect) and are explained in more detail in the sections that follow.

NOTE

Besides being able to detect and set error bits for the 3 error conditions mentioned, the UP HIC will also set the UP HIC Internal Error bit (UP_HIC_IE) in the IPCNSE register whenever it detects an internal error condition that occurs in the UP HIC gate array itself.

Table 8–4: UP HOSE Errors Detectable by UP HIC

UP HOSE Error	Description
Parity Error	The UP HIC detects a parity error during a valid cycle (i.e. HOSEn_UP_DATA_VALID asserted)
Packet Error	The UP HIC detects an UP HOSE packet that has good parity but contains the wrong number of cycles, or the LENGTH code in the 1st cycle of the packet is illegal, or the COMMAND code in the 1st cycle of the packet is illegal
Buffer Overflow	The UP HIC receives an UP HOSE Packet and the UP HIC buffers for that hose are already full

8.4.1 UP HOSE Parity Error

There is an "odd parity" bit protecting each UP HOSE Bus. The parity bit (HOSEn_UP_PAR where "n" equals the HOSE number) protects the 36 bits of data (HOSEn_UP_DATA<31:0> plus HOSEn_UP_CTRL<3:0>) that come from the I/O Adapter Module.

If an UP HOSE Parity error is detected by the UP HIC, the bad packet is never sent across the UP Vortex. The UP HIC will set the appropriate UP_HOSE_PAR_ERR<3:0> bit in the IPCNSE register corresponding to the HOSE with the parity error. Setting this bit will cause an error interrupt if interrupts are enabled.

Even though the packet is lost, the IOP can continue to function from this point.

8.4.2 UP HOSE Packet Error

An UP HOSE Packet Error occurs whenever the UP HIC detects either an illegal command or a sequence error on one of the four UP HOSEs.

8.4.2.1 UP HOSE Packet Error caused by illegal command

An illegal command on the UP HOSE occurs when the control field (HOSEn_UP_CTRL<3:0>) does NOT contain a valid control code, even though the parity was good.
If an illegal command is detected by the UP HIC, the packet will never be issued across the UP Vortex. The UP HIC will set the appropriate UP_HOSE_PKT_ERR<3:0> bit in the IPCNSE register corresponding to the HOSE with the illegal command error. Setting this bit will cause an error interrupt if interrupts are enabled.

Even though the packet is lost, the IOP can continue to function from this point.

8.4.2.2 UP HOSE Packet Error caused by sequence error

A sequence error occurs on the UP HOSE Bus whenever an UP HOSE packet contains the wrong number of UP HOSE cycles, or whenever the LEN code field of the 1st cycle contains and illegal length code.
There are two ways that the UP HIC checks for sequence errors on the UP HOSE. The first is to count the number of UP HOSE cycles in a packet and if it doesn't match the number of expected cycles for that particular packet type, the packet has a sequence error. The second way is to read the LEN code field during the 1st cycle of a packet. If it is an illegal LEN code, the packet has a sequence error.

If a sequence error is detected by the UP HIC, the packet will never be issued across the UP Vortex. The UP HIC will set the appropriate UP_HOSE_PKT_ERR<3:0> bit in the IPCNSE register corresponding to the HOSE with the sequence Error. Setting this bit will cause an error interrupt if interrupts are enabled.

Even though the packet is lost, the IOP can continue to function from this point.

8.4.3 UP HOSE Buffer Overflow

The UP HIC has four buffers per UP HOSE that are be used to store UP HOSE packets. If all four buffers for a particular HOSE are full and another packet destined for that HOSE is sent across the UP HOSE Bus, the UP HIC will detect a Buffer Overflow condition.

If a Buffer Overflow occurs, the UP HOSE packet that caused the overflow will never be issued on the UP Vortex. The UP HIC will, however, set the appropriate UP_HOSE_ OFLO<3:0> bit in the IPCNSE register corresponding to the HOSE with the Buffer Overflow Error. Setting this bit will cause an error interrupt if interrupts are enabled.

Even though the packet is lost, the IOP can continue to function from this point.

8.5 Internal Errors Detectable on the IOP Module

Both the IPC and HIC gate arrays contain logic that allow them to detect internal errors. Internal Errors are errors inside the gate array that should have be impossible to occur unless something went very wrong, such as a hard failure in the gate array itself.

Internal Errors detected by the IPCs and HICs include illegal controller states; detection of both slave and master ports in the IPCs trying to drive the same LSB cycle; detection that an LMBPR write was CNF'ed but the LMBPR controller thinks the 2-deep queue for that register is full; detection of a Mailbox Status Return packet on the UP Vortex Bus but there was no Mailbox transaction pending; detection of an Illegal packet while reading the packet out of a buffer after the packet was previously checked to be VALID when it was loaded into the buffer (eg bad RAM); detection of a Packet Counter Underflow (ie. number of DEC_PKT_CNT assertions is greater than the number of packets sent).

Whenever the IPCs detect an Internal Error, they will set the IPC_IE bit in the IPCNSE register.

Whenever the UP HIC detects an Internal Error, it will set the UP_HIC_IE bit in the IPCNSE register.

Whenever the DOWN HIC detects an Internal Error, it will set the DN_VRTX_ERR bit in the IPCNSE register. This is because DOWN HIC Internal Error was merged with DN_ VRTX_ERR in order to save an extra pin on the HICs and IPCs since all 4 DOWN Vortex errors and Internal error are fatal to the DOWN HIC and require the IOP to be reset at that point.

CHAPTER 9

DIAGNOSTIC FEATURES

9.1 Overview

This chapter will consolidate the Diagnostic features of the IOP and demonstrate the programming of those features. The Chapter will be organized according to the IOP's bus interconnects, ie the LSB, Vortex and the Hose. Each IOP error condition will be specified and the *forced* programming of it described.

9.2 Laser System Bus Diagnostic Features

All LSB modules must monitor the Laser Bus for errors which it or other modules could generate. LSB detected errors are reported to a CPU node via the assertion of the LSB<ERR> signal.

The assertion of LSB<ERR> will eventually result in an interrupt/trap/exception at a CPU node. It is then the responsibility of this CPU node to read the LSB error registers and take the appropriate actions.

In dealing with any of the following IOP forced Laser Bus error conditions, an appropriate interrupt/trap/exception handler must be structured to *recover* from all forced Laser Bus errors. It should be noted, that Laser Bus error handling procedures are processor specific. Refer to the specific error handling procedures for your Laser Bus implementation.

Table 9–1 summarizes all IOP detectable LSB Errors and the respective IOP Diagnostic Test Hooks designed to forced that error.

Table 9–1: Laser Bus Error Summary

LSB Error	Diagnostic Hook
LBER.nses[4]	Any IOP Specific Error
LBER.ctce[1]	**no means to force**
LBER.dtce[1]	**no means to force**
LBER.die[2]	Not implemented on IOP
LBER.she[2]	Not implemented on IOP
LBER.cae	IPCDR.frc_cae
LBER.nxae	Program controlled
LBER.cnfe	IPCDR.frc_cnfe
LBER.ste	**no means to force**
LBER.tde	IPCDR.diag_ecc_en, IPCDR.diag_ecc<6:0>, IPCDR.frc_dat_pe, IPCDR.frc_cmd_pe
LBER.cdpe2	IPCDR.frc_dat_pe
LBER.cdpe	IPCDR.frc_dat_pe
LBER.cpe2	IPCDR.frc_cmd_pe
LBER.cpe	IPCDR.frc_cmd_pe
LBER.ce2	IPCDR.diag_ecc_en, IPCDR.diag_ecc<6:0>
LBER.ce	IPCDR.diag_ecc_en, IPCDR.diag_ecc<6:0>
LBER.uce2	IPCDR.diag_ecc_en, IPCDR.diag_ecc<6:0>
LBER.uce	IPCDR.diag_ecc_en, IPCDR.diag_ecc<6:0>
LBER.e[3]	any of the above

[1]The estimated increase in logic and additional GA pins requirements does not warrant module test coverage gained.

[2]The Current implementation of the IOP does not drive this LSB signal.

[3]The IOP sets this bit whenever it detects LSB<ERR> asserted.

[4]This bit will be set when any IPCHST or IPCNSE error bit is set.

9.2.1 LBER.cae, Command/Address Error

The IPCDR.frc_cae diagnostic bit when set, will force the IOP to assert the LSB CA signal during a cycle which is not legal. This will result in all nodes monitoring the LSB, including the IOP, to record the error by setting LBER.cae.

IPCDR.frc_cae is a Write Only Bit. Setting it will induce a single Laser Bus LBER.cae error to occur.

9.2.2 LBER.nxae, Non-Existent Address Error

This LSB error condition can be forced by Software by writing a Non Existent LSB Memory address into a Mail Box Pointer Register (LMBPR).

9.2.3 LBER.cnfe, Confirmation Error

The IPCDR.frc_cnfe diagnostic bit when set, will force the IOP to assert the LSB CNF signal during a cycle which is not legal. This will result in all nodes monitoring the LSB, including the IOP, to record the error by setting LBER.cnfe.

IPCDR.frc_cnfe is a Write Only Bit. Setting it will induce a single Laser Bus LBER.cnfe error to occur.

9.2.4 LBER.tde, Transmit During Error

The IOP sets this error bit if it was driving a Command or Data on the LSB data lines during a cycle in which an error was detected. If the error occurred during a *memory data* cycle then LBER.ce or LBER.uce is also set. If the error occurred during the transmission of a *command* cycle, then the LBER.cpe will also be set. If the IOP was returning CSR read data when the error occurred, then the LBER.cdpe will also be set.

This LBER.tde error bit will therefore be set while the IOP is programmed to force:

- LSB ECC Errors via IPCDR.diag_ecc_en and IPCDR.ecc<6:0>.
- Parity errors on Command/Address cycles via IPCDR.frc_cmd_pe.
- Parity errors on CSR data cycles via IPCDR.frc_dat_pe.

9.2.5 LBER.cdpe, LBER.cdpe2, Parity Errors

Each module on the Laser Bus is required to check all LSB CSR data cycles for parity errors. If a parity error is detected during a CSR data cycle, the module(s) detecting the error is required to assert LSB<ERR>.

LBER.cdpe and LBER.cdpe2 can be forced by setting IPCDR.frc_dat_pe and doing an IOP CSR read(s).

9.2.6 LBER.cpe, LBER.cpe2, Parity Errors

Each module on the Laser Bus is required to check LSB Command/Address Cycles for parity errors. If a parity error is detected during a Command/Address cycle, the module(s) detecting the error is required to assert LSB<ERR>.

LBER.cpe and LBER.cpe2 can be forced by setting IPCDR.frc_cmd_pe and initiating a Mailbox transaction(s).

9.2.7 LSB ECC Error Description

The LSB's 128-bit data path, is divided into four longwords. Each longword has a corresponding 7-bit ECC check word. The following demonstrates ECC usage per each longword of data:

D<31:0>	covered by ECC<6:0>
D<63:32>	covered by ECC<13:7>
D<95:64>	covered by ECC<20:14>
D<127:96>	covered by ECC<27:21>

During ECC related errors, the IOP additionally captures the failing ECC Syndrome, in LBESR0-3.syndrome<6:0> registers.

The IOP provides a Diagnostic feature, (IPCDR.diag_ecc<6:0>), which when enabled (IPCDR.diag_ecc_en), will replace internally generated ECC codes for LSB memory write data cycles, with forced ECC Diagnostic Codes.

D<31:0>	covered by IPCDR.ecc<6:0>
D<63:32>	covered by IPCDR.ecc<6:0>
D<95:64>	covered by IPCDR.ecc<6:0>
D<127:96>	covered by IPCDR.ecc<6:0>

Diagnostic test routines can utilize this feature to force ECC errors in either of two methods:

1. Good ECC<6:0> codes, force bad data D<127:0>.

   This method utilizes *good* ECC codes across the entire data field of D<127:0>. By altering good data patterns by either one or two bits, a resulting single or double bit error will be forced.

2. Force Bad ECC<6:0> codes with good data.

   This method does not alter the data patterns by one or two bits, but rather selects a diagnostic ECC code which results in a single or double bit error being detected.

