

# **PCI-HPDI32A-** **DIPHASE2**

# **User Manual**

**Preliminary**

**General Standards Corporation**  
**8302A Whitesburg Drive**  
**Huntsville, AL 35802**  
**Phone: (256) 880-8787**  
**Fax: (256) 880-8788**  
**URL: [www.generalstandards.com](http://www.generalstandards.com)**  
**E-mail: [support@generalstandards.com](mailto:support@generalstandards.com)**

CREATED ON JUNE 21, 2001

## **PREFACE**

---

### **General Standards Corporation**

Preliminary, Revised: July 20, 2000

Copyright (C) 2000 **General Standards Corp.**

Additional copies of this manual or other literature may be obtained from:

**General Standards Corporation**  
8302A Whitesburg Dr.  
Huntsville, Alabama 35802  
Tele: (256) 880-8787  
FAX: (256) 880-8788  
E-mail: [support@generalstandards.com](mailto:support@generalstandards.com)

The information in this document is subject to change without notice.

**General Standards Corp.** makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Although extensive editing and reviews are performed before release to ECO control, **General Standards Corp.** assumes no responsibility for any errors that may exist in this document. No commitment is made to update or keep current the information contained in this document.

**General Standards Corp.** does not assume any liability arising out of the application or use of any product or circuit described herein, nor is any license conveyed under any patent rights or any rights of others.

**General Standards Corp.** assumes no responsibility for any consequences resulting from omissions or errors in this manual, or from the use of information contained herein.

**General Standards Corp.** reserves the right to make any changes, without notice, to this product to improve reliability, performance, function, or design.

### **All rights reserved**

No part of this document may be copied or reproduced in any form or by any means without prior written consent of **General Standards Corp.**

This user's manual provides information on the specifications, theory of operation, register level programming, installation of the board and information required for customized hardware/software development.

**RELATED PUBLICATIONS**

The following manuals and specifications provide the necessary information for in depth understanding of the specialized parts used on this board.

EIA Standard for the RS-422A Interface (EIA order number EIA-RS-422A)

PCI Local Bus Specification Revision 2.1 June 1, 1995. Questions regarding the PCI specification be forwarded to:

PCI Special Interest Group  
P.O. Box 14070  
Portland, OR 97214  
(800) 433- 5177 (U.S.)  
(503) 797-4207 (International)  
(503) 234-6762 (FAX)

**PCI-HPDI32A-DIPHASE2 Documentation History**

- 1) June 21, 2001 – Initial Release.

# Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>5</b>
<b>1.1</b>	<b>FUNCTIONAL DESCRIPTION .....</b>	<b>5</b>
<b>1.2</b>	<b>THEORY OF OPERATION .....</b>	<b>6</b>
1.2.1	TRANSMITTER OPERATION: .....	7
1.2.2	RECEIVER OPERATION: .....	9
<b>1.3</b>	<b>CABLE INTERFACE .....</b>	<b>11</b>
<b>1.4</b>	<b>FIFOS .....</b>	<b>11</b>
<b>2</b>	<b>REGISTERS .....</b>	<b>12</b>
<b>2.1</b>	<b>PCI-HPDI32A- DIPHASE2 REGISTER INFORMATION .....</b>	<b>12</b>
<b>2.2</b>	<b>PCI-HPDI32A-DIPHASE2 LOCAL REGISTERS, BIT DESCRIPTIONS .....</b>	<b>13</b>
2.2.1	FIRMWARE REVISION: (OFFSET 0x00 - RO) .....	14
2.2.2	BOARD CONTROL: (OFFSET 0x04 - RW) .....	14
2.2.3	BOARD STATUS: (OFFSET 0x08 - RO) .....	16
2.2.4	TX ALMOST: (OFFSET 0x0C - RW) .....	17
2.2.5	RX ALMOST: (OFFSET 0x10 - RW) .....	17
2.2.6	FIRMWARE FEATURES: (OFFSET 0x14 - RW) .....	18
2.2.7	TX OUTPUT DATA PORT REGISTER: (OFFSET 0x14 - RW) .....	18
2.2.8	FIFO I/O: (OFFSET 0x18 - RW) .....	18
2.2.9	TX STATUS LENGTH REGISTER: (OFFSET 0x1C - RO) .....	18
2.2.10	TX ROW VALID LENGTH REGISTER: (OFFSET 0x20 - RW) .....	18
2.2.11	TX ROW INVALID LENGTH REGISTER: (OFFSET 0x24 - RW) .....	18
2.2.12	RX STATUS BLOCK LENGTH COUNTER: (OFFSET 0x28 - RO) .....	19
2.2.13	RX ROW LENGTH COUNTER: (OFFSET 0x2C - RO) .....	19
2.2.14	INTERRUPT CONTROL: (OFFSET 0x30 - RW) .....	20
2.2.15	INTERRUPT STATUS: (OFFSET 0x34 - RC) .....	22
2.2.16	TX CLOCK DIVIDER: (OFFSET 0x38 - RW) .....	25
2.2.17	RESERVED: (OFFSET 0x3C - RW) .....	25
2.2.18	TX FIFO SIZE: (OFFSET 0x40 - RO) .....	26
2.2.19	RX FIFO SIZE: (OFFSET 0x44 - RO) .....	26
2.2.20	TX FIFO WORD COUNT: (OFFSET 0x48 - RO) .....	26
2.2.21	RX FIFO WORD COUNT: (OFFSET 0x4C - RO) .....	26
2.2.22	INTERRUPT EDGE REGISTER: (OFFSET 0x50 - RW) .....	27
2.2.23	INTERRUPT HI/LO REGISTER: (OFFSET 0x54 - RW) .....	27
<b>3</b>	<b>PROGRAMMING .....</b>	<b>28</b>
<b>3.1</b>	<b>INITIALIZATION .....</b>	<b>28</b>
<b>3.2</b>	<b>RESETS .....</b>	<b>28</b>
<b>3.3</b>	<b>FIFOS .....</b>	<b>29</b>
<b>3.4</b>	<b>INTERRUPTS .....</b>	<b>30</b>
<b>4</b>	<b>HARDWARE CONFIGURATION .....</b>	<b>31</b>