Having the IOP *transmit* bad ECC Codes as apposed to just altering *received* LSB ECC Codes, is the only means by which IPC I/O Pins can be pin-wiggled. This however, will result in **ALL** LSB Nodes monitoring the LSB to also detect ECC errors at their LSB Interface. Appropriate cleanup code to clear the Laser System Bus of these ECC errors must be performed after using this IOP diagnostic feature.

9.2.7.1 LCNR.ceen, Correctable Error Detection Enable

During normal operation, the IOP will check all LSB memory data cycles for ECC Errors. The LCNR.ceen register bit is used to control whether or not the IOP reports the error via the assertion of the LSB<ERR> signal. Correction of single bit errors *always* occurs, only the logging and reporting of the error is controlled by this bit. The IOP will always assert the LSB<ERR> signal if it detects uncorrectable LSB data ECC errors.

9.2.7.2 LBER.ce, Correctable ECC Error

This LBER error bit can be tested by forcing a single bit ECC Error code via IPCDR.diag_ecc<6:0> and issuing a Mailbox transaction.

9.2.7.3 LBER.ce2, Second Correctable Data Error

This LBER error bit can be tested by forcing a single bit ECC Error code via IPCDR.diag_ecc<6:0> and issuing a Mailbox transaction while LBER.ce is already set.

9.2.7.4 LBER.uce, Uncorrectable Data Error

This LBER error bit can be tested while forcing a double bit ECC Error code via IPCDR.diag_ecc<6:0> and issuing a Mailbox transaction.

9.2.7.5 LBER.uce2, Second Uncorrectable Data Error

This LBER error bit can be tested while forcing a double bit ECC Error code via IPCDR.diag_ecc<6:0> and issuing a Mailbox transaction, while LBER.uce is already set.

9.2.7.6 LBER.e, LSB<ERR> line asserted

This LBER register bit is set by the IOP *anytime* the LSB<ERR> signal is asserted.

9.2.7.7 LBECR0-1 Laser Bus Error Command Register

When the IOP detects a CE, UCE, CPE (Command Parity Error) or CDPE (CSR Data Parity Error), it latches the associated LSB Command and Address information in the LBECR0-1 registers. These registers remain *locked* until the *locking* error bit is cleared, i.e. CE, UCE, etc.

9.3 Vortex Bus Diagnostic Features

The Vortex Bus is the physical interconnect between the IPC and HIC. It is used to flow control Command/Data/Status packets between the LSB and I/O Bus Devices.

Vortex Bus transactions are checked for the following types of errors:

- Parity
- Illegal Command
- Sequence Error
- Buffer Overflow

If a Vortex error is detected, either the UP_VRTX_ERR bit or the DN_VRTX_ERR bit is set in the IPCNSE register. If bit<31>, INTR_NSES, is set in the IPCNSE register, an IPL17 interrupt will be posted to an LSB CPU(s).

9.3.1 Down Vortex Bus errors

Down Vortex errors are detected by the HIC and are collectively logged within IPCNSE.dn_vrtx_err. The Down HIC will signal the IPC of any Down Vortex errors via the HIC-IPC Error Interface signal, HIC_DN_ERROR. If enabled, the IPC can notify the Laser System Bus of the error by generating an IPL17 interrupt.

NOTE

It is important to note that the *Vortex packet counter* protocol is not maintained when Down Vortex errors occur. This means a potential *hang* condition exists if 3 Down Vortex Errors are forced on packets targetting a single HOSE, and then a fourth packet is in the IPC buffer targetting that same hose. To avoid this hang condition the diagnostic must issue a IOP Node Reset to reset the IPC packet counters.

The following sections will describe each Down Vortex error which the Down HIC can detect.

Table 9–2 summarizes Down Vortex detectable errors and the respective Diagnostic Test Hook utilized to force the error.

9.3.1.1 Down Vortex *Data Parity* Error Description

Down Vortex Parity Error detection is done across the Vortex's quadword data path as follows:

- IPC_PAR_DN<0> protects IPC_D_DN<31:0>
- IPC_PAR_DN<1> protects IPC_D_DN<63:32>

Parity errors can be forced across the Down Vortex by setting the following IPC diagnostic register bit(s) and issuing a Mailbox transaction. The HIC will collectively record the occurrence of Down Vortex Parity Errors via IPCNSE.dn_vrtx_err.

- IPCDR.frc_dn_dpe<0> forces IPC_D_DN<31:0>

- IPCDR.frc_dn_dpe<1> forces IPC_D_DN<63:32>

9.3.1.2 Down Vortex *Sequence* Error Description

A Down Vortex Sequence error is the result of the IPC transmitting a packet to the HIC which is of incorrect size as specified by the transmitted packet header (i.e. command/length field).

A sequence error can be forced across the Down Vortex by setting the IPCDR.frc_dn_seq_err diagnostic register bit and issuing a Mailbox transaction. The HIC will collectively record the occurrence of Down Vortex Sequence Errors via IPCNSE.dn_vrtx_err.

9.3.1.3 Down Vortex *Illegal Command* Error Description

This error indicates the IPC sent an Illegal Vortex command (with good parity) to the DOWN HIC. An Illegal Command Error can be forced across the Down Vortex by setting the IPCDR.frc_dn_ill_cmd diagnostic register bit and issuing a Mailbox transaction. The HIC will collectively record the occurrence of Down Vortex Illegal Command Errors via IPCNSE.dn_vrtx_err.

9.3.1.4 Down Vortex *Buffer Overflow* Error Description

This error indicates the IPC attempted to load a packet into the DOWN HIC when its packet buffer was already full. The HIC will collectively record the occurrence of a Down Vortex Buffer Overflow Error via IPCNSE.dn_vrtx_err.

Forcing of this IOP specific error will not enhance Gate Array pin or interconnect error coverage and will therefore not be forced.

9.3.1.5 Down HIC *Internal* Error Description

This internal HIC error condition indicates a hardware problem, usually indicating that control logic encountered an illogical condition.

The same error line used by the HIC to signal the IPC that a Down Vortex error has occurred is also used to signal that a HIC Internal Error has been detected. There is no specific error bit to indcate that a Down HIC Internal Error has been detected, rather this error condition is recorded via IPCNSE.dn_vrtx_err.

Forcing of the HIC Internal Error will not enhance Gate Array pin or interconnect error coverage and will therefore not be forced.

9.3.2 Up Vortex Bus Errors

Up Vortex errors are detected by the IPC and recorded within IPCNSE.up_vrtx_err. This error bit collectively records the following UP Vortex Errors.

- Parity
- Illegal Command
- Sequence Error
- Buffer Overflow

If bit<31>, INTR_NSES, is set in the IPCNSE register, an IPL17 interrupt will be posted to an LSB CPU(s).

The following sections will describe each Up Vortex error which the IPC can detect. Table 9–2 summarizes all Up Vortex detectable errors and the respective Diagnostic Test Hooks designed to force that error.

9.3.2.1 Up Vortex *Parity* Error Description

Up Vortex Parity Error detection is done across the Vortex's Quadword Data path as follows:

- HIC_UP_PAR<0> protects HIC_UP_DATA<31:0> and HIC_UP_CMD_A<3:0> going to IPCA.
- HIC_UP_PAR<1> protects HIC_UP_DATA<63:32>and HIC_UP_CMD_B<3:0> going to IPCB.

An Up Vortex Parity Error can be forced by programming a Mailbox Loopback transfer with *Enable Forcing Loopback Errors* set, and *Force IPCA Parity Error* (FAPE) and/or *Force IPCB Parity Error* (FBPE) set. Refer to Table 9–2.

9.3.2.2 Up Vortex *Sequence* Error Description

An Up Vortex Sequence Error is the result of the HIC transmitting a packet to the IPC which is of incorrect size as specified by the transmitted packet header (i.e. command/length field).

An Up Vortex Sequence Error can be forced by programming a Mailbox Loopback transfer with *Enable Forcing Loopback Errors* set, and *Force IPCA Sequence Error* (FASE) and/or *Force IPCB Sequence Error* (FBSE) set. Refer to Table 9–2.

9.3.2.3 Up HIC *Buffer Overflow* Error Description

IPCNSE.up_vrtx_err can also indicate the HIC attempted to load a packet into the IPC when the IPC packet buffer was already full.

Forcing of this IOP specific error will not enhance Gate Array pin or interconnect error coverage and will therefore not be forced.

9.3.2.4 IPC *Internal* Errors

IPCNSE.ipc_ie indicates a hardware problem where control logic encountered an illogical condition. Forcing of this IPC specific error will not enhance Gate Array pin or interconnect error coverage and will therefore not be forced.

9.3.2.5 UP HIC *Internal* Errors

IPCNSE.up_hic_ie is the result of the UP HIC detecting an internal hardware problem, usually when control logic encountered an illogical condition. Forcing of this internal HIC error will not enhance Gate Array pin or interconnect error coverage and will therefore not be forced.

Table 9–2: Vortex Bus Error Summary

DOWN Vortex Summary

Vortex Error	Diagnostic Hook
IPCNSE.dn_vrtx_err[1]	IPCDR.frc_dn_ill_cmd, IPCDR.frc_dn_seq_err, IPCDR.frc_dn_dpe<1:0>

UP Vortex Summary

Vortex Error	Diagnostic Hook
IPCNSE.up_hic_ie	**no means to force**
IPCNSE.ipc_ie	**no means to force**
IPCNSE.up_hic_oflo	Refer to Section 9.4.2.1
IPCNSE.up_vrtx_err	FAPE, Force IPC-A Parity Error, FBPE, Force IPC-B Parity Error, FASE, Force IPC-A Sequence Error, FBSE, Force IPC-B Sequence Error

[1]The DOWN HIC Chip, utilizes a single error signal to report Down Vortex and HIC Internal errors to the IPC. For this reason, a single composite IPCNSE error bit is used.

9.4 Hose Bus Diagnostic Features

The Hose Bus is the primary interconnect between the IOP and the I/O Adapter.

Down Hose errors are detected locally within each I/O Adapter. The I/O Adapter signals the IPC of a non-recoverable *FATAL* Down Hose error by driving the Hose<ERROR> signal. The assertion of this Hose<ERROR> signal causes the IPC to generate an interrupt to the CPU(s) if enabled.

Up Hose errors are detected by the UP HIC and are recorded within IPCNSE, located within the IPC. The UP HIC informs the IPC of these UP Hose errors via the HIC-IPC Error Interface signals HIC_ERROR_UP<3:0> where:

- HIC_UP_ERR_HOSE_ID<1:0> - Specify the failing Hose ID.
- HIC_UP_ERR_CODE<1:0> - Specify the error condition.
  - 00 - Illegal Packet Error
  - 01 - Parity Error
  - 10 - Overflow Error
  - 11 - Internal Error
- HIC_UP_ERR_VALID - Indicates that the HIC_UP_ERR_HOSE_ID<1:0> and HIC_UP_ERR_CODE<1:0> lines are contain valid information and should be logged in the IPCNSE.

Table 9–3 summarizes Hose Errors and IOP CSR Error bits. Refer then to Section 9.6 for forcing of these errors.

9.4.1 Up Hose Errors

When *not* programmed in Loopback Mode, the IOP will perform normal Up Hose packet transmissions. Forcing of Up Hose errors while in this mode, is a function of the attached I/O Adapter diagnostic features.

When the IOP is programmed in Loopback Mode, the Down HIC can be programmed to force *most* Hose Errors by looping Down Hose traffic to UP Hose traffic. The UP HIC will then detect any forced Hose errors and report them to the IPC via the HIC-IPC error interface.

9.4.1.1 Hose Connection Errors

IPCHST.Hx_stat<3> indicates a transition occurred on the Hose *PWROK* signal.

IPCHST.Hx_stat<2:1> indicates the current level of the Hose Port signals: *CBLOK* and *PWROK*. These register bits must be checked for the proper state prior to executing any Mailbox Loopback transactions to the corresponding Hose.

When IPCHST.Hx_stat<0> is set, it indiacates that the Hose signal *ERROR* asserted. If IPCNSE<31>, INTR_NSES, is set and IPCHST.Hx_stat<0> also sets, then the IPC will post an IPL17 interrupt to an LSB CPU(s). This bit being set indicates that the attached

I/O Adapter has encountered a *FATAL* error condition, usually requiring a RESET of the adapter.

These signals represent the state of the I/O Subsystem and are therefore not diagnostically programmable.

9.4.1.2 Up Hose Parity Errors

The IPCNSE.up_hose_par_err<3:0> record UP Hose Parity Errors. These errors can be forced by programming the IOP in Loopback Mode and instructing the Down HIC to generate Down Hose parity errors. Refer to Section 9.6 for details.

9.4.1.3 Up Hose Illegal Command and Sequence Errors

The IPCNSE.up_hose_pkt_err<3:0> field are composite error bits. With one bit for each Hose and each representing either Up Hose Illegal Command or Sequence errors.

These errors can be forced by programming the IOP in Loopback Mode and instructing the Down HIC to generate either Down Hose Illegal Command or Sequence errors. Refer to Section 9.6 for details.

9.4.1.4 Up Hose FIFO Overflow Errors

IPCNSE.up_hose_oflo_err<3:0> field, indicates the attached I/O Adapter attempted to load a packet into the UP HIC when the UP HIC's packet buffer was already full.

Refer to Section 9.4.2.1 for details on forcing this error.

9.4.2 UP Hose Queue Management

The UP HIC and IPC gate arrays, maintain a queue of data buffers. The IPC has a one queue by two deep buffer across all four UP Hose Ports. The UP HIC has a one queue by three deep for each UP Hose Port. Collectively then, these *UP queues* allow the Laser I/O Subsystem to pipeline packet transmissions between the I/O Devices and the Laser Bus.

The IPC provides a diagnostic feature, IPCDR.dis_lsb_cmd, which inhibits the IPC arbitration for the LSB. The sole exception is when an LMBPR is written. The IOP will arb for the bus to fetch Mailbox command and send it down the hose. This feature, in combination with Loopback Mode and Mailbox transactions, can be used to test the UP HIC and IPC queue management and data buffer integrity.

9.4.2.1 Mailbox Loopback Programming

The following provides the programming and anticipated results using the IPCDR.dis_lsb_cmd feature:

1. Enable Loopback mode, IPCDR.hic_lpbck_en
2. Disable LSB commands, IPCDR.dis_lsb_cmd
3. Configure in LSB Memory, *8* Diagnostic Mailbox Structures all targeting the same Hose Port. Refer to Figure 9–2.
4. Write LMBPR0 with addresses of Mailbox Structures 1 thru 8.

   Poll IPCNSE<3:0> to see that the Mailbox Command Packet has been tranmitted across the Down Vortex. Once the packet has been transmitted, the corresponding IPCNSE.mbx_tip<n> must be cleared. Clearing this bit will allow the IPC to increment to the next entry in the Mailbox Transaction queue, and process the next command.

   Repeating the *polling/clearing* sequence 4 times will cause the Down HIC to send the first 5 Mailbox Command packets down the Hose. These are looped back to the UP HIC as Mailbox Return Status packets. Five packets are sufficient to fill all available data buffer queues in the IPC and UP HIC.

   Anticipate no Return Mailbox Status and no IPCNSE Errors.
5. *Poll/Clear* for packets 6-8.

   An UP_HOSE_OFLO error will be detected by the UP HIC when the DOWN HIC transmits packets #6-8.

   If IPCNSE<31>, INTR_ON_NSES, is set, an IPL17 error interrupt pending flag sets in the IPC. Note the interrupt will not be posted until the IPCDR bit, IPCDR.dis_lsb_cmd, is cleared. IPCNSE<7:4> should be checked to verify the correct error bit was set.

   Transmission of the eighth Mailbox Packet has the effect of aligning the IPC's queue pointer to the first queued LMBPR entry. This is needed so that the status from the five error-free packets, in the IOP queues, is returned to the correct Mailbox Data Structure in LSB Memory, after IPCDR.dis_lsb_cmd is cleared.
6. Enable LSB commands by clearing IPCDR.dis_lsb_cmd.

   This will enable the IPC to begin requesting the LSB and resume the communications between the UP HIC and IPC.

   Anticipate Mailbox Loopback data in the first 5 Mailbox Data Structures after 300 LSB Cycles. The remaining 3 Mailbox Data Structures should remain unchanged.

NOTE

If interrupts are enabled, an interrupt will be posted before the status packets are processed.

9.4.3 Down Hose Errors

When *not* programmed in Loopback Mode, the IOP will perform normal Down Hose packet transmissions. Forcing of Down Hose errors while in this mode, is a function of the attached I/O Adapter diagnostic features.

When the IOP is programmed in Loopback Mode, the IPC will drive the Hose signal ERROR, thus disabling all attached I/O Adapters from receiving Down Hose packets. All forced Down Hose errors are then directed to the Up HIC via the HIC's Loopback feature. Refer to Section 9.6 for Loopback Mode programming.

9.4.4 Down Hose Queue Management

The DN HIC and IPC gate arrays maintain a queue of Down Hose data buffers. The IPC has a one queue by two deep across all four DOWN Hose Ports. While the DN HIC has a one queue by three deep for each Down Hose Port. Collectively these *DOWN queues* allow the IOP to pipeline packet transmissions between the Laser Bus and I/O Devices.

Due to the nature of Down Hose Flow Control protocol, it is *currently not* possible to diagnostically program the Down HIC or IPC such that Down Hose queue control is thoroughly tested. However, due to the nature of these queues, data integrity is implicitly tested.

Table 9–3: Hose Bus Error Summary

IOP **Up** Hose Error Detection Summary

Hose Error	Diagnostic Hook
IPCNSE.up_hose_par_err<3:0>	Refer Section 9.6
IPCNSE.up_hose_pkt_err<3:0>	Refer Section 9.6
IPCNSE.up_hose_oflo<3:0>	Refer Section 9.4.2.1

IOP **Down** Hose Forced Error Summary

Hose Error	Diagnostic Hook
Parity Error	Refer Section 9.6
Illegal Command	Refer Section 9.6
Sequence Errors	Refer Section 9.6
Down Fifo Overflow	Not programmable

9.5 IOP Miscellaneous features

This section will describe remaining errors and features of the IOP which are not specifically related to any one bus.

9.5.1 IOP LED Usage

The IOP will employ a single *yellow* LED. This LED will be connected to the LCNR.stf and will visually display the IOP's Selftest Completion Status. When LCNR.stf is set the LED is lit. When LCNR.stf is clear, the LED is not lit. On powerup or node reset, the LCNR.stf bit is clear, therefore the LED is not lit.

After successful IOP Specific and I/O Adapter testing, the Laser I/O Subsystem RBD will set LCNR.stf, therefore lighting the IOP's LED. If an error is detected during I/O Adapter testing, and cannot be isolated to the I/O Adapter itself, then the LED on the IOP will not be lit. If the error can be isolated to the I/O Adapter, then the IOP LED will be lit.

In the event of *un*successful IOP Specific Testing, the Laser I/O Subsystem RBD will *not* set LCNR.stf therefore keeping the IOP's LED extinguished. This LED will remain extinguished during LAMB and FLAG testing regardless if these I/O Adapter tests pass or fail.

Both the LAMB and FLAG modules will employ their own LEDs to visually display their Selftest results.

9.5.2 On Board EEPROM

The IOP has 2 Kbytes of EEPROM storage. The intended use of the EEPROM storage is for on-line error logging and module revision and repair history.

9.6 IOP Hose Loopback

The IOP supports a Hose Loopback feature Figure 9–1, which will loopback Diagnostic Mailbox Structures built in Main Memory Figure 9–2. Once enabled, any Hose Port can be loopback tested as long as there is an attached I/O Adapter to provide proper termination.

NOTE

To avoid Hose cable electrical reflection problems, IOP internal loopback testing should be restricted to Hose Ports having properly configured Remote Bus Adapters. This can be determined through Hose status signals of *PWROK=1, CBLOK=1* and *PWRCYCLED=1*

When the IOP is placed in loopback mode the Hose<ERROR> signal is automatically driven by the IPC. This signals the I/O Adapters to ignore all Down Hose traffic and not to generate any Up Hose traffic. This prevents the attached I/O Adapters from processing IOP specific diagnostic packets.

9.6.1 Loopback Functional Operation

The IOP's Hose Loopback feature functions exactly like any normal Mailbox operation with the exception of the Mailbox Return Status. During Loopback Mode, the returned Mailbox Status is replaced with LW0, LW1, LW2 and LW3. All other returned status Longwords are don't cares, refer to Figure 9–2
.

Normal Loopback Operation

To program a Normal Loopback operation, with *NO* forced Hose errors, the programmer is instructed to follow the following procedure:

1. Enable Loopback Mode by setting IPCDR.hic_lpbck_en.

   This will force the IPC to drive the Hose<ERROR> signal. All I/O Adapters are then disabled from processing any Hose packets. *It is recommended this be done first to allow for the I/O Adapter(s) to settle out before loopback operations are initiated.*

2. Build a Diagnostic Mailbox Structure in Main Memory.

   Defining the Hose Port to be loopbacked LW1<17:16>, clearing LW0<31>, define LW0<30:7>, LW1<15:0>, LW2<31:0> and LW3<31:0> with loopback Write data, and clearing the Loopback Read fields, LW8, LW9, LW10 and LW11.

3. Initiate the Diagnostic Mailbox Loopback operation by writing the Physical Memory Address of this Mailbox Structure to LMBPR0, *A00 0C00*.

4. Loopback Read Data at Mailbox locations LW8, LW9, LW10, LW11, should then be *polled* to determine when it becomes available.

   It is recommended that a software timer be programmed to prevent infinite loops. In the event that a Mailbox transaction does not complete diagnostics can continue to execute subsequent Mailbox transactions by clearing the MBX_TIP<3:0> bit.

Forced Hose Error Loopback Operation

To program a loopback operation with forced Up Hose errors, the programmer is instructed to follow the following procedure:

1. Enable Loopback Mode by setting IPCDR.hic_lpbck_en as above.

2. Build a Diagnostic Mailbox Structure in Main Memory.

   Defining the Hose Port to be loopbacked LW1<17:16>, set LW0<31>, define LW0<6:0> with the Hose/Vortex error to be forced. Due to the forced Hose Errors, no loopback data will be done. Therefore LW2 and LW3 are don't care fields. However these don't care fields can be written with a non-zero value to know if the data did get written back unexpectedly.

3. Initiate the Diagnostic Mailbox Loopback operation by writing the Physical Memory Address of this Mailbox Structure to LMBPR0, *A00 0C00*.

4. Poll the IPCNSE register for the anticipated Up Hose Port and Hose to be error forced.

To execute subsequent MailBox transactions, the MBX_TIP<3:0> bit must be cleared.