<b>4.1</b>	<b>THE ON-BOARD TRANSMIT CLOCK .....</b>	<b>31</b>
<b>4.2</b>	<b>JUMPERS (J5).....</b>	<b>32</b>
<b>4.3</b>	<b>CABLE CONNECTOR.....</b>	<b>33</b>
<b>Appendix A</b>	<b>PLX REGISTER PROGRAMMING.....</b>	<b>34</b>
<b>A.1</b>	<b>CONFIGURATION EEPROM .....</b>	<b>34</b>
<b>A.2</b>	<b>PCI CONFIGURATION REGISTER BIT MAPS .....</b>	<b>35</b>
A.2.1	PCI CONFIGURATION ID REGISTER (PCI CONFIGURATION OFFSET 0x00).....	35
A.2.2	PCI COMMAND REGISTER (PCI CONFIGURATION OFFSET 0x04) .....	36
A.2.3	PCI STATUS REGISTER (PCI CONFIGURATION OFFSET 0x06) .....	37
A.2.4	PCI REVISION ID REGISTER (PCI CONFIGURATION OFFSET 0x08) .....	37
A.2.5	PCI CLASS CODE REGISTER (PCI CONFIGURATION OFFSET 0x09..0B).....	38
A.2.6	PCI CACHE LINE SIZE REGISTER (PCI CONFIGURATION OFFSET 0x0C).....	38
A.2.7	PCI LATENCY TIMER REGISTER (PCI CONFIGURATION OFFSET 0x0D).....	38
A.2.8	PCI HEADER TYPE REGISTER (PCI CONFIGURATION OFFSET 0x0E).....	38
A.2.9	PCI BUILT-IN SELF TEST (BIST) REGISTER (PCI CONFIGURATION OFFSET 0x0F).....	38
A.2.10	PCI BASE ADDRESS REGISTER FOR MEMORY ACCESS TO RUNTIME REGISTERS (PCI CONFIGURATION OFFSET 0x010)	39
A.2.11	PCI BASE ADDRESS REGISTER FOR I/O ACCESS TO RUNTIME REGISTERS (PCI CONFIGURATION OFFSET 0x14)	39
A.2.12	PCI BASE ADDRESS REGISTER FOR MEMORY ACCESS TO LOCAL ADDRESS SPACE 0 (PCI CONFIGURATION OFFSET 0x18).....	40
A.2.13	PCI BASE ADDRESS REGISTER FOR MEMORY ACCESS TO LOCAL ADDRESS SPACE 1 (PCI CONFIGURATION OFFSET 0x1C) .....	40
A.2.14	PCI BASE ADDRESS REGISTER (PCI CONFIGURATION OFFSET 0x20).....	40
A.2.15	PCI BASE ADDRESS REGISTER (PCI CONFIGURATION OFFSET 0x24).....	40
A.2.16	PCI CARDBUS CIS POINTER REGISTER (PCI CONFIGURATION OFFSET 0x28).....	40
A.2.17	PCI SUBSYSTEM VENDOR ID REGISTER (PCI CONFIGURATION OFFSET 0x2C).....	41
A.2.18	PCI SUBSYSTEM ID REGISTER (PCI CONFIGURATION OFFSET 0x2E).....	41
A.2.19	PCI EXPANSION ROM BASE REGISTER (PCI CONFIGURATION OFFSET 0x30) .....	41
A.2.20	PCI INTERRUPT LINE REGISTER (PCI CONFIGURATION OFFSET 0x3C).....	41
A.2.21	PCI INTERRUPT PIN REGISTER (PCI CONFIGURATION OFFSET 0x3D).....	41
A.2.22	PCI MIN_GNT REGISTER (PCI CONFIGURATION OFFSET 0x3E) .....	41
A.2.23	PCI MAX_LAT REGISTER (PCI CONFIGURATION OFFSET 0x3F) .....	41
<b>A.3</b>	<b>LOCAL CONFIGURATION REGISTERS.....</b>	<b>42</b>
A.3.1	LOCAL ADDRESS SPACE 0 RANGE REGISTER FOR PCI TO LOCAL BUS (PCI OFFSET 0x00).....	43
A.3.2	LOCAL ADDRESS SPACE 0 LOCAL BASE ADDRESS (RE-MAP) REGISTER FOR PCI TO LOCAL BUS (PCI OFFSET 0x04)	43
A.3.3	MODE/ARBITRATION REGISTER (PCI OFFSET 0x08).....	44
A.3.4	BIG/LITTLE ENDIAN DESCRIPTOR REGISTER (PCI OFFSET 0x0C).....	45
A.3.5	LOCAL EXPANSION ROM RANGE REGISTER FOR PCI TO LOCAL BUS (PCI OFFSET 0x10).....	46
A.3.6	LOCAL EXPANSION ROM LOCAL BASE ADDRESS (RE-MAP) REGISTER FOR PCI TO LOCAL BUS AND BREQO CONTROL (PCI OFFSET 0x14).....	46
A.3.7	LOCAL ADDRESS SPACE 0/EXPANSION ROM BUS REGION DESCRIPTOR REGISTER (PCI OFFSET 0x18).....	46
A.3.8	LOCAL RANGE REGISTER FOR DIRECT MASTER TO PCI (PCI OFFSET 0x1C).....	47
A.3.9	LOCAL BUS BASE ADDRESS REGISTER FOR DIRECT MASTER TO PCI MEMORY (PCI OFFSET 0x20).....	47
A.3.10	LOCAL BASE ADDRESS FOR DIRECT MASTER TO PCI IO/CFG REGISTER (PCI OFFSET 0x24).....	48
A.3.11	PCI BASE ADDRESS (RE-MAP) REGISTER FOR DIRECT MASTER TO PCI MEMORY (PCI OFFSET 0x28).....	48
A.3.12	PCI CONFIGURATION ADDRESS REGISTER FOR DIRECT MASTER TO PCI IO/CFG (PCI OFFSET 0x2C).....	49
A.3.13	LOCAL ADDRESS SPACE 1 RANGE REGISTER FOR PCI TO LOCAL BUS (PCI OFFSET 0xF0).....	49
A.3.14	LOCAL ADDRESS SPACE 1 LOCAL BASE ADDRESS (REMAP) REGISTER (PCI OFFSET 0xF4) .....	50
A.3.15	LOCAL ADDRESS SPACE 1 BUS REGION DESCRIPTOR REGISTER (PCI OFFSET 0xF8) .....	50
<b>A.4</b>	<b>RUNTIME REGISTERS .....</b>	<b>51</b>
A.4.1	MAILBOX REGISTER 0 (PCI OFFSET 0x40).....	51

A.4.2	MAILBOX REGISTER 1 (PCI OFFSET 0x44).....	51
A.4.3	MAILBOX REGISTER 2 (PCI OFFSET 0x48).....	51
A.4.4	MAILBOX REGISTER 3 (PCI OFFSET 0x4C).....	52
A.4.5	MAILBOX REGISTER 4 (PCI OFFSET 0x50).....	52
A.4.6	MAILBOX REGISTER 5 (PCI OFFSET 0x54).....	52
A.4.7	MAILBOX REGISTER 6 (PCI OFFSET 0x58).....	52
A.4.8	MAILBOX REGISTER 7 (PCI OFFSET 0x5C).....	52
A.4.9	PCI TO LOCAL DOORBELL REGISTER DESCRIPTION (PCI OFFSET 0x60).....	52
A.4.10	LOCAL TO PCI DOORBELL REGISTER DESCRIPTION (PCI OFFSET 0x64).....	52
A.4.11	INTERRUPT CONTROL /STATUS (PCI OFFSET 0x68).....	53
A.4.12	SERIAL EEPROM CONTROL, PCI COMMAND CODES, USER I/O CONTROL, INIT CONTROL REGISTER (PCI OFFSET 0x6C).....	55
A.4.13	PCI PERMANENT CONFIGURATION ID REGISTER (PCI OFFSET 0x70).....	56
A.4.14	PCI PERMANENT REVISION ID REGISTER (PCI OFFSET 0x74).....	56
<b>A.5</b>	<b>LOCAL DMA REGISTERS.....</b>	<b>57</b>
A.5.1	DMA CHANNEL 0 MODE REGISTER (PCI OFFSET 0x80).....	57
A.5.2	DMA CHANNEL 0 PCI ADDRESS REGISTER (PCI OFFSET 0x84).....	58
A.5.3	DMA CHANNEL 0 LOCAL ADDRESS REGISTER (PCI OFFSET 0x88).....	58
A.5.4	DMA CHANNEL 0 TRANSFER SIZE (BYTES) REGISTER (PCI OFFSET 0x8C).....	59
A.5.5	DMA CHANNEL 0 DESCRIPTOR POINTER REGISTER (PCI OFFSET 0x90).....	59
A.5.6	DMA CHANNEL 1 MODE REGISTER (PCI OFFSET 0x94).....	59
A.5.7	DMA CHANNEL 1 PCI DATA ADDRESS REGISTER (PCI OFFSET 0x98).....	60
A.5.8	DMA CHANNEL 1 LOCAL DATA ADDRESS REGISTER (PCI OFFSET 0x9C).....	60
A.5.9	DMA CHANNEL 1 TRANSFER SIZE (BYTES) REGISTER (PCI OFFSET 0xA0).....	60
A.5.10	DMA CHANNEL 1 DESCRIPTOR POINTER REGISTER (PCI OFFSET 0xA4).....	60
A.5.11	DMA COMMAND/STATUS REGISTER (PCI OFFSET 0xA8).....	61
A.5.12	DMA CHANNEL 1 COMMAND/STATUS REGISTER 0 (PCI OFFSET 0xA8).....	61
A.5.13	DMA ARBITRATION REGISTER 1 (PCI OFFSET 0xAC).....	62
A.5.14	DMA THRESHOLD REGISTER 1 (PCI OFFSET 0xB0).....	63
<b>A.6</b>	<b>MESSAGING QUEUE REGISTERS.....</b>	<b>64</b>
A.6.1	OUTBOUND POST LIST FIFO INTERRUPT STATUS REGISTER (PCI OFFSET 0x30).....	64
A.6.2	OUTBOUND POST LIST FIFO INTERRUPT STATUS REGISTER (PCI OFFSET 0x34).....	64
A.6.3	INBOUND QUEUE PORT REGISTER (PCI OFFSET 0x40).....	65
A.6.4	OUTBOUND QUEUE PORT REGISTER (PCI OFFSET 0x44).....	65
A.6.5	MESSAGING QUEUE CONFIGURATION REGISTER (PCI OFFSET 0xC0).....	65
A.6.6	QUEUE BASE ADDRESS REGISTER (PCI OFFSET 0xC4).....	65
A.6.7	INBOUND FREE HEAD POINTER REGISTER (PCI OFFSET 0xC8).....	66
A.6.8	INBOUND FREE HEAD TAIL REGISTER (PCI OFFSET 0xCC).....	66
A.6.9	INBOUND POST HEAD POINTER REGISTER (PCI OFFSET 0xD0).....	66
A.6.10	INBOUND POST TAIL POINTER REGISTER (PCI OFFSET 0xD4).....	66
A.6.11	OUTBOUND FREE HEAD POINTER REGISTER (PCI OFFSET 0xD8).....	66
A.6.12	OUTBOUND FREE TAIL POINTER REGISTER (PCI OFFSET 0xDC).....	66
A.6.13	OUTBOUND POST HEAD POINTER REGISTER (PCI OFFSET 0xE0).....	67
A.6.14	OUTBOUND POST TAIL POINTER REGISTER (PCI OFFSET 0xE4).....	67
A.6.15	QUEUE STATUS/CONTROL REGISTER (PCI OFFSET 0xE8).....	67



## **Table of Figures**

Figure 1.1-1 Block Diagram .....	5
Figure 1.2-1 Basic Cable Sync .....	6
Figure 1.2-2 Cable Data Encoding.....	6
Figure 2.2-1 Register Map .....	13
Figure 4.1-1 Oscillator Pinout .....	31
Figure 4.2-1 Jumper J5 .....	32
Figure 4.3-1 Cable Pinout.....	33