Figure 9–1: IOP Diagnostic Loopback Diagram

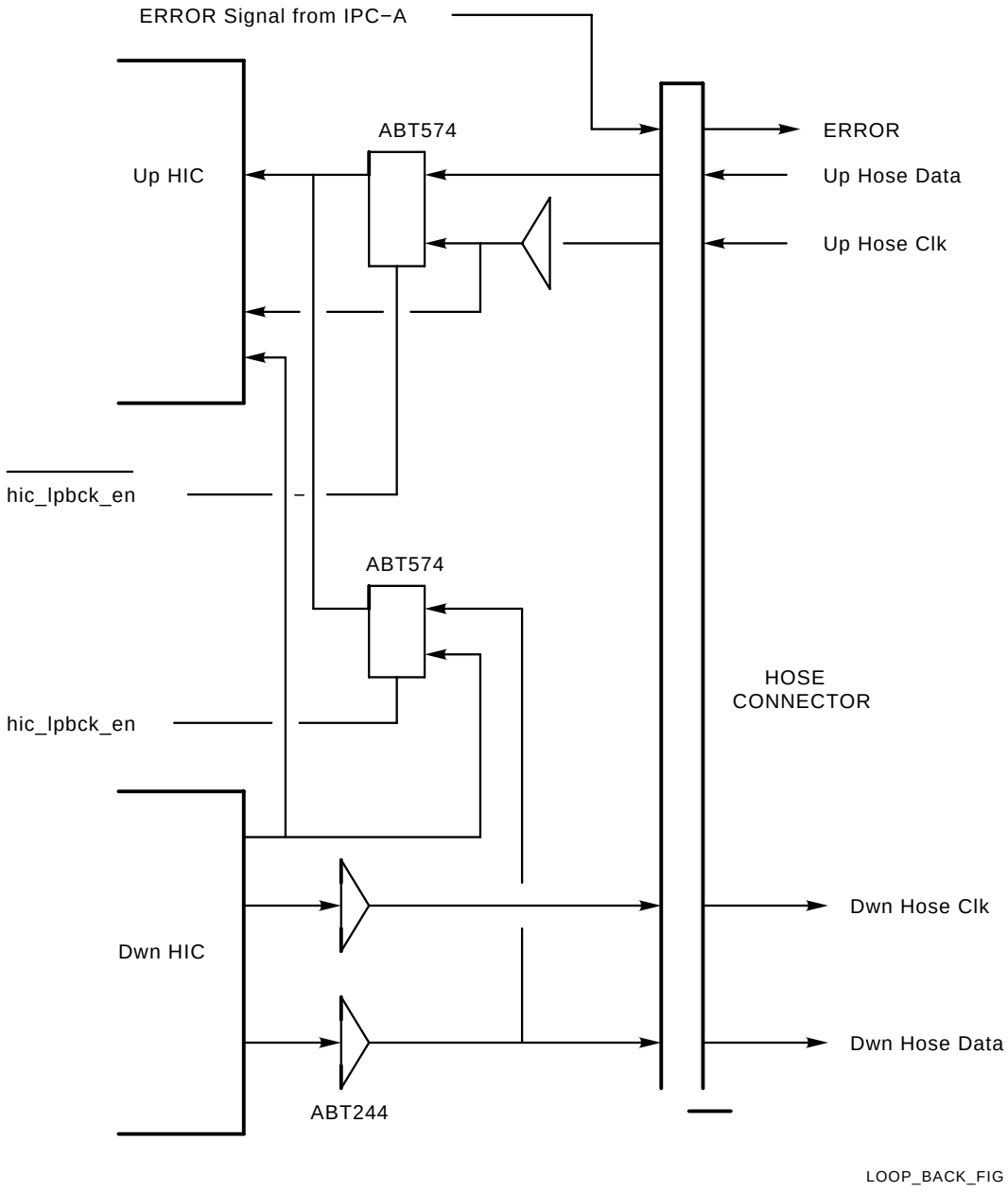
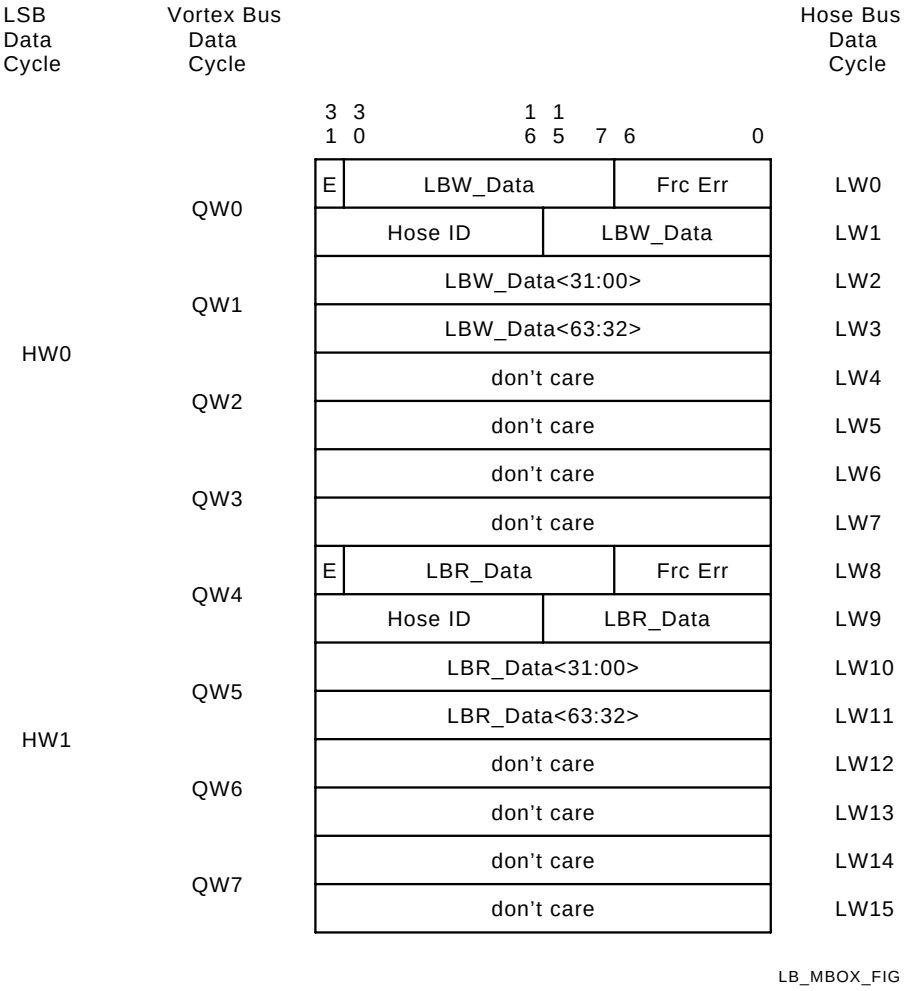


Figure 9–2: Loopback Mailbox Structure Diagram



LB_MBOX_FIG

Table 9–4: Mailbox Loopback Structure

Name	Long word	Bit(s)	Description
E	0	31	Enable Forcing Loopback Errors When Loopback Mode is enabled, this bit instructs the HIC's to either loopback normal Mailbox Data (when cleared; 0) or to force loopback errors (when set; 1). In addition to this bit, LW0<9:0> instructs the DN HIC to force Down HOSE errors or for the UP HIC to force UP Vortex Errors.
LBW_Data	0	30:7	Loopback Write Data The DN HIC will loopback this LW0<30:10> data to the UP HIC when Loopback is enabled. When the transfer is complete, this data field will be readable from the Mailbox Structure LW8<30:10>.
Frc Err	0	6:0	Forced Errors This bit field represents all possibe Down Hose and Up Vortex forced error conditions. A bit set will force the associated error. Any combination of concurrent forced errors are possible. Forcing **UP Vortex** Errors — Bit<6> FBSE, IPC-B Sequence Error — Bit<5> FASE, IPC-A Sequence Error — Bit<4> FBPE, IPC-B Parity Error — Bit<3> FAPE, IPC-A Parity Error Forcing **DOWN HOSE** Errors — Bit<2> FDHSE, Down Hose Seq Error — Bit<1> FDHICE, Down Hose Ill Cmd Error — Bit<0> FDHPE, Down Parity Error
LBW_Data	1	15:0	Loopback Write Data The DN HIC will loopback this LW1<15:0> data to the UP HIC when Loopback is enabled. When the transfer is complete, this LW1<15:0> data will be readable from the Mailbox Structure LW9<15:0>.
Hose ID	1	31:16	Hose ID This field instructs the DN HIC which Hose Port to direct this Loopback transaction to. This field will also be loopbacked to LW9<31:16>.

Table 9–4 (Cont.): Mailbox Loopback Structure

Name	Long word	Bit(s)	Description
LBW_DATA<31:0>	2	31:0	Loop Back WRITE DATA<31:0> The DN HIC will loopback this Longword of data to the UP HIC when Loopback is enabled. When the transfer is complete, this Longword of data will be readable from the Mailbox Structure LW10.
LBW_DATA<63:32>	3	31:0	Loop Back WRITE DATA<63:32> The DN HIC will loopback this Longword of data to the UP HIC when Loopback is enabled. When the transfer is complete, this Longword of data will be readable from the Mailbox Structure LW11.
don't care	7:4	31:0	Don't Care These Longwords are ignored while the IOP Module is programmed in Loopback mode. However, these fields should be written with *non-zero* data patterns such to further isolate loopback control errors
LW8	8	31:0	Loopback Read Data<LW0> The DN HIC will loopback Mailbox LW0<31:0> to this Mailbox location.
LW9	9	31:0	Loopback Read Data<LW1> The DN HIC will loopback Mailbox LW1<31:0> to this Mailbox location.
LBR_Data<31:00>	10	31:0	Loopback Read Data<31:0> The DN HIC will loopback Mailbox LW2<31:0> to this Mailbox location.
LBR_Data<63:32>	11	31:0	Loopback Read Data<63:32> The DN HIC will loopback Mailbox LW3<31:0> to this Mailbox location.
don't care	15:12	31:0	Don't Care These Longwords are ignored while the IOP Module is programmed in Loopback mode. However, these fields should be written with *zero's* data patterns such to further isolate loopback control errors.

9.7 IOP IPL17 Error Interrupts

The IOP is capable of generating Laser System Bus interrupts on detection of IOP *specific* errors. When IOP IPL17 Error interrupts are enabled thru IPCNSE.intr_nses, the setting of any IPCNSE Node Specific Error bit will cause the IOP to interrupt the selected CPU(s) at IPL17. Refer to the register description of the LCPUMASK register for details on how CPU(s) are selected to service interrupts.

To post an interrupt, the IOP will issue a CSR Write command to the LIOINTR register. CPU(s) that are targetted are selected based on the contents of the LCPUMASK register.

When a CPU reads the LILID3 register and the IPC has an outstanding IPL17 error interrupt posted, the vector that gets returned is from the IPCVR. No new IOP specific interrupts will be issued until the LBER.nses bit is cleared, i.e. all IOP specific error bits must be cleared.

NOTE

IOP IPL17 error interrupts will take precedence over any previously queued HOSE<3:0> **IPL17 interrupts.**

9.7.1 Forcing IOP IPL17 Error Interrupts

While the IOP is programmed in loopback mode, IOP IPL17 Error interrupts can be generated by forcing any Vortex or Hose Bus error which causes an IPCNSE error bit to set.

9.8 Laser I/O Subsystem Programming notes

The following specify some of the do's and don'ts related to Laser I/O Subsystem during Diagnostic testing.

9.8.1 Hose Loopback settling time

Before any loopback transaction is initiated, care must be taken to allow sufficient time (1 LSB transaction) for any attached I/O Adapter to recognize that the IOP's loopback feature has been enabled.

Furthermore, it is also mandatory for all outstanding Hose transactions to be completed before enabling loopback mode.

CHAPTER 10

SYSTEM CLOCK GENERATION LOGIC

The Laser System Clock Generation logic resides on the IOP. This logic supplies no special function to the IOP other than to provide it with the Laser System Clocks, as it does for all other Laser Nodes.

10.1 Overview of the LSB Clock System

The LSB clocking scheme supports a bus cycle time of 17 ns, which translates to a system clock frequency of 58.823 MHz with a tolerance of the oscillator being +/- 75 PPM (+/- 4411.725 Hz). This includes the basic crystal tolerance of +/- 25 PPM plus oscillator operation over 0 to 70 degrees C.

Bus cycle times based on the above tolerance are:

— 16.999 ns (maximum)

— 17.000 ns (typical)

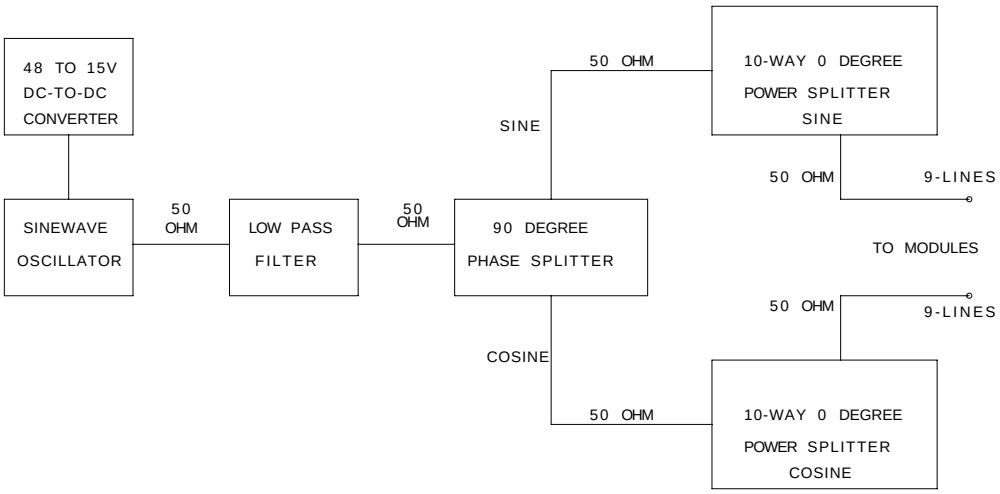
— 17.001 ns (minimum)

A detailed description of the LSB clock system is presented in the *LSB Electrical Spec*

Clock Generator

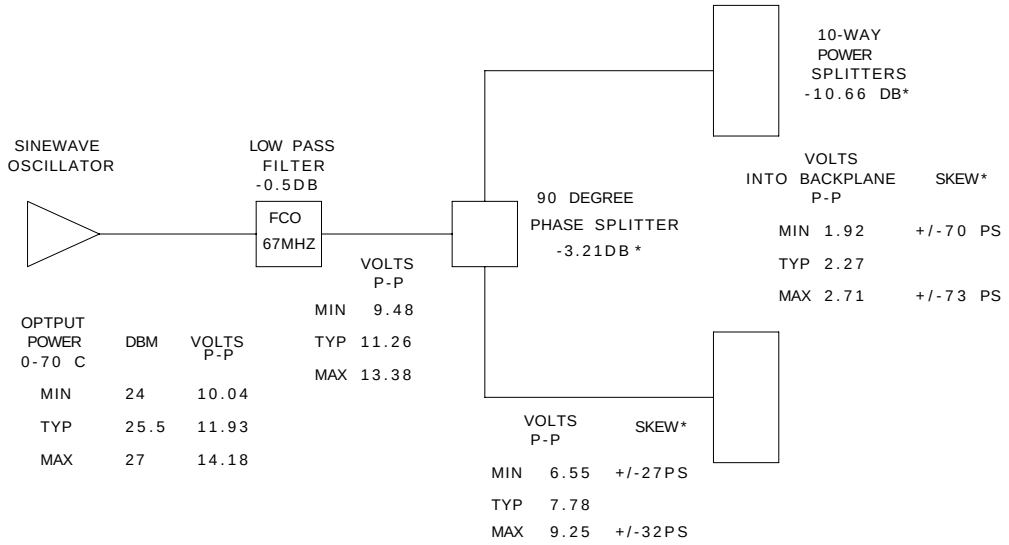
The LSB clocks are distributed via a five tier clock distribution system. The first tier is a radially distributed sine and cosine clock. All of the radially distributed clock signals will be generated by a sinusodial clock generator circuit located on the IOP module. A Vectron Inc. CO 484 series sinewave oscillator, DPN 18-34250-01 is used as the primary clock source. The output from the oscillator is sent to a low pass filter DPN 12-34896-01 to minimize harmonic content of the sinewave. This signal will be split into a sine and cosine clock signal using a 90 degree 2-way splitter, DPN 16-17994-02. The sine and cosine clock signals will be split into 9 copies of each using two 10 way power splitters, DPN 16-17993-01. A block diagram of the clock generator is shown in Figure 10–1. The clock generator components, including the 15V, 5 Watt DC-to-DC converter to power it, take a total board area of approximatly 6 sq. inches.

Figure 10–1: LSB Clock Generator Block Diagram



The expected voltage levels at each point in the Clock Generator Block Daigram are shown in Figure 10–2.

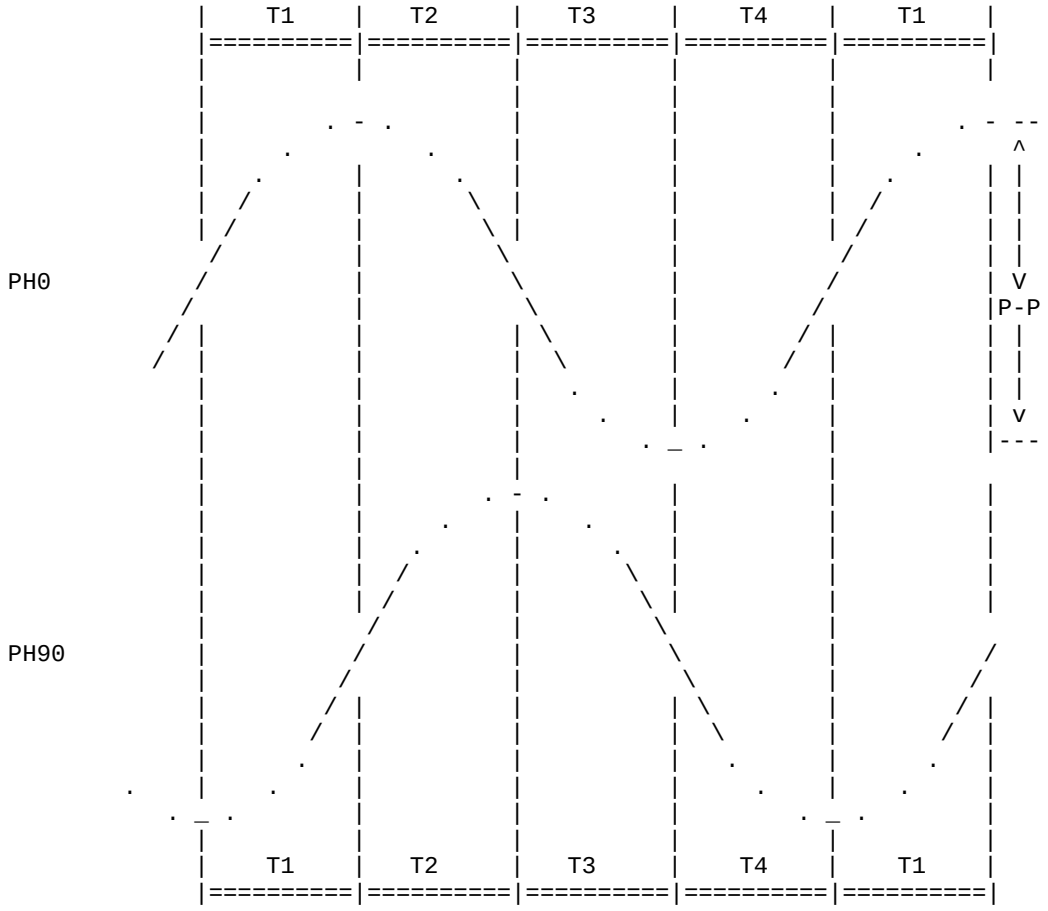
Figure 10–2: Clock Generator Voltage Levels



* MEASURED BY VENDOR OVER 0 TO 70 DEGREES C

The voltage waveforms at the output of the Clock Generator are shown in Figure 10–3.

Figure 10–3: LSB Clock Waveforms



The detailed Clock Generator drawing is shown in Figure 10–4.

Figure 10–4: Clock Generator Drawing

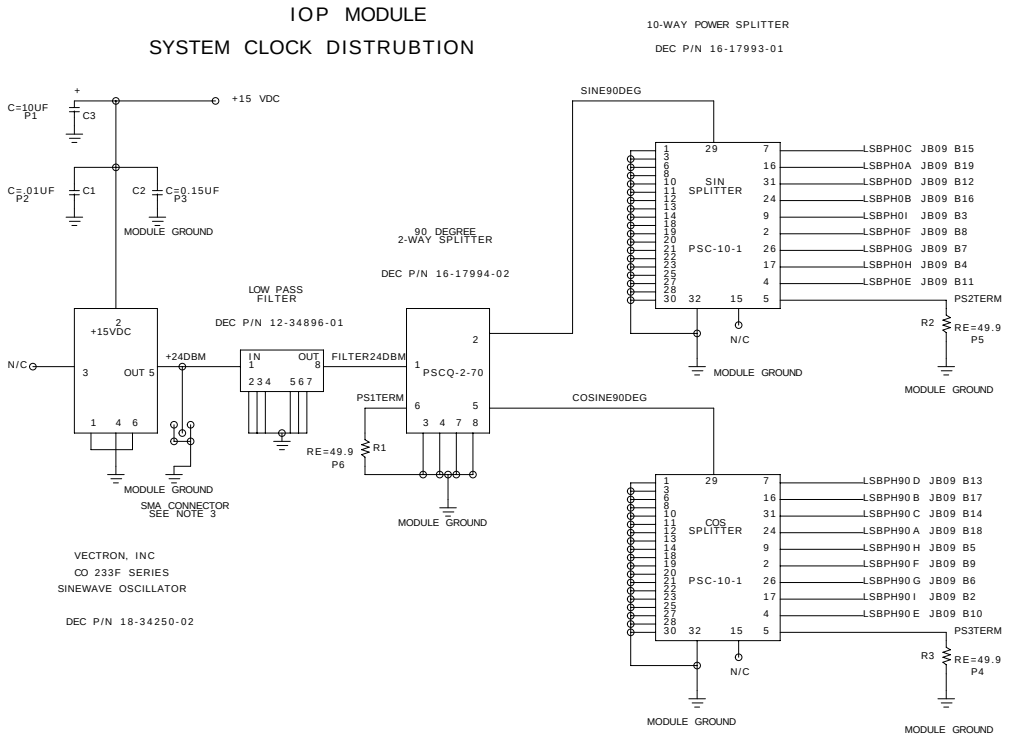
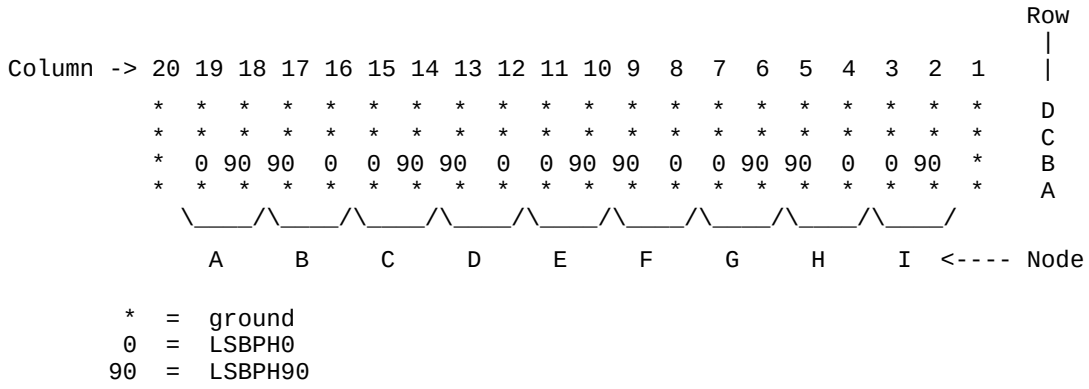


Table 10–1: Clock Generator Parts List

Part	Value	DPN
C1	0.01 uF 50V	10-24053-01
C2	0.22 uF 50V	10-24053-12
C3	10 uF 25V	10-24455-19
R1,R2,R3	49.9 Ohms	13-23827-68
Osc	58.823 MHz Sinewave	18-34250-02
Filter	Low Pass, Fco=67 MHz	12-34896-01
PSCQ-2-70	90 Degree Power Splitter	16-17994-02
PSC-10-1	10-Way 0 Degree Power Splitter	16-17993-01
(not shown)	15V DC-to-DC Converter	12-xxxxx-xx

The Laser IOP module will include an extra Teradyne connector segment which supplies the clocks to the backplane. The physical layout of the clock generator circuit will be under PCA rev control.