## **Table of Tables**

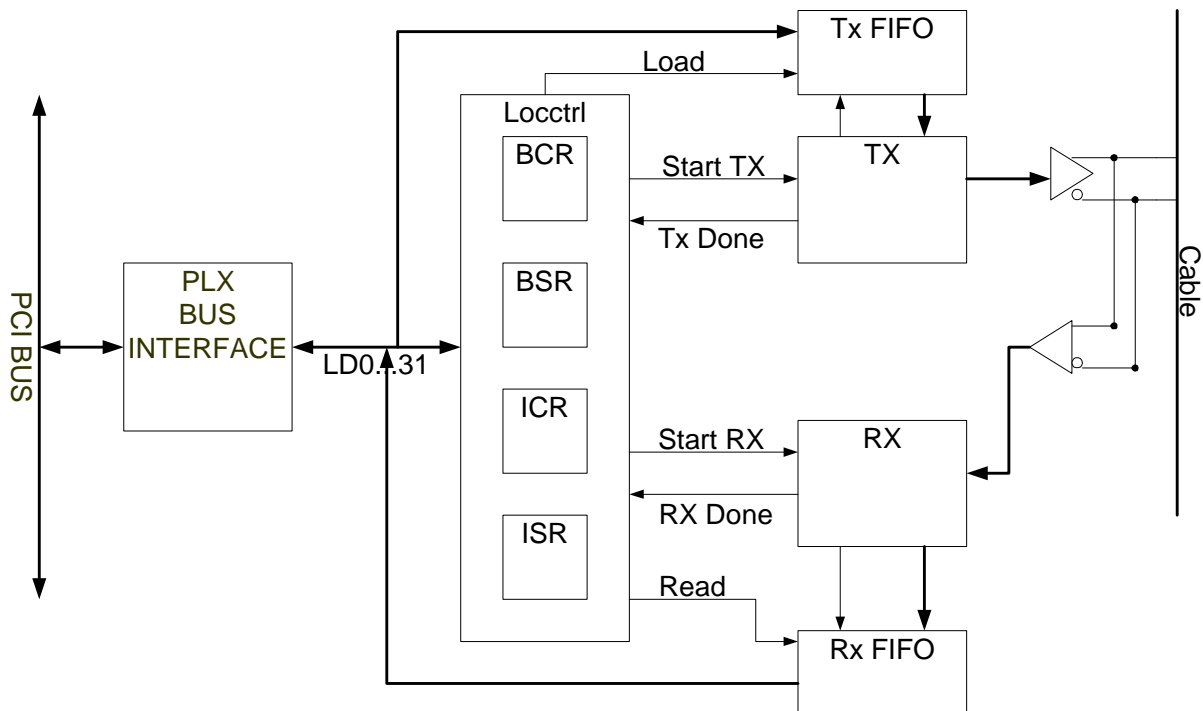
Table 1.2-1 Message Size Encoding.....	9
Table A Configuration EEPROM Contents.....	34
Table B Pci Configuration Register Mapping.....	35
Table C LOCAL CONFIGURATION REGISTERS .....	42
Table D RUN TIME REGISTERS .....	51
Table E DMA REGISTERS .....	57
Table F MESSAGING QUEUE REGISTERS .....	64

# 1 INTRODUCTION

## 1.1 FUNCTIONAL DESCRIPTION

The PCI-HPDI32A-DIPHASE2 Board includes a DMA Controller, 64/128/256/512 Kbytes of FIFO buffering, a cable Transmit controller, a cable Receive controller, and cable transceivers (RS-422/485 or differential pseudo ECL). The DMA on this board is intended for reading and writing the FIFOs. After the DMA is initialized and started, the host CPU will be free to proceed with other duties and need to respond only to interrupts. The DMA controller is capable of transferring data to host memory using D32 transfers; whereas the FIFO memory provides a means for continuous transmission of data without interrupting the DMA or requiring intervention from the host CPU. The board also provides for DMA chaining, interrupt generation for various states of the board, including End-Of-Transfer, TX FIFO Almost Empty, RX FIFO Almost Full, and more. The Transmitter and Receiver are in FPGAs that provide a configurable interface that is highly flexible in data width and transfer protocol.

Figure 1.1-1 Block Diagram



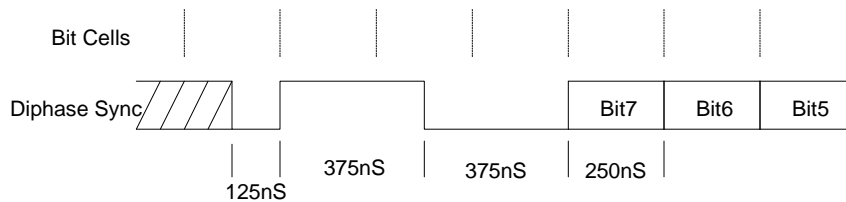
## 1.2 Theory of Operation

The transmitter and receiver of the standard HPDI32A board have been replaced by a Diphase serial Data Transmitter and Diphase serial data receiver with the following characteristics:

- Data rate of 4.0 +/-0.1% megabits per second
- Manchester II di-phase data encoding..
  - Logic one (1) is transmitted as a positive pulse followed by a negative pulse with the transition occurring at the mid point of the bit cell time.
  - Logic zero (0) is transmitted as a negative pulse followed by a positive pulse with the transition occurring at the mid point of the bit cell time.
- A transition through zero occurs at the midpoint of each bit cell time.
- Each byte transferred is 8 bits long.
- Each byte is transferred MSB, Bit 7, First.
- Each message block transferred shall start with a Sync signal: an invalid Manchester Sync, 3 bit times wide, the waveform being positive for the first one and one-half bit times and negative for the following one and one-half bit times.
- There is no sync signal between bytes within the message block.
- Bytes within the message block are transferred continuously without a gap between bytes.

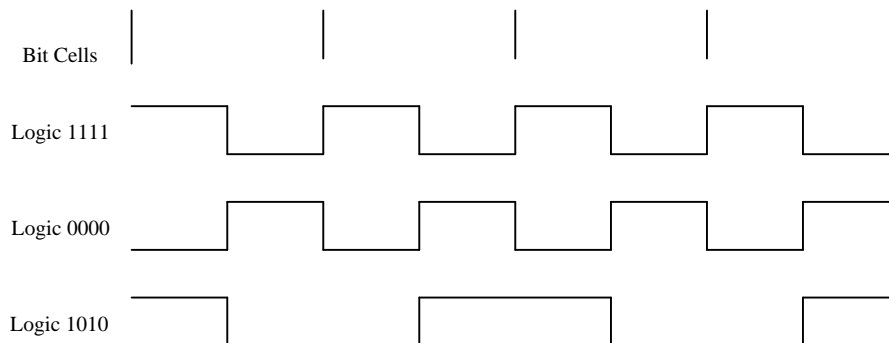
The Sync signal is shown in Figure 1.2-1 below.

**Figure 1.2-1 Basic Cable Sync**



Examples of data bit encoding are shown in Figure 1.2-2 below.

**Figure 1.2-2 Cable Data Encoding**



The PCI-HPDI32A-Diphase2 board also contains 32 programmable I/O Lines that are under software control. The lines are driven onto the cable on the Cable Data D0 thru Cable Data D31 Lines. Tri-state control on a byte wide basis is available thru the Board Control Register.

### 1.2.1 Transmitter Operation:

The PCI-HPDI32A-DIPHASE2 contains a Diphase data Transmitter, Diphase data Receiver, and 32 programmable I/O lines. When reset, the PCI-HPDI32A-DIPHASE2 will tri-state 'hi-z' the Diphase data line and the 32 programmable I/O Lines. The board will tri-state 'on' and drive the 2 lines that indicate if the Transmitter and Receiver are running. They control signals are as follows:

- Diphase Data Line – Reset to tri-State 'hi-z'.
- Transmitter Driving the Cable – Reset to tri-state 'on' and driven to logic low.
- Receiver Running – Reset to tri-state 'on' and driven to logic low.
- 32 programmable I/O Lines - Reset to tri-State 'hi-z'.

The transmitter is very flexible with mode bits and software controlled counters to control the following Software Selectable options:

- Big Endian / Little Endian
- Number of D8 bytes per Message.
- Pre-Drive Enable under software control.
- Pre-Drive Level select under software control.
- Pre-Drive Length under software control.

The Transmitter will also generate the following Status that may be read thru the Board Status Register and may generate Interrupts thru the Interrupt Control and Interrupt Status Registers.

- Tx Transmitter Driving the Cable – This board is driving the Cable
- Tx Transmit Busy – The Transmitter is sending the Sync, or data bits.
- Tx Transmit Done – The message Transmit is complete.

When a transmit is started by the Host processor the Transmitter will begin providing any Pre-Drive that has been selected by the Host processor. After the Pre-Drive is complete and there is data in the Transmit FIFO then the Transmitter will begin sending the Data Sync. The Data Sync will always begin with 125nS of a logic low to separate the Sync from a High Pre-Drive or from a Floating High cable. The Data Sync will begin with one and one half bit cell times, 375 nSec, of a logic high followed by one and one half bit cell times, 375 nSec, of a logic low. After the Data Sync, the Transmitter will begin sending data in 8 bit bytes with the MSB, Bit 7, always being transmitted first. Bytes shall be transmitted back to back with no further Syncs and no gaps between bytes. The transmit shall continue until the Transmit FIFO is Empty, or until the Transmit Row Length counter has been decremented down to zero. At the end of the Data Transmit, the last state of the bus shall be held for 375 nSec before the Transmitter asserts Transmit Done. When the Transmitter asserts Transmit Done, the board shall automatically Tri-State OFF , 'hi-z', the driver for the Diphase Data Line.

#### 1.2.1.1 TX Big Endian Mode – BCR D16

Normally, the PCI-HPDI32A-DIPHASE2 will transmit the D32 FIFO word as 4 D8 cable words, with D7..D0 being transmitted first, D15..D8 being transmitted second, D23..D16 being transmitted third, and D31..D24 being transmitted fourth. When this bit is set then the data will be taken out of the FIFO in Big Endian (Motorola) style. Using Big Endian, D31..D24 will be transmitted first, D23..D16 will be transmitted second, D15..D8 will be transmitted third, and D7..D0 will be transmitted fourth.

#### 1.2.1.2 Tx Raw Data Mode. BCR D17

No data encoding is made. The 8 bit word is loaded into the shift register and shifted once each clock cycle. After 8 bits are shifted out, another 8 bit word is loaded into the shift register. This mode is useful for raw data manipulation and is not intended for normal operation. This mode of operation has not been tested as of this time.

**1.2.1.3 Tx Pre-Drive Level Select. BCR D18**

When Pre-Drive is enabled thru BCR D19, then this bit will select the Pre-Drive Level. When enabled and this bit being a '1' then the Diphase Data bit will be driven to a logic 'high' for the Pre-Drive cycle. When enabled and this bit being a '0' then the Diphase Data bit will be driven to a logic 'low' for the Pre-Drive cycle. Default to '0'.

**1.2.1.4 Tx Pre-Drive Enable. BCR D19**

This bit controls if there is a Pre-Drive inserted before the cable data sync. When this bit is a '1' then the Pre-Drive Level will be inserted into the Diphase Data before the Diphase Sync. When this bit is a '0' then no Pre-Drive Level will be inserted before the Diphase Sync. Default to '0'.

**1.2.1.5 Tx Data Port Output Enable D7-D0. BCR D12**

This bit controls the output enable for the Data Port Bits D7-D0. A '1' will enable the board to Drive Cable Data D7-D0. A '0' will prevent the board from driving Cable Data D7-D0. Default to '0'.

**1.2.1.6 Tx Data Port Output Enable D15-D8. BCR D13**

This bit controls the output enable for the Data Port Bits D15-D8. A '1' will enable the board to Drive Cable Data D15-D8. A '0' will prevent the board from driving Cable Data D15-D8. Default to '0'.

**1.2.1.7 Tx Data Port Output Enable D23-D16. BCR D14**

This bit controls the output enable for the Data Port Bits D23-D16. A '1' will enable the board to Drive Cable Data D23-D16. A '0' will prevent the board from driving Cable Data D23-D16. Default to '0'.

**1.2.1.8 Tx Data Port Output Enable D31-D24. BCR D15**

This bit controls the output enable for the Data Port Bits D31-D24. A '1' will enable the board to Drive Cable Data D31-D24. A '0' will prevent the board from driving Cable Data D31-D24. Default to '0'.

**1.2.1.9 Tx Row Valid Length Counter**

The Transmit Row Valid Length Counter is the Number of D8 bytes that will be transmitted in a single message. If this counter not used, then the Transmitter will transmit data until the FIFO becomes empty. This Counter MUST be used for any message that is not an even multiple of D32 Fifo Size bytes long.

If the FIFO becomes empty, then the transmitter will end the message before the counter has decremented to zero.

**1.2.1.10 Tx Row Invalid Counter**

The Row Invalid Counter will control the length of the Transmit Pre-Drive, if this option is enabled by the system software. This counter is decremented once each clock cycle, 32 Mhz, until it reaches zero, then the Transmitter will begin sending the Sync, followed by the data bits. Reset to 104 which will give a Pre-Drive of 3.25uSec. If there is no data in the FIFO, The Transmitter will wait for data and the Pre-Drive will be extended until there is data in the FIFO. The Pre-Drive Countdown will begin after the Transmitter has been enabled to drive the cable AND the Transmitter has been started. Default is 0x0068, decimal 104.

**1.2.1.11 Tx Status Block Length Counter**

The Transmit Status Block Length Counter has been replaced by a counter that will count the actual number of bytes transmitted. If the Transmit FIFO becomes empty during a Transmit operation, the Transmitter will STOP. This counter can be used to detect if the transmit DMA operation was interrupted by other PCI bus traffic.

**1.2.1.12 Tx Clock Division Counter**

This Counter has been removed from the PCI-HPDI32A-DIPHASE2.

**1.2.1.13 Tx Output Data Port Register**

The Firmware Features Register has been removed from the PCI-HPDI32A-DIPHASE2. It has been replaced by a 32 bit register that can is under software control. When Enabled by the proper output enable

bits in the Board Control Register, the contents of this register will be driven out onto the cable on the Cable Data Bits on a 1 to 1 basis, D0 to D0 thru D31 to D31. This register can be read or written. When you read back this register, you get the contents of the holding register, NOT the contents of the Data Cable.

In order to access this register using the standard device driver you must include a define similar to what is shown below and access the register using this define in the routines as shown.

```
#define SPL_TX_OUTPUT_REG      5

// Write the Test Data to the Output Port Register
WriteLocal(TxBoard, SPL_TX_OUTPUT_REG, TestData) ;

// Read Back the port register and verify it's contents
ReadData = ReadLocal(TxBoard, SPL_TX_OUTPUT_REG) ;
```

### 1.2.2 Receiver Operation:

The Receiver is very flexible with mode bits to control the following Software Selectable options:

- Big Endian / Little Endian
- Software Selectable Data Header Size, 10 Byte or 14 Byte.
- Software Selectable use or do not use the message size encoded in the message.
- Software Selectable Auto Receive reply.
- Number of D8 Bytes Received in this Message Counter .
- Register for the Cable D31-D0 Lines.

When the Receiver is started it will begin searching for the Diphase Data Sync. When the Data Sync is detected then the Receiver will begin collecting the serial data, assembling the data into bytes, assembling the bytes into D32 words, and writing the D32 words into the Receive FIFO. The serial data is received and assembled into bytes with the MSB first, bit 7, the first serial bit received. The second byte received from the cable is assumed to have the message size encoded into it in bits 7 thru 5 as shown in the following table. Message Size 10 byte header or Message Size 14 byte header is software selectable.

**Table 1.2-1 Message Size Encoding**

D7,D6,D5	Message Size 10 Byte	Message Size 14 byte
000	10 - D8 Bytes	14 - D8 Bytes
001	32 - D8 Bytes	36 - D8 Bytes
010	74 - D8 Bytes	78 - D8 Bytes
011	138 - D8 Bytes	142 - D8 Bytes
100	522 - D8 Bytes	526 - D8 Bytes
101	1034 - D8 Bytes	1038 - D8 Bytes
110	4106 - D8 Bytes	4110 - D8 Bytes
111	8202 - D8 Bytes	8206 - D8 Bytes

The Receiver will receive up to the Message Size number of bytes and then stop receiving, or the Receiver can under software control receive all bytes until the Diphase data line goes Idle. When the message has been received the receiver will assert Receive Done as a signal to the Host processor that this message is complete.

The Receiver will generate the following status in the Board Status Register that can be read by the Host Processor and can be used to generate Interrupts thru the Interrupt control register.

- Rx Receive Running – The Receiver is enabled and is not yet done.
- Rx Receive Busy – The Receiver is in the process of receiving a message.
- Rx Receive Done – The receive of a message is complete.

A special feature of the DIPHASE2 receiver is the ability to automatically receive a response to a message that has been transmitted. If the Transmitter and the Receiver are started at the same time, then the Receiver will be held in IDLE and the Transmitter will send its message over the Diphase Data Line. When Transmit Done occurs, the board will automatically Tri-State 'hi-z' the Diphase data line and the Receiver will be released from IDLE. The Receiver will then begin searching for a response to the message that was transmitted. It will still be possible to send self test loop back messages thru the use of the Rx Loop back Enable bit, BCR bit 7.

#### 1.2.2.1 Rx Fifo Big Endian Mode – BCR D24

Normally, the PCI-HPDI32A-DIPHASE2 will receive cable bytes and assemble them into a D32 FIFO word, with D7..D0 being received first, D15..D8 being received second, D23..D16 being received third and D31..D24 being received fourth. When this bit is set then data will be written into the FIFO in Big Endian (Motorola) style. Using Big Endian, D31..D24 being received first, D23..D16 being received second, D15..D8 being received third and D7..D0 being received fourth. Default is '0'.

#### 1.2.2.2 Rx Message Size Select - BCR D25

When this bit is set to a '1', the Receiver will expect the message to carry a 14 byte header. When this bit is set to a '0' the Receiver will expect the message to carry a 10 byte header. Default is '0'.