Figure 10–5: IOP Connector Segment For Distributing LSB Clocks



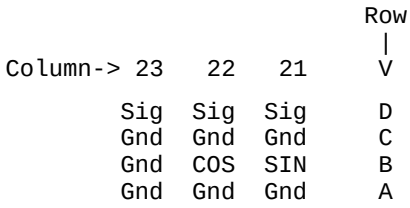
Each module will receive two wires, one for the sine (PH0) and one for cosine (PH90) clock signals. All of the clock signals in the backplane will satisfy the following peak to peak voltage requirements:

Figure 10–6: Clock Generator Output Voltages

Oscillator Power	Minimum	Typical	Maximum
13 dBm (2.82V)	1.92V		
17 dBm (4.49V)			2.71V

All etch runs from the clock generator circuitry to the modules must be balanced in length to control the inter-module clock skew. In addition, these two clock signals must be shrouded by returns as they pass through the backplane to module connector to prevent noise being coupled to the clocks. The LSB backplane has been routed with the sine clock connected to pin B21 and the cosine clock connected to pin B22. Connector pins A21, A22, A23, B23, C21, C22, and C23 surround the clock pins and are tied to ground. See Figure 10–7 below.

Figure 10–7: Connector Segment For Receiving LSB Clocks



All of the clock runs are on layer L8 of the backplane and use 7/20 etch technology to provide a nominal impedance of 50 ohms. From the Laser signal integrity standards document :

```
layer L8 reserved for clocks (7/20)

   minimum:        45.5 ohms @ 185.6ps/inch
   nominal:        50.5 ohms @ 175.7ps/inch
   maximum:        55.6 ohms @ 169.5ps/inch
```

CHAPTER 11

IOP PHYSICAL AND ELECTRICAL CHARACTERISTICS

11.1 IOP Physical Characteristics

The IOP module is an Extended-Hex+ (i.e. +1 inch) FR4 Epoxy base module with the following dimensions:

HEIGHT 15.7 inches

WIDTH 12.91 inches

THICKNESS 0.093 inches

The maximum DC power consumption will not exceed 130 watts.

The "Extended Hex+" Module provides 202 square inches of usable space. The current layup is 10 layers.

Figure 11–1 and Figure 11–2 contain IOP Module layout drawings of side 1 and side 2 respectively.

11.2 IOP Electrical Characteristics

The IOP module is provided a 48 volt DC supply from the central power system. An on module DC to DC Converter provides the 5 volt power for the module. The current 5 volt power requirement is estimated to be 135 watts.

A second on module DC to DC Converter provides the 15 volt power for the Laser System Clock Generation logic. The current 15 volt power requirement is estimated to be 1 watt.

APPENDIX A

APPLICATION HINTS

SMALLER THAN DOUBLE HEXAWORD TRANSACTIONS

The I/O Port Module (IOP) functionally supports smaller than Double Hexaword Transactions across the Laser System Bus (LSB). However, it should be noted that severe I/O performance degredation and excessive LSB bandwidth utilization will occur if such transactions are frequently executed. Consider the following case.

Typical XMI Write transactions occur in octaword sizes. A single fast XMI node may write up to four octawords to the XMI Adapter Module (LAMB) in a little over 1.5 microseconds. This assumes that a typical XMI Node will require at least 6 XMI cycles per transaction (I. E. 1 Arb cycle, 1 C/A cycle, 2 data cycles, and 2 CNF cycles).

64ns x 6 Cyc/Xaction x 4 Xactions/OW = 1536ns

If the XMI MORE protocol is not used the LAMB will receive these writes and pass them directly to the IOP as octaword write packets. No packing of data into larger size packets will occur.

In order for the IOP to effectively perform an octaword write across the LSB it must READ the Double Hexaword from Laser Memory which contains the desired octaward to be written, merge the octaword received from the LAMB into the correct position within the Laser Double Hexaword, then write the Double Hexaword back to Laser Memory. It must perform this sequence four times, once for each of the octawords received from the LAMB, in order to write an entire Double Hexaword's worth of data to the Laser Memory. This entire operation will take approximately 2.4 microseconds across the LSB.

17ns (15 Cyc/Xaction x 2 Xactions/OW + 7 Stalls/OW) 4 Xactions = 2516ns

Note that the 7 stall cycles are caused by the fact that the IOP is doing back-to-back WRITES followed by READS to the same memory bank (I.E. READ/WRITE, READ/WRITE, READ /WRITE, READ/WRITE, ETC.)

From the above analysis it can be clearly seen that the LSB does not have sufficient peak bandwidth to support even one XMI Node executing bursts of octaword write traffic.

If the XMI MORE protocol is used, however, the LAMB Module will pack 4 XMI octaword writes into one Double Hexaword write and pass it to the IOP as such. The IOP will then complete the entire write to LSB Memory as one Double Hexaword LSB Write transaction, more than 8 times as efficient as the above described scenerio.

MAILBOX TRANSACTIONS

The I/O Port Module uses Laser Mailbox Transactions to access CSR Registers which reside in remote bus nodes (I.E. XMI or Futurebus+). These transactions are VERY inefficient and should be avoided as much as possible. Consider the following scenerio:

I/O adapter modules such as the Futurebus+ to Laser (FLAG) module use memory mapping registers to map 32-bit Futurebus+ addresses into 40-bit Laser Bus addresses.

For direct mapped I/O one mapping register in the FLAG must be loaded via a Mailbox Write Transaction for every page of data transferred to/from the Futurebus+.

A complete Mailbox Write Transaction requires at least of 6 Laser Bus Transactions. See the below chart.

```
Transaction Type              Quantity
----------------              --------

CPU Writes Mailbox          1 LSB Transaction <- MBX to Memory
CPU Writes LMBPR Reg        1 LSB Transaction <- CPU to IOP
IOP 1st Read of Mailbox     1 LSB Transaction <- MBX to I/O Adapter
IOP 2nd Read of Mailbox     1 LSB Transaction <- MBX Stat from I/O Adapter
IOP Writes Mailbox          1 LSB Transaction <- MBX Stat to Memory
CPU Reads Mailbox           1 LSB Transaction <- CPU Checks MBX Status
                          ------
                            6 LSB Transactions = 384 Bytes
```

A VAX page size is only 512 bytes. Thus, 384 bytes of LSB bandwidth is wasted on the Mailbox Transaction to transfer only 512 bytes of I/O data to/from Laser Memory. This is clearly not a very efficient use of LSB Bandwidth

It should be noted that with larger page sizes (I.E. 8 K Bytes) such as supported by EVMS and UNIX this performance problem becomes greatly reduced.

APPENDIX B

THE VORTEX BUS

B.1 Introduction

The Vortex bus is the physical interconnect between the LASER I/O port chips (IPC) and the Hose Interface Chips (HIC). The Vortex Bus consists of point to point connections between the IPCs and the HICs. These connections are shown in Figure B–1. The bus cycle time is equal to the LASER system bus cycle time of 17 nS. The Vortex Bus passes command, status, mask, parity, and data between the IPCs and the HICs. The Vortex Bus may be thought of as two separate 64-bit busses, one going up and the other going down. UP is defined as going from I/O to the LASER system bus, or from the Up HIC to the IPCs, while DOWN is defined as going from the LASER system bus to I/O, or from the IPCs to the Down HIC.

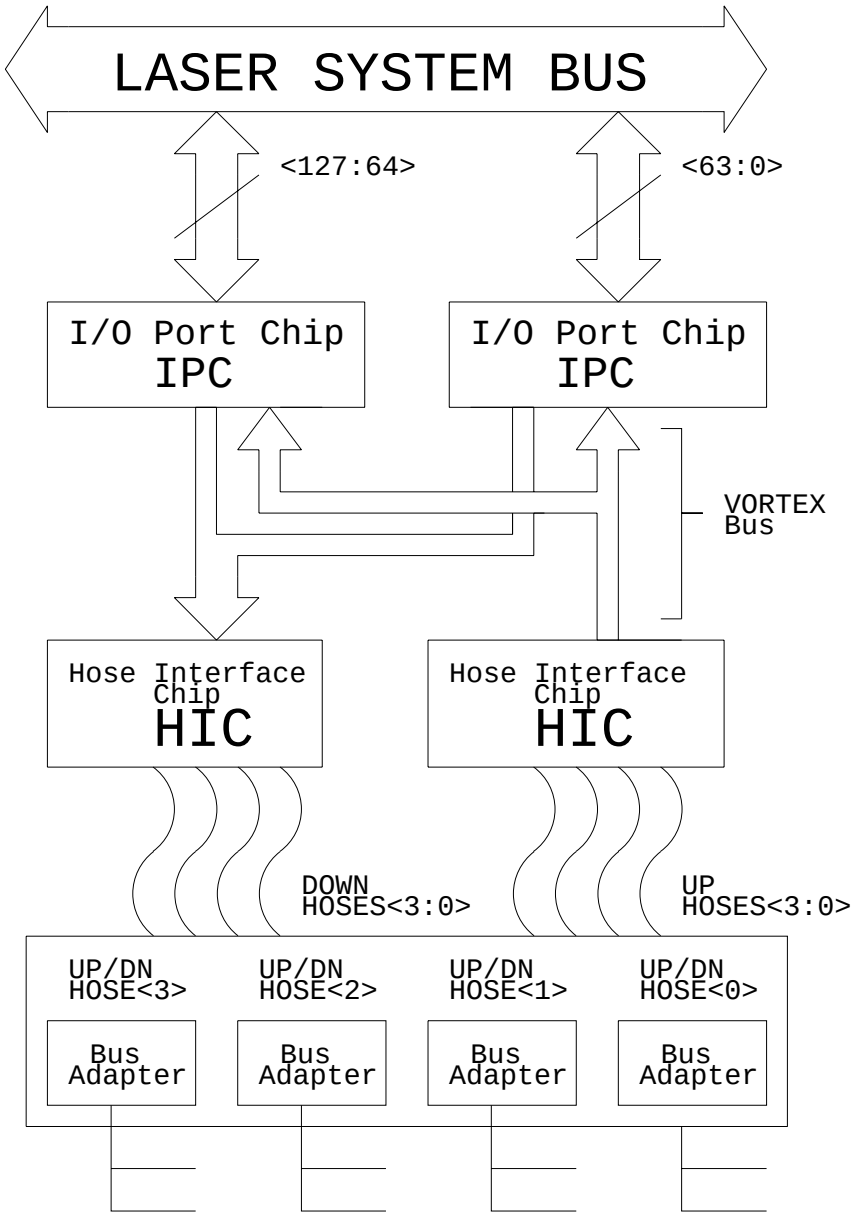
Each HIC chip is identical to the other. One HIC is used for UP packets, while the other is used for DOWN packets. The IPC chips are also identical to each other, and are "bit sliced", or more precisely, quadword sliced. 64 data lines are available on the HIC, and these lines split into two 32-bit busses, one from each IPC. This is true in both directions. One IPC interfaces to the upper quadword of the LASER system bus, while the other interfaces to the lower quadword. Since the HIC is always bus master for the Up Vortex, and the IPC is always master of the Down Vortex, there is no need for arbitration mechanisms. There is, however, a need for flow control. This is provided by the IPC_DEC_PKT_BUF and HIC_DN_DEC_PKT_CNT<3:0> signals, as well as a set of internal counters. These counters keep track of the available number of buffers. When a packet is transmitted, the appropriate counter is incremented. When a packet is removed from a buffer, the appropriate DEC_PK_CNT signal is asserted, causing the appropriate packet counter to be decremented. The corresponding bus master may use the bus if the slave has a packet buffer available.

Packets are sent across the Vortex Bus for the following reasons:

- An I/O node desires to interrupt a CPU
- A CPU desires access to an I/O node's CSRs (read or write)
- An I/O node desires access to Laser memory (read or write)
- Status of a previous packet (interrupts and mailbox commands)

Figure B–1 on the following page shows a block diagram of the I/O Port Module (IOP) while highlighting the Vortex bus.

Figure B–1: I/O Port Module Block (IOP) Block Diagram



NOTE:

A maximum configuration allows for 4 bus adapters. A minimum configuration would be one bus adapter.

B.2 Signal Descriptions

B.2.1 Conventions

By convention, the source of the signal prefixes the signal name, for example, HIC_EOP L, which is driven by the HIC chip, or IPC_DEC_PKT_BUF L, which is driven by the IPC. Unless otherwise indicated, all signals are asserted when the voltage level is low.