#### 1.2.2.3 Rx Receive All Data - BCR D26

When this bit is set to a '1', the Receiver will receive all data until from the sync until the Diphase Data Line goes Idle. When this bit is set to a '0' the Receiver will stop receiving data after the message size number of bytes have been received. Default is '0'.

#### 1.2.2.4 Rx Loop Back Enable - BCR D7

When this bit is set to a '1', the Receiver will be active while the Transmitter is active and the Receiver will receive the message that this board is transmitting. When this bit is set to a '0' the Receiver will be held IDLE until after Transmit Done, then will become active to automatically receive a response to the message that was transmitted. Default is '0'.

#### 1.2.2.5 Rx Row Length Counter

This counter will count the number of D8 bytes received in the current message. It will be reset to zero on a board reset, or every time the Receiver Starts up. When in the middle of a message, the Row Length Counter will have the Current count of the number of D8 bytes received in this message..

#### 1.2.2.6 Rx Status Block Length Counter

The Status Block Length counter has been removed from the DIPHASE2 board. It has been replaced by a 32 bit register that will allow the Host processor to read the current state of the 32 Cable Data lines. The RS-485 Receivers used on the HPDI32A-DIPHASE2 project are guaranteed that a Floating Cable will be received as a logic '1'.

#### 1.2.2.7 Rx Disable Output Status Lines - BCR D6

In normal operation, the board will drive Cable Command D5 and Cable Command D6 all of the time. Cable Command D5 will be driven to a logic 'high' when this board is driving the Diphase Data Line. Cable Command D6 will be driven to a logic 'high' when the receiver on this board is running. When this bit is set to a '1', this board will be prevented from driving Cable Command D5 and Cable Command D6. When this bit is set to a '0', then this board will drive Cable Command D5 and D6. This bit is included for testing 2 boards cabled back to back. Default is '0'.

### 1.3 CABLE INTERFACE

The cable interface consists of 32 bi-directional Cable Data lines, the bi-directional Dipphase Data Line, and 2 control / Status output lines, Transmitter Driving the Cable and Receiver Running, which can employ either differential I/O RS485/422 compatibility or differential pseudo ECL. Refer to cable pin-out in Figure 4.3-1.

The Cable Command Lines are assigned as follows.

- Dipphase Data Line – Connected to Cable Command 1.
- Transmitter Driving Dipphase Data Line - Connected to Cable Command 5.
- Receiver Running – Connected to Cable Command 6.
- 32 I/O Lines - Connected to Cable Data 0 thru Cable Data 31.

### 1.4 FIFOs

The FIFOs on the PCI-HPDI32A-DIPHASE2 are used for buffering the transmit and receive data. There are a total of eight FIFOs on the board; 1 set of 4 FIFOs for transmit, and 1 set of 4 FIFOs for receive. Each set consists of 32 bits of data and 4 status flags. The typical configuration is the -256K board ordering option, which is 128Kbytes of Transmit FIFO and 128Kbytes of Receive FIFO. Both FIFOs are organized as 32K deep by 32 bits wide. The receive FIFOs are loaded by the cable receive control logic and read by either the CPU or the DMA. The transmit FIFOs are loaded by either the CPU or the DMA and read by the cable transmit control logic. The 4 status flags that accompany the FIFOs are all active low ('0' being TRUE) and are as follows: Empty, Almost Empty, Almost Full, Full. The Almost Empty and the Almost Full status flags can be programmed by the software to become true at most desired levels.

In addition there are 3 FIFO Event Flags that are part of the FIFO system.

- Rx FIFO Underflow – Read of the Rx FIFO while the FIFO is Empty.
- Rx FIFO Overflow – Write to the Rx FIFO while the Rx FIFO is Full.
- Tx FIFO Overflow – Write to the Tx FIFO while the Tx FIFO is Full.

The Rx FIFO events are cleared by a Rx FIFO Reset. The Tx FIFO Event flag is cleared by a Tx FIFO reset.



---

## 2 REGISTERS

---

### 2.1 PCI-HPDI32A- DIPHASE2 REGISTER INFORMATION

The PCI-HPDI32A-DIPHASE2 Card complies with the **plug-n-play** concept. That is, at the time of power-up, an attempt will be made by the CPU to set up the board to meet the configuration requirements of the system. In doing this, the CPU will map the amount of I/O space requested by the PCI-HPDI32A-DIPHASE2 Card and return the configuration base address into the PCI Configuration Register (Offset 0x18) of this Board. All PCI bus interface functions are handled by the PLX9080-3 PCI bus interface. For more information regarding the PCI bus interface, please refer to **Appendix A, PLX Register Programming**.

## 2.2 PCI-HPDI32A-DIPHASE2 LOCAL REGISTERS, BIT DESCRIPTIONS

The PCI-HPDI32A-DIPHASE2 board contains the following registers.

**Figure 2.2-1 Register Map**

Offset Address	Size	Access*	Register Name	Value after Reset
0x00	D32	RO	Firmware Revision	0x000E0200
0x04	D32	RW	Board Control	0x00000000
0x08	D32	RO	Board Status	0x0000CCXX
0x0C	D32	RW	Tx Programmable Almost	0x01000008
0x10	D32	RW	Rx Programmable Almost	0x01000008
0x14	N/A	N/A	Firmware Features	Removed
0x14	D32	RW	Tx Output Data Port Register	0x00000000
0x18	D32	RW	Tx/Rx FIFOs	EMPTY
0x1C	D32	RO	Tx Status Block Length Counter	0x00000000
0x20	D32	RW	Tx Row Valid Length Counter	0x00000000
0x24	D16	RW	Tx Line Invalid Length	0X00000068
0x28	D32	RO	Rx Status Block Length	Undefined
0x2C	D32	RO	Rx Row Length Counter	0x00000000
0x30	D32	RW	Interrupt Control	0x00000000
0x34	D32	RC	Interrupt Status	0x0000031X
0x38	D32	N/A	Tx Clock Divider - Removed	Removed
0x2C	N/A	N/A	Reserved	N/A
0x40	D32	RO	Tx FIFO Size	0x00000000
0x44	D32	RO	Rx FIFO Size - Removed	Removed
0x48	D32	RO	Tx FIFO Word Count	0x00000000
0x4c	D32	RO	Rx FIFO Word Count - Removed	Removed
0x50	D32	R/W	Interrupt Edge Select	0x00000000
0x54	D32	R/W	Interrupt Hi/Lo Level Select	0xFFFFFFFF

- \* RO - read only
- RW - read/write capability
- RC - read clear (a write clears the specified bits)

### 2.2.1 FIRMWARE REVISION: (Offset 0x00 - RO)

This Register is used to determine the version of firmware that is programmed into the board. If the logic is changed to accommodate a modification for any reason then the value in this register is incremented.

Revision 0x000E0200: Original version after debug and final release.

<b>D7..D0</b>	Firmware Revision
<b>D15..D8</b>	Board Revision Currently HPDI32A Revision A
<b>D23..D16</b>	Board Identifier 0x000E Identifies the HPDI32A-DIPHASE2
<b>D30..D24</b>	Reserved
<b>D31</b>	Firmware Features Identifier. Indicates that this board does not contains the Firmware Features Register.

### 2.2.2 BOARD CONTROL: (Offset 0x04 - RW)

The Board Control Register is strictly under software control and provides the following functionality's:

- a) Resets the board;
- b) Resets the transmit FIFOs;
- c) Resets the receive FIFOs;
- d) Enables the receive logic;
- e) Enables the transmit logic;
- f) Starts the transmission of data;
- g) Enables the cable to start the transmission of data.
- h) Mode control for the transmitter.
- i) Mode control for the receiver.

<b>D0</b>	Board Reset Writing a 1 to this bit will generate a self-timed pulse that is used to reset the on-board logic and the FIFOs. There is no need for the software to clear this bit, the bit will clear itself.
<b>D1</b>	Tx FIFO Reset Writing a 1 to this bit will generate a self-timed pulse that will be used to reset the Tx FIFOs. After setting this bit, there should be a minimum of 25 Rx clk periods, or 1 millisecond if the receivers are turned off, before any local accesses are performed. There is no need for the software to clear this bit, the bit will clear itself.
<b>D2</b>	Rx FIFO Reset Writing a 1 to this bit will generate a self-timed pulse that will be used to reset the Rx FIFOs. After setting this bit, there should be a minimum of 25 Rx clk periods, or 1 millisecond if the receivers are turned off, before any local accesses are performed. There is no need for the software to clear this bit, the bit will clear itself.
<b>D3</b>	Reserved
<b>D4</b>	Enable Tx Writing a 1 to this bit will enable the transmit logic to drive the cable. Writing a 0 to this bit will disable the transmit logic from driving the cable.
<b>D5</b>	Start Rx Writing a 1 to this bit will start the receiver. Writing a 0 to this bit will stop the receiver.
<b>D6</b>	Rx Disable Output Status Lines When this bit is set to a '0', then this board will drive Cable Command D5 and D6 When this bit is set to a '1', this board will be prevented from driving Cable Command D5 and D6 Default is '0'.

<b>D7</b>	Rx Loop Back Enable When set to a '0', then this board will NOT receive the message that this board transmits. When set to a '1', then this board WILL receive the message that this board transmits. Default is '0'
<b>D8</b>	Start Tx Writing a 1 to this bit will start the transmitter. Writing a 0 to this bit will stop the transmitter.
<b>D11..D9</b>	Reserved
<b>D12</b>	Tx Data Port Output Enable D7-D0 A '1' will enable the board to Drive Cable Data D7-D0. A '0' will prevent the board from driving Cable Data D7-D0. Default to '0'.
<b>D13</b>	Tx Data Port Output Enable D15-D8 A '1' will enable the board to Drive Cable Data D15-D8. A '0' will prevent the board from driving Cable Data D15-D8. Default to '0'.
<b>D14</b>	Tx Data Port Output Enable D23-D16 A '1' will enable the board to Drive Cable Data D23-D16. A '0' will prevent the board from driving Cable Data D23-D16. Default to '0'.
<b>D15</b>	Tx Data Port Output Enable D31-D24 A '1' will enable the board to Drive Cable Data D31-D24. A '0' will prevent the board from driving Cable Data D31-D24. Default to '0'.
<b>D16</b>	Tx Big Endian Mode. Writing a 1 to this bit will enable transmitting in Big Endian Mode Writing a 0 to this bit will enable transmitting in Little Endian Mode. Default is '0', Little Endian Mode.
<b>D17</b>	Tx Raw Data Mode Writing a '1' to this bit will enable Transmit Raw Data Mode No data encoding is made. The 8 bit word is loaded into the shift register and shifted once each clock cycle. After 8 bits are shifted out, another 8 bit byte is loaded into the shift register. This mode is useful for raw data manipulation and is not intended for normal operation. This mode of operation has not been tested as of this time.
<b>D18</b>	Tx Pre-Drive Level Select If Pre-Drive is Enabled by Bit D19, then Writing a '1' to this bit will Pre-Drive the Cable to a logic 'high'. Writing a '0' to this bit will Pre-Drive the Cable to a logic 'low'. Default is '0'.
<b>D19</b>	Tx Pre-Drive Enable Writing a '1' to this bit will enable Pre-Driving the cable before the actual Data Transmission begins. Writing a '0' to this bit will inhibit Pre-Driving the Cable before the Data Transmission. Default is '0'.
<b>D23..D20</b>	Reserved
<b>D24</b>	Rx Big Endian Mode Writing a 1 to this bit will enable receiving data in Big Endian Mode Writing a 0 to this bit will enable receiving data in Little Endian Mode. Default is Little Endian Mode.
<b>D25</b>	Rx Message Size Select When this bit is set to a '1' the Receiver will expect the message to carry a 14 byte header. When this bit is set to a '0' the Receiver will expect the message to carry a 10 byte header. Default is '0'.

<b>D26</b>	Rx Receive All Data When this bit is a 1, the receiver will receive all data until the Diphase Data becomes idle. When this bit is a 0, the receiver will receive up to the encoded message size number of bytes and then stop Default is '0'.
<b>D31..D27</b>	Reserved

### 2.2.3 BOARD STATUS: (Offset 0x08 - RO)

The Board Status Register is used to return information to the software about the most current status of the board, at the time of the reading. Listed below is the information that this register contains:

<b>D0</b>	Rx Cable Command D0
<b>D1</b>	Rx Cable Command D1.
<b>D2</b>	Rx Cable Command D2
<b>D3</b>	Rx Cable Command D3.
<b>D4</b>	Rx Cable Command D4.
<b>D5</b>	Rx Cable Command D5.
<b>D6</b>	Rx Cable Command D6.
<b>D7</b>	Tx Transmit Done. A 1 will indicate that a transmit is complete and the transmitter is waiting for the host processor to remove 'Start Tx'. A 0 will indicate that the transmitter is IDLE, or a transmit is in progress.
<b>D8</b>	Tx FIFO Empty L <i>1</i> will indicate the Tx FIFO is not empty. <i>0</i> will indicate the Tx FIFO is empty
<b>D9</b>	Tx FIFO Almost Empty L <i>1</i> will indicate the Tx FIFO is not almost empty. <i>0</i> will indicate the Tx FIFO is empty
<b>D10</b>	Tx FIFO Almost Full L <i>1</i> will indicate the Tx FIFO is not almost full. <i>0</i> will indicate the Tx FIFO is almost full
<b>D11</b>	Tx FIFO Full L <i>1</i> will indicate the Tx FIFO is not full. <i>0</i> will indicate the Tx FIFO is full.
<b>D12</b>	Rx FIFO Empty L <i>1</i> will indicate the Rx FIFO is not empty. <i>0</i> will indicate the Rx FIFO is empty.
<b>D13</b>	Rx FIFO Almost Empty L <i>1</i> will indicate the Rx FIFO is not almost empty. <i>0</i> will indicate the Rx FIFO is empty.
<b>D14</b>	Rx FIFO Almost Full L <i>1</i> will indicate the Rx FIFO is not almost full. <i>0</i> will indicate the Rx FIFO is almost full.
<b>D15</b>	Rx FIFO Full L <i>1</i> will indicate the Rx FIFO is not full. <i>0</i> will indicate the Rx FIFO is full.
<b>D16</b>	Board Jumper 0 1 will indicate that Board Jumper 0 is present 0 will indicate that board jumper 0 is absent
<b>D17</b>	Board Jumper 1 1 will indicate that Board Jumper 1 is present 0 will indicate that board jumper 1 is absent

<b>D18</b>	Reserved
<b>D19</b>	Rx FIFO Underflow. 1 will indicate that a Receive FIFO underflow has occurred 0 will indicate that no Receive FIFO underflow has occurred
<b>D20</b>	Rx FIFO Overflow. 1 will indicate that a Receive FIFO overflow has occurred 0 will indicate that no Receive FIFO overflow has occurred
<b>D21</b>	Tx FIFO Overflow. 1 will indicate that a Transmit FIFO overflow has occurred 0 will indicate that no Transmit FIFO overflow has occurred
<b>D22</b>	Rx Receive Done 1 will indicate that the Receiver has completed 1 full Message and is waiting for the host processor to remove 'Start Rx'. 0 will indicate that the Receiver is IDLE, or is busy receiving data.
<b>D23</b>	Rx Receive Busy. 1 will indicate that the Receiver is currently receiving a message. 0 will indicate that the Receiver is not currently receiving a message.
<b>D24</b>	Rx Receive Running 1 will indicate that the Receiver has been started and is not Receive Done. 0 will indicate that the Receiver has not been started, or has been started but is now done.
<b>D25</b>	Tx Transmit Busy 1 will indicate the Transmitter is Transmitting data. 0 will indicate the Transmitter is either sending the Pre-Drive, or is IDLE or Done.
<b>D26</b>	Tx Transmitter Driving the Cable 1 will indicate the Transmitter is Driving the cable and is not Transmit Done. 0 will indicate the Transmitter is not Driving the Cable
<b>D31..D27</b>	Reserved

#### 2.2.4 TX ALMOST: (Offset 0x0C - RW)

This register contains the values that are used to program the Almost Flags of the transmit FIFOs.

Default is 0x01000008.

Almost Empty = 0x08

Almost Full = 0x0100 from full.

<b>D7..D0</b>	Low byte, Almost Empty
<b>D15..D8</b>	High byte, Almost Empty
<b>D23..D16</b>	Low byte, Almost Full
<b>D31..D24</b>	High byte, Almost Full

#### 2.2.5 RX ALMOST: (Offset 0x10 - RW)

This register contains the values that are used to program the Almost Flags of the receive FIFOs.

Default is 0x01000008.

Almost Empty = 0x08

Almost Full = 0x0100 from full.

<b>D7..D0</b>	Low byte, Almost Empty
<b>D15..D8</b>	High byte, Almost Empty
<b>D23..D16</b>	Low byte, Almost Full
<b>D31..D24</b>	High byte, Almost Full

**2.2.6 FIRMWARE FEATURES: (Offset 0x14 - RW)**

Register to note new features present in FW. This Register has been removed from the DIPHASE2.

**2.2.7 TX OUTPUT DATA PORT REGISTER: (Offset 0x14 - RW)**

This is the holding register for data to be output on the 32 programmable I/O Lines. The outputs are enabled to drive the cable thru the Board Control Register Bits 15 thru 12. Reading this address will give the contents of the holding register, NOT the contents of the Cable Data Lines. To read the current state of the Cable Data Lines read the Rx Status Block Length register.

<b>D31..D0</b>	Tx Output Data Port Register.
----------------	-------------------------------

**2.2.8 FIFO I/O: (Offset 0x18 - RW)**

The 64/128/256/512 Kbytes FIFOs are used for buffering data. This gives the software a means of buffering the data before it is transmitted to the cable or retrieved from the FIFOs. The FIFOs are also used by the DMA for the same purpose. This eliminates unnecessary PCI bus arbitration which provides for faster and more efficient bus cycles for transfers; hence, a faster and more efficient system. Typical configuration is 256Kbytes which is 128Kbytes Transmit FIFO (32K by 32 bits) and 128Kbytes Receive FIFO (32 K by 32 bits) .