B.2.2 Up Vortex Bus Signals

Table B–1: Up Vortex Bus Signals

Signal Name	Explanation
HIC_CMD_UP_A<3:0> L HIC_CMD_UP_B<3:0> L	The 8 command lines accompanying the UP Vortex Bus data lines. HIC_CMD_UP_B<3:0> are connected to IPC-B, while HIC_CMD_UP_A<3:0> are connected to IPC-A. These lines also carry write mask information during DMA Write Masked command when write data is being driven on the HIC_UP_DATA<63:0> lines. The encoding of these lines is shown below.

HIC_CMD_UP_A<3:0> HIC_CMD_UP_B<3:0>	Command
0001	DMA Read Command
0010	DMA IREAD Command
0100	Mailbox Status Command
0101	DMA Unmasked Write
0111	DMA Masked Write
1000	INTR/IDENT Command

HIC_UP_HOSE_ID<1:0> L	These lines carry the hose ID number on the Up Vortex Bus. The hose ID reflects the hose number on which the I/O node resides which generated the packet.
HIC_UP_DATA<63:0> L	The 64 data lines for the Up Vortex Bus. Data lines HIC_UP_DATA<63:32> are connected to IPC-B, and data lines HIC_UP_DATA<31:0> to IPC-A.
HIC_UP_END_OF_PKT_A L HIC_UP_END_OF_PKT_B L	These lines indicate to the IPCs that the Up HIC is driving the last cycle of a packet on the Up Vortex Bus. HIC_UP_END_OF_PKT_A L goes to IPC-A, while HIC_UP_END_OF_PKT_B L goes to IPC-B. They are identical in all other respects. This line is used to allow for checking for the correct number of cycles in a packet, and also permits the differentiation of back-to-back packets on the Up Vortex bus.

Table B–1 (Cont.): Up Vortex Bus Signals

Signal Name	Explanation
HIC_UP_PAR<1:0> L	HIC_UP_PAR<0> is an ODD parity line protecting HIC_UP_DATA<31:0> L and HIC_UP_CMD_A<3:0>, which goes to IPC-A. HIC_UP_PAR<1> is an ODD parity line protecting HIC_UP_DATA<63:32> L and HIC_UP_CMD_B<3:0>, and goes to IPC-B.
HIC_UP_ERR_VALID L	Indicates to the IPCs that the Up HIC detected an error. These errors include, hose parity errors, hose packet errors, buffer overflow, and HIC internal errors.
HIC_UP_ERR_CODE<1:0> L	Indicates to the IPCs the type of error detected by the Up HIC. Valid only when HIC_UP_ERR_VALID is asserted. The encoding of these lines is shown below.

HIC_UP_ERR_CODE<1:0>	Error Condition
00	Packet error
01	Parity error
10	Overflow error
11	Internal error

HIC_UP_ERR_HOSE_ID<1:0> L	Indicates to the IPCs, on which hose the HIC detected the error. Valid only when HIC_UP_ERR_VALID is asserted. The encoding of these lines is shown below.

HIC_UP_ERR_HOSE_ID<1:0>	Hose in Error
00	Hose 0
01	Hose 1
10	Hose 2
11	Hose 3

HIC_UP_VTX__VALID_A L	
HIC_UP_VTX__VALID_B L	These lines indicates to the IPCs that the Up HIC is driving valid information onto the Up Vortex Bus. The _A variant goes to IPC-A, while the _B variant goes to IPC-B.
IPC_DEC_PKT_BUF L	Indicates to the Up HIC that the IPC has removed a packet from its packet buffer. The HIC maintains an internal counter representing the number of free buffers in the IPC. When the HIC sends a packet to the IPCs, the HIC increments this counter. When the IPC removes a packet from the buffer, the IPC asserts this signal, telling the HIC to decrement the packet counter. This line is driven by IPC-A only.

B.2.3 Down Vortex Bus Signals

Table B–2: Down Vortex Bus Signals

Signal Name	Explanation
IPC_DOWN_RESET<3:0> L	These are Reset lines used to reset each of the 4 hoses indivually.
HIC_DN_DEC_PKT_CNT<3:0> L	The IPC maintains a counter representing the number of free packet buffers in the HIC. When the IPC sends a packet to the HIC, the counter for the appropriate hose is incremented. When the HIC removes a packet from its buffer, it must tell the IPC to decrement the counter so that the IPC always knows how many available packet buffers are in the HIC. The assertion of this line indicates that the IPC must decrement the appropriate counter, ie. bit 0 is asserted to indicate that the counter for buffer 0 must be decremented, etc.
HIC_DN_ERROR L	When asserted, this line indicates to the IPC that the Down HIC detected an error. These errors include Down Vortex parity errors, Down Vortex Sequence errors, and Down HIC internal errors.
IPC_DN_END_OF_PKT L	When asserted, this line indicates to the Down HIC that the last cycle of a Down Vortex Bus packet is being driven on the bus.
IPC_DN_DATA<63:0> L	These lines are the 64 data lines for the Down Vortex Bus. Data lines IPC_DN_DATA<31:0> are connected to IPC-A, and data lines IPC_DN_DATA<63:32> to IPC-B. The Down HIC receives all 64 lines.
IPC_DN_VTX_VALID L	Indicates to the Down HIC that the IPC is driving valid information on the Down Vortex Bus.
IPC_DN_PAR<1:0> L	The ODD parity lines which protect the IPC_DN_DATA lines. IPC_DN_PAR<0> protects IPC_DN_DATA<31:0> and IPC_DN_PAR<1> protects IPC_DN_DATA<63:32>.
IPC_DN_HOSE_ID<1:0> L	Indicates to the HIC the hose destination number for the packet on the Down Vortex Bus.
IPC_LOOPBACK H	This line is asserted to put the IOP into diagnostic internal loopback mode. It causes the Down HIC to loopback Down Vortex Mailbox transactions to the Up HIC, which then get sent across the Up Vortex.

B.3 Transaction Descriptions

B.3.1 Up Vortex Bus Transactions

B.3.1.1 DMA Read Command

The DMA Read command is issued by an I/O node desiring to read Laser memory. The address field is 40 bits. This command may be issued in a variety of lengths, namely OW, hexaword (HW), and double hexaword (DHW). Associated with the DMA Read packet is a tag field. This field is generated by the I/O adapter, and is used to match DMA read data return with the corresponding DMA Read command. Each half of the Up Vortex Bus is identical during the two cycles of a DMA read packet. Each IPC needs the entire packet so that the LSB can be driven correctly.

B.3.1.2 DMA IREAD Command

The DMA IREAD command is issued by an I/O node desiring to read and lock a data structure in memory. This command will only be issued in the octaword length. When the IOP receives this command, it will issue a READ/MODIFY/WRITE on the LSB. The WRITE will set the lock bit in the data structure (bit 0 of the first quadword). The data returned to the I/O node will be the ORIGINAL unmodified data from memory. The tag field is used by the I/O adapter to associate returning read data with the corresponding DMA Read command.

B.3.1.3 MBOX Status

The MBOX (Mailbox) Status command is used for two things: 1) to return status to a CPU concerning a mailbox write command to an I/O node's CSR, and 2) to read data return and status to a CPU resulting from reading an I/O node's CSR. Currently, this packet is specified to allow for a maximum of a quadword of read data to be returned. The device specific field is I/O adapter implementation dependent. This field could be used for extended error information. The DONE bit shall be set, while the ERROR bit is set accordingly.

B.3.1.4 DMA Unmasked Write Command

The DMA Write command is issued by an I/O node desiring to write to Laser memory. The address field is 40 bits. The command may be issued with DHW lengths only. When issued, the lower quadword of data goes to IPC-A, and the upper quadword to IPC-B (see Figure B–4). This is done so that each IPC has the appropriate data to be driven on the LSB. IPC-B receives data which needs to be driven on LSB<127:64>, while IPC-A receives data to be driven on LSB<63:0>.

B.3.1.5 DMA Masked Write Command

This command is the same as the DMA Unmasked Write command, with the exception that mask information, carried on the HIC_CMD_UP_x<3:0> lines (while write data is carried on the HIC_D_UP<63:0> lines), is now significant. This command can be issued in lengths of OW, HW, or DHW.

B.3.1.6 INTR/IDENT

An I/O node wishing to interrupt the CPU will generate an interrupt packet on its I/O bus. The I/O bus adapter will then issue whatever packets are necessary to obtain an interrupt vector from the interrupting node. In the case of the XMI bus, this would be an IDENT packet. Once the interrupt and vector are obtained by the I/O adapter, this information, along with the interrupt priority level, is sent up the hose to the Up HIC, and then across the Up Vortex Bus.

B.3.2 Down Vortex Bus transactions

B.3.2.1 DMA Read Data Return

This packet is used to return data from Laser memory to a requesting I/O node. The tag field from the corresponding DMA Read command is inserted into this packet for the purpose of matching the DMA Read Command with the resulting DMA Read Data Return. Note the asymmetry regarding the longword data organization on the Down Vortex bus (see Figure B–2). This is due to the fact that IPC-B received LSB<127:64>, and therefore the upper quadword of data, while IPC-A receives LSB<63:0>, or the lower quadword of data. Thus, on the Down Vortex bus, IPC-A drives LW_DATA1 while IPC-B drives LW_DATA3, followed in the next cycle by IPC-A driving LW_DATA2, while IPC-B drives LW_DATA4. This process continues until all of the requested read data has been transferred across the bus.

B.3.2.2 INTR/IDENT Status

This packet returns to the I/O adapter the status of an INTR/IDENT command. This status is used by the I/O adapter to clear interrupt pending flops. The IPL field is the same IPL as that used in the INTR/IDENT command, and is used for matching status to the originating command. The "E" field is an error bit, set accordingly.

B.3.2.3 MBOX Command

This packet sends a mailbox (MBOX) command to an I/O node. Mailbox commands are constructed in a data structure in Laser memory by a CPU desiring to access a CSR on an I/O node. The IPC fetches these commands out of Laser memory after the LMBPR register is written with the address of the mailbox data structure. Included in the command are the I/O target address, which is 64 bits, a 32 bit command field, and 2 longwords of data. The hose number identifies the hose onto which the targetted I/O node resides. The mask field is used for byte masking. The command field is I/O node specific.

Table B–3: Summary of I/O to LASER Packets

HIC CMD UP<3:0>	Command	Explanation
0001	DMA Read	I/O node reads LASER memory.
0010	DMA IRead	I/O node reads LASER memory and sets the lock bit in the data structure.
0100	MBOX Status	The status and/or read data associated with a mailbox packet.
0101	DMA Unmasked Write	I/O node writes LASER memory.
0111	DMA Masked Write	I/O node does a masked write of memory.
1000	INTR/IDENT	Interrupt and vector from I/O node.

Table B–4: Summary of LASER to I/O packets

IPC_DN_DATA<13:12>	Command	Explanation
00	INTR/IDENT Status	Indicates status of an interrupt on the LSB.
01	DMA Read Data Return	The data from memory, being returned as the result of a DMA Read.
10	MBOX CMD	A mailbox command initiated by a CPU desiring to read or write an I/O node's CSRs.

Table B–5: Summary of packet pairs - Command vs. Response

Command	Response	Explanation
DMA Read	DMA Read Return	
	Read Data	I/O reads memory and gets data back.
DMA Write with write data	None	I/O writes memory and includes the data.
INTR/IDENT	INTR/IDENT Status	I/O interrupts a CPU and gets status back upon completion.
MBOX CMD	MBOX Status	The CPU sets up an I/O packet in a mailbox in system memory. The MBOX CMD issues the command to the I/O node, and MBOX status is returned.

Table B–6: Length field encoding

Length Code	Data length
000	Hexaword (32 bytes)
001	Longword (4 bytes)
010	Quadword (8 bytes)
011	Octaword (16 bytes)
100	Double Hexaword(64 bytes)

B.3.3 Data flow from HOSE to VORTEX and vice versa

During normal operation of the IOP, the data flowing from the HOSE to the VORTEX and then to the LASER system bus undergoes some manipulations. These manipulations are necessary since the HOSE is a 32 bit wide bus, the VORTEX is 64 bits wide, and the LSB is 128 bits wide.