<b>D31..D0</b>	FIFO Data 0..31 This is the buffer that contains both the transmit and receive data. A write to this offset will load into the Transmit FIFO. A read from this offset will read from the Receive FIFO.
----------------	---

**2.2.9 TX STATUS LENGTH REGISTER: (Offset 0x1C – RO)**

The Transmit Status Length Register has been replaced by a counter that will count the number of bytes that were transmitted in the current message. Reset to Zero when the transmitter starts. Useful for checking if the DMA was interrupted by bus loading.

<b>D31..D0</b>	Data 0..31 The number of Bytes transmitted in the current message. Reset to Zero when the Transmitter Starts. .
----------------	--

**2.2.10 TX ROW VALID LENGTH REGISTER: (Offset 0x20 - RW)**

This register / counter controls the number of D16 words per Line of data. For each D16 word that is transmitted, the counter will be decremented. When the Counter reaches Zero, the transmitter will insert a GAP between lines.

<b>D31..D0</b>	Data 0..31 The number of D8 Bytes to transmit per message.
----------------	---

**2.2.11 TX ROW INVALID LENGTH REGISTER: (Offset 0x24 - RW)**

This register / counter is used to control the Pre-Drive inserted before the Sync when a message is transmitted. Not used if Pre-Drive is not enabled. Reset to 104 which will give a Pre-Drive of 3.250 uSec. Set in Increments of 31.250nSec. 16 Bit counter.

<b>D15..D0</b>	Data 0..15 Sets the size of the Gap between lines within a Frame of Data. .
<b>D31..D16</b>	Not Used.

**2.2.12 RX STATUS BLOCK LENGTH COUNTER: (Offset 0x28 - RO)**

The Rx Status Block Length Counter has been replaced by a Read Only Register that will read the current state of the 32 Cable Data Lines, Cable Data D31-D0. Note that the RS-485 receivers used on the HPDI32A-DIPHASE2 are guaranteed to receive a logic '1' if the RS-485 input is floating.

<b>D31..D0</b>	Cable Data D31 thru Cable Data D0
----------------	-----------------------------------

**2.2.13 RX ROW LENGTH COUNTER: (Offset 0x2C - RO)**

This counter contains the number of D8 bytes received in this message. It is reset to 0 when the receiver starts running. The Host Processor can read it at any time.

<b>D31..D0</b>	Counter 0..31 The number of D8 bytes received in this message. It is reset to 0 when the Receiver is started.
----------------	---



### 2.2.14 INTERRUPT CONTROL: (Offset 0x30 - RW)

The Interrupt Control Register provides the software with a means of selecting what conditions will be allowed to generate an interrupt.

<b>D0</b>	Enable Interrupt on Cable Diphase Data Rising Edge. <i>I</i> will allow an interrupt on the Cable Diphase Data Lines rising edge. <i>O</i> will disallow an interrupt on the Cable Diphase Data Lines rising edge.
<b>D1</b>	Enable Interrupt on Cable Diphase Data Falling Edge. <i>I</i> will allow an interrupt on Cable Diphase Data Falling Edge. <i>O</i> will disallow an interrupt on Cable Diphase Data Falling Edge.
<b>D2</b>	Enable Interrupt on Cable Diphase Data a logic '1'. <i>I</i> will allow an interrupt on Cable Diphase Data a logic '1'. <i>O</i> will disallow an interrupt Cable Diphase Data a logic '1'.
<b>D3</b>	Enable Interrupt on Cable Command 2 a logic '1'. <i>I</i> will allow an interrupt on Cable Command 2 a logic '1' <i>O</i> will disallow an interrupt Cable Command 2 a logic '1'
<b>D4</b>	Enable Interrupt on Cable Command 4 a logic '1'. <i>I</i> will allow an interrupt on Cable Command 4 a logic '1' <i>O</i> will disallow an interrupt Cable Command 4 a logic '1'
<b>D5</b>	Enable Interrupt on Cable Command 5 a logic '1'. – Cable Transmitter Driving the Cable <i>I</i> will allow an interrupt on Cable Command 5 a logic '1' <i>O</i> will disallow an interrupt Cable Command 5 a logic '1'
<b>D6</b>	Enable Interrupt on Cable Command 6 a logic '1'. – Cable Receiver Ready to Receive <i>I</i> will allow an interrupt on Cable Command 6 a logic '1' <i>O</i> will disallow an interrupt Cable Command 6 a logic '1'
<b>D7</b>	Enable Interrupt on Tx Transmit Done. <i>I</i> will allow an interrupt on Tx Transmit Done. <i>O</i> will disallow an interrupt on Tx Transmit Done.
<b>D8</b>	Enable Interrupt on Tx FIFO Empty <i>I</i> will allow an interrupt on Tx FIFO empty. <i>O</i> will disallow an interrupt on Tx FIFO empty.
<b>D9</b>	Enable Interrupt on Tx FIFO Almost Empty <i>I</i> will allow an interrupt on Tx FIFO almost empty. <i>O</i> will disallow an interrupt on Tx FIFO almost empty.
<b>D10</b>	Enable Interrupt on Tx FIFO Almost Full <i>I</i> will allow an interrupt on Tx FIFO almost full. <i>O</i> will disallow an interrupt on Tx FIFO almost full.
<b>D11</b>	Enable Interrupt on Tx FIFO Full <i>I</i> will allow an interrupt on Tx FIFO full. <i>O</i> will disallow an interrupt on Tx FIFO full.
<b>D12</b>	Enable Interrupt on Rx FIFO NOT Empty <i>I</i> will allow an interrupt on Rx FIFO NOT empty. <i>O</i> will disallow an interrupt on Rx FIFO NOT empty.
<b>D13</b>	Enable Interrupt on Rx FIFO NOT Almost Empty <i>I</i> will allow an interrupt on Rx FIFO NOT Almost empty. <i>O</i> will disallow an interrupt on Rx FIFO NOT Almost empty.
<b>D14</b>	Enable Interrupt on Rx FIFO Almost Full <i>I</i> will allow an interrupt on Rx FIFO almost full. <i>O</i> will disallow an interrupt on Rx FIFO almost full.

<b>D15</b>	Enable Interrupt on Rx FIFO Full <i>I</i> will allow an interrupt on Rx FIFO full. <i>O</i> will disallow an interrupt on Rx FIFO full.
<b>D16</b>	Enable Interrupt on Rx FIFO Underflow. <i>I</i> will allow an interrupt on Rx FIFO Underflow. <i>O</i> will disallow an interrupt on Rx FIFO Underflow.
<b>D17</b>	Enable Interrupt on Rx FIFO Overflow. <i>I</i> will allow an interrupt on Rx FIFO Overflow. <i>O</i> will disallow an interrupt on Rx FIFO Overflow.
<b>D18</b>	Enable Interrupt on Tx FIFO Overflow. <i>I</i> will allow an interrupt on Tx FIFO Overflow. <i>O</i> will disallow an interrupt on Tx FIFO Overflow.
<b>D19</b>	Enable Interrupt on Rx Receive Done. <i>I</i> will allow an interrupt on Rx Receive Done. <i>O</i> will disallow an interrupt on Rx Receive Done.
<b>D20</b>	Enable Interrupt on Rx Receive Busy. <i>I</i> will allow an interrupt on Rx Receive Busy. <i>O</i> will disallow an interrupt on Rx Receive Busy.
<b>D21</b>	Enable Interrupt on Rx Receive Running. <i>I</i> will allow an interrupt on Rx Receive Running. <i>O</i> will disallow an interrupt on Rx Receive Running.
<b>D22</b>	Enable Interrupt on Tx Transmitter Busy. <i>I</i> will allow an interrupt on Tx Transmitter Busy. <i>O</i> will disallow an interrupt on Tx Transmitter Busy.
<b>D23</b>	Enable Interrupt on Tx Transmitter Driving the Cable. <i>I</i> will allow an interrupt on Tx Transmitter Driving the Cable. <i>O</i> will disallow an interrupt on Tx Transmitter Driving the Cable.
<b>D31..24</b>	Reserved

## 2.2.15 INTERRUPT STATUS: (Offset 0x34 - RC)

The Interrupt Status Register serves as a dual-purpose register. Each bit in this register operates independently of each other. If an interrupt condition is enabled, in the Interrupt Control Register, the appropriate bit, in the Interrupt Status Register, will indicate if an interrupt has occurred or not, and it will continue to indicate this until the software resets that bit. If an interrupt bit is not enabled, in the Interrupt Control Register, then the appropriate bit, in the Interrupt Status Register, will indicate whether or not the condition exists for an interrupt request.