The HIC takes the 32 bit packet format from the HOSE and converts it to the 64 bit packet format of the VORTEX. When the IPC receives this data from the VORTEX, it converts it to the 128 bit LSB format.

Similarly in the down direction, the LSB format is converted to the VORTEX format by the IPC. The Down HIC then converts the VORTEX to the HOSE format. These packet format changes from bus to bus are shown in Figure B–2, Figure B–3, and Figure B–4.

Figure B–2: Down Vortex Bus to Down HOSE Packet Conversion

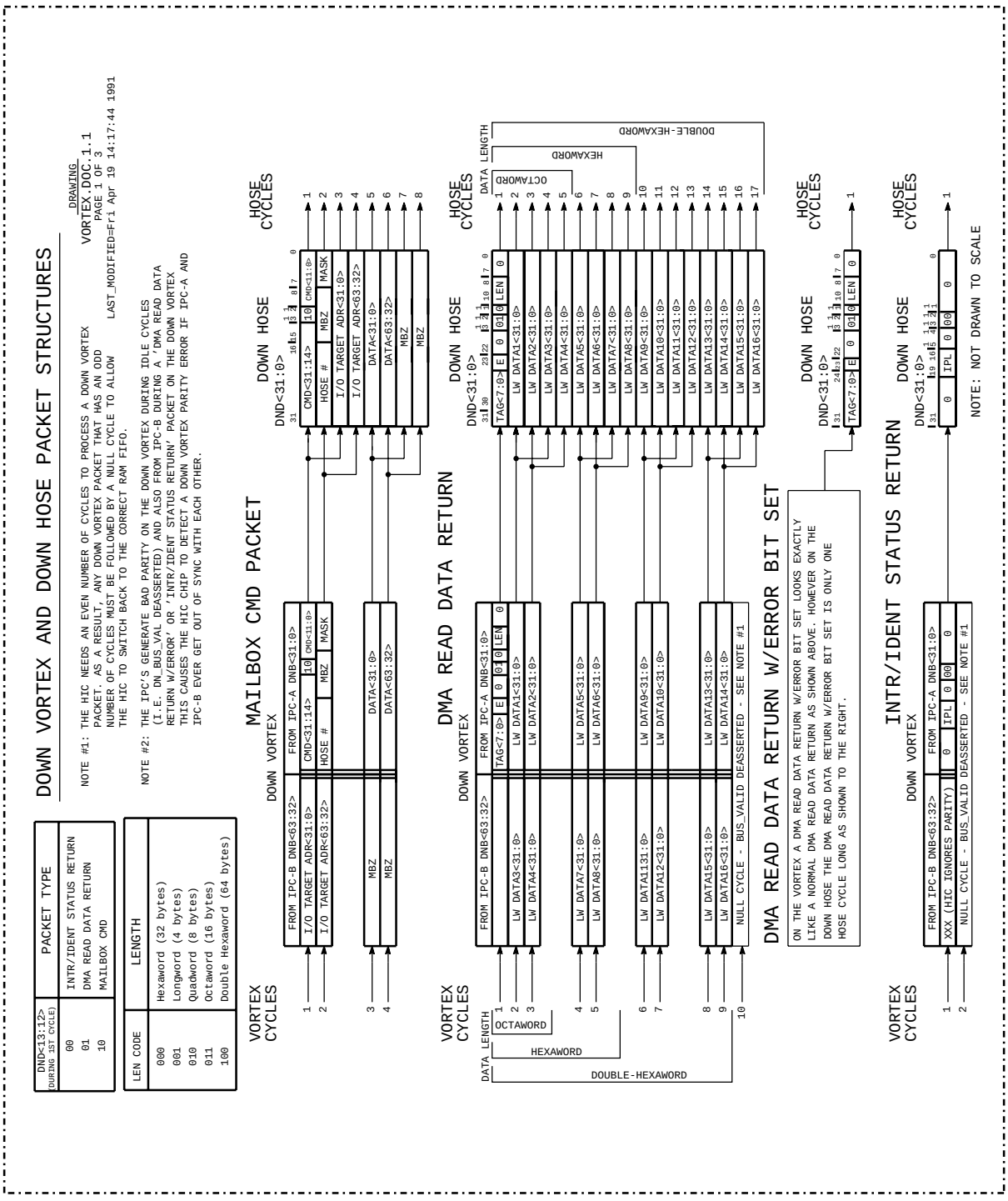


Figure B–3: Up HOSE to Up Vortex Bus Packet Conversion

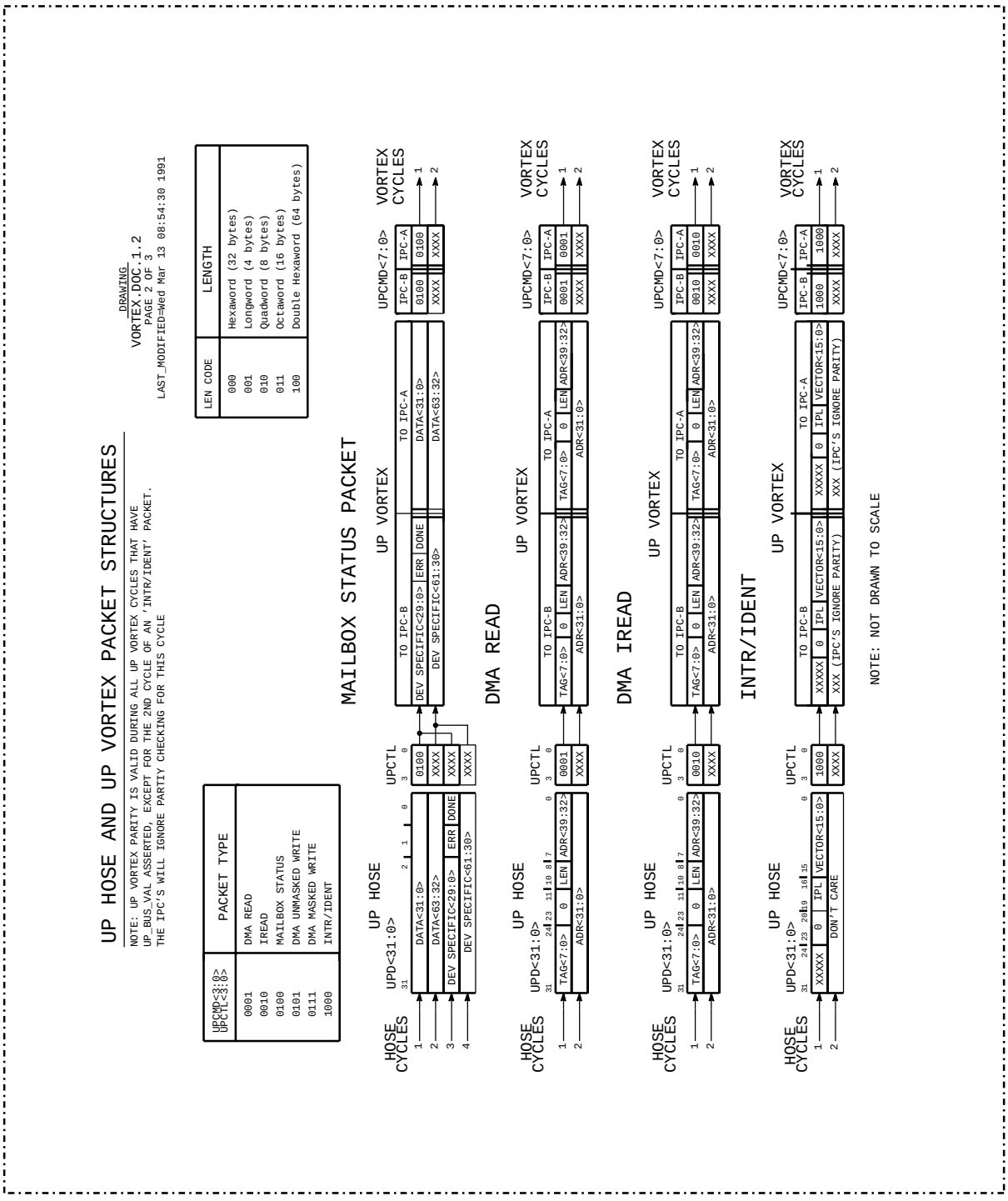
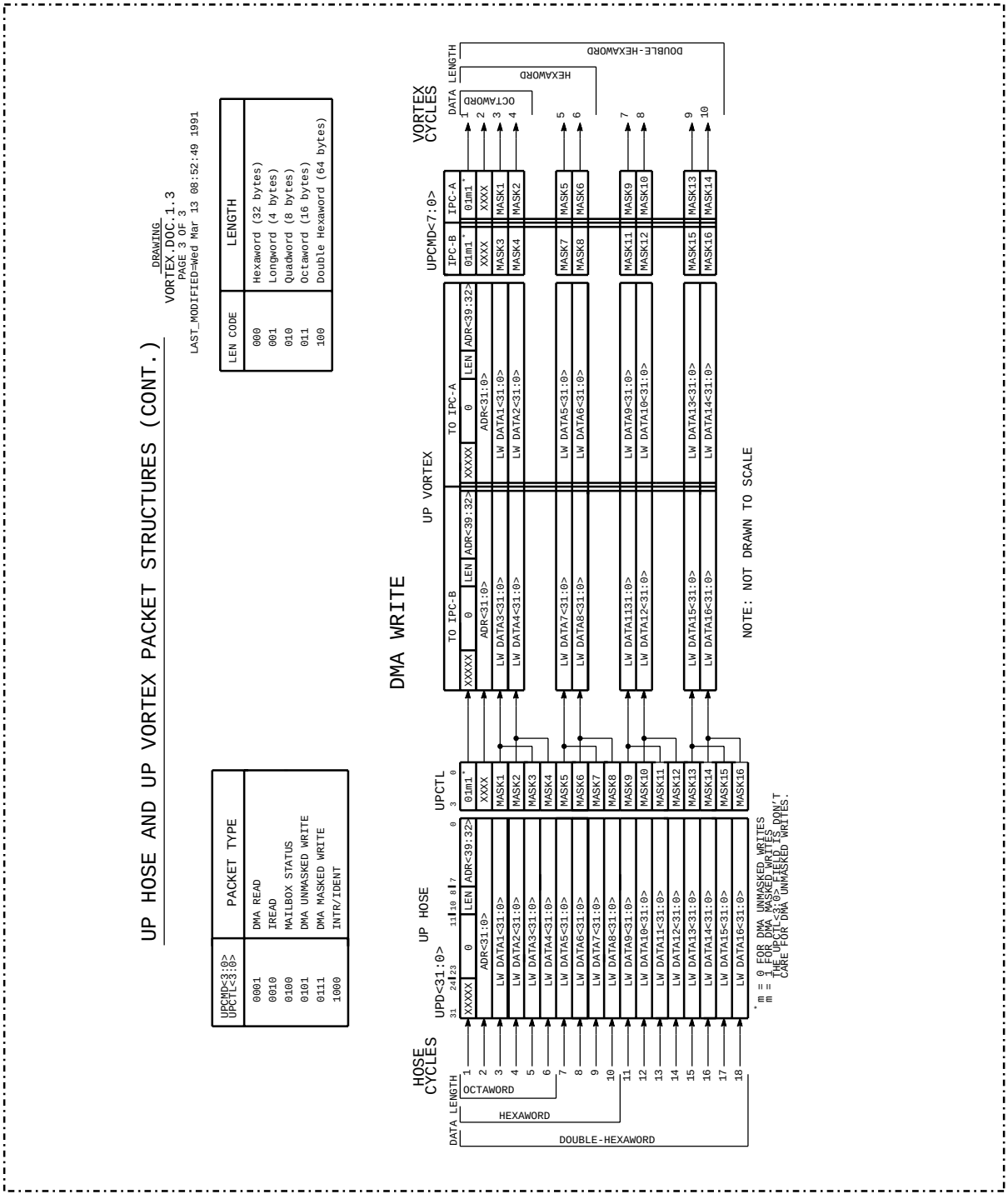


Figure B–4: Up HOSE to Up Vortex Bus Packet Conversion for DMA Write



INDEX