<b>D0</b>	<p>Cable Diphas Data Rising Edge                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on the Cable Diphas Data Rising Edge has occurred.                          a 0 will indicate that an interrupt on the Cable Diphas Data Rising Edge has not occurred.                      If this bit is not enabled as an interrupt                          a 1 will indicate that the Cable Diphas Data is a Rising Edge now.                          a 0 will indicate that the Cable Diphas Data is not a rising edge.</p>
<b>D1</b>	<p>Cable Diphas Data Falling Edge                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on the Cable Diphas Data Falling Edge has occurred.                          a 0 will indicate that an interrupt on the Cable Diphas Data Falling Edge has not occurred.                      If this bit is not enabled as an interrupt                          a 1 will indicate that the Cable Diphas Data is a Falling Edge now.                          a 0 will indicate that the Cable Diphas Data is not a Falling edge.</p>
<b>D2</b>	<p>Cable Diphas Data                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on Diphas Data has occurred.                          a 0 will indicate that an interrupt on Diphas Data has not occurred.                      If this bit is not enabled as an interrupt                          a 1 will indicate that Cable Diphas Data is currently a '1'.                          a 0 will indicate that Cable Diphas Data is not currently a '1'.</p>
<b>D3</b>	<p>Cable Command 2                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on Command 2 has occurred.                          a 0 will indicate that an interrupt on Command 2 has not occurred.                      If this bit is not enabled as an interrupt                          a 1 will indicate that Cable Command 2 is currently a '1'.                          a 0 will indicate that Cable Command 2 is not currently a '1'.</p>
<b>D4</b>	<p>Cable Command 4                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on Cable Command 4 has occurred.                          a 0 will indicate that an interrupt on Cable Command 4 has not occurred.                      If this bit is not enabled as an interrupt                          a 1 will indicate that Cable Command 4 is currently a '1'.                          a 0 will indicate that Cable Command 4 is not currently a '1'.</p>

<b>D5</b>	<p>Cable Command 5 - Cable Transmitter Driving the Cable</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on Cable Command 5 has occurred.</li> <li>a 0 will indicate that an interrupt on Cable Command 5 has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that Cable Command 5 is currently a '1'</li> <li>a 0 will indicate that Cable Command 5 is not currently a '1'.</li> </ul>
<b>D6</b>	<p>Cable Command 6 – Cable Receiver Ready to Receive</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on Cable Command 6 has occurred.</li> <li>a 0 will indicate that an interrupt on Cable Command 6 has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that Cable Command 6 is currently a '1'</li> <li>a 0 will indicate that Cable Command 6 is not currently a '1'.</li> </ul>
<b>D7</b>	<p>Tx Transmit Done</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on Tx Transmit Done has occurred.</li> <li>a 0 will indicate that an interrupt on Tx Transmit Done has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that Tx Transmit Done has occurred.</li> <li>a 0 will indicate that Tx Transmit Done has not occurred.</li> </ul>
<b>D8</b>	<p>Tx FIFO Empty</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on the Tx FIFO empty has occurred.</li> <li>a 0 will indicate that an interrupt on the Tx FIFO empty has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that the Tx FIFO is currently empty</li> <li>a 0 will indicate that the Tx FIFO is not currently empty.</li> </ul>
<b>D9</b>	<p>Tx FIFO Almost Empty</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on the Tx FIFO almost empty has occurred.</li> <li>a 0 will indicate that an interrupt on the Tx FIFO almost empty has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that the Tx FIFO is currently almost empty.</li> <li>a 0 will indicate that the Tx FIFO is not currently almost empty.</li> </ul>
<b>D10</b>	<p>Tx FIFO Almost Full</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on the Tx FIFO almost full has occurred.</li> <li>a 0 will indicate that an interrupt on the Tx FIFO almost full has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that the Tx FIFO is currently almost full.</li> <li>a 0 will indicate that the Tx FIFO is not currently almost full.</li> </ul>
<b>D11</b>	<p>Tx FIFO Full</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on the Tx FIFO full has occurred.</li> <li>a 0 will indicate that an interrupt on the Tx FIFO full has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that the Tx FIFO is currently full.</li> <li>a 0 will indicate that the Tx FIFO is not currently full.</li> </ul>

<b>D12</b>	<p>Rx FIFO Empty</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on the Rx FIFO empty has occurred.</li> <li>a 0 will indicate that an interrupt on the Rx FIFO empty has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that the Rx FIFO is currently empty</li> <li>a 0 will indicate that the Rx FIFO is not currently empty.</li> </ul>
<b>D13</b>	<p>Rx FIFO Almost Empty</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on the Rx FIFO almost empty has occurred.</li> <li>a 0 will indicate that an interrupt on the Rx FIFO almost empty has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that the Rx FIFO is currently almost empty.</li> <li>a 0 will indicate that the Rx FIFO is not currently almost empty.</li> </ul>
<b>D14</b>	<p>Rx FIFO Almost Full</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on the Rx FIFO almost full has occurred.</li> <li>a 0 will indicate that an interrupt on the Rx FIFO almost full has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that the Rx FIFO is currently almost full.</li> <li>a 0 will indicate that the Rx FIFO is not currently almost full.</li> </ul>
<b>D15</b>	<p>Rx FIFO Full</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on the Rx FIFO full has occurred.</li> <li>a 0 will indicate that an interrupt on the Rx FIFO full has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that the Rx FIFO is currently full.</li> <li>a 0 will indicate that the Rx FIFO is not currently full.</li> </ul>
<b>D16</b>	<p>Enable Interrupt on Rx FIFO Underflow.</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on Rx FIFO Underflow has occurred.</li> <li>a 0 will indicate that an interrupt on Rx FIFO Underflow has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that Rx FIFO Underflow has occurred.</li> <li>a 0 will indicate that Rx FIFO Underflow is not occurred.</li> </ul>
<b>D17</b>	<p>Enable Interrupt on Rx FIFO Overflow.</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on Rx FIFO Overflow has occurred.</li> <li>a 0 will indicate that an interrupt on Rx FIFO Overflow has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that Rx FIFO Overflow has occurred.</li> <li>a 0 will indicate that Rx FIFO Overflow has not occurred.</li> </ul>
<b>D18</b>	<p>Enable Interrupt on Tx FIFO Overflow.</p> <p>If this bit is enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that an interrupt on Tx FIFO Overflow has occurred.</li> <li>a 0 will indicate that an interrupt on Tx FIFO Overflow has not occurred.</li> </ul> <p>If this bit is not enabled as an interrupt</p> <ul style="list-style-type: none"> <li>a 1 will indicate that a Tx FIFO Overflow has occurred.</li> <li>a 0 will indicate that a Tx FIFO Overflow has not occurred.</li> </ul>

<b>D19</b>	<p>Enable Interrupt on Rx Receive Done.                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt Rx Receive Done has occurred.                          a 0 will indicate that an interrupt Rx Receive Done has not occurred.</p> <p>If this bit is not enabled as an interrupt                          a 1 will indicate that a Rx Receive Done has occurred.                          a 0 will indicate that a Rx Receive Done has not occurred.</p>
<b>D20</b>	<p>Enable Interrupt on Rx Receive Busy.                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on Rx Receive Busy has occurred.                          a 0 will indicate that an interrupt on Rx Receive Busy has not occurred.</p> <p>If this bit is not enabled as an interrupt                          a 1 will indicate that Rx Receive Busy has occurred.                          a 0 will indicate that Rx Receive Busy has not occurred</p>
<b>D21</b>	<p>Enable Interrupt on Rx Receive Running.                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on Rx Receive Running has occurred.                          a 0 will indicate that an interrupt on Rx Receive Running has not occurred.</p> <p>If this bit is not enabled as an interrupt                          a 1 will indicate that Rx Receive Running has occurred.                          a 0 will indicate that Rx Receive Running has not occurred</p>
<b>D22</b>	<p>Enable Interrupt on Tx Transmitter Busy.                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on Tx Transmitter Busy has occurred.                          a 0 will indicate that an interrupt on Tx Transmitter Busy has not occurred.</p> <p>If this bit is not enabled as an interrupt                          a 1 will indicate that Tx Transmitter Busy has occurred.                          a 0 will indicate that Tx Transmitter Busy has not occurred</p>
<b>D23</b>	<p>Enable Interrupt on Tx Transmitter Driving the Cable.                      If this bit is enabled as an interrupt                          a 1 will indicate that an interrupt on Tx Transmitter Driving the Cable has occurred.                          a 0 will indicate that an interrupt on Tx Transmitter Driving the Cable has not occurred.</p> <p>If this bit is not enabled as an interrupt                          a 1 will indicate that Tx Transmitter Driving the Cable has occurred.                          a 0 will indicate that Tx Transmitter Driving the Cable has not occurred</p>
<b>D31..21</b>	Reserved

**2.2.16 TX CLOCK DIVIDER: (Offset 0x38 - RW)**

The Transmit Clock Divider has been removed from the PCI-HPDI32A-DIPHASE2.  
 NOTE: This register cannot be accessed using the currently released Windows NT Device Driver.

**2.2.17 RESERVED: (Offset 0x3C - RW)**

This register is reserved for future use.  
 NOTE: This register cannot be accessed using the currently released Windows NT Device Driver.

**2.2.18 TX FIFO SIZE: (Offset 0x40 – RO)**

Upon Power up (or any time FPGA code is reloaded), the logic will fill Tx FIFO and count the number of writes until the FIFO is full. This value is saved as the FIFO size, and the FIFOs are reset. The Counter is only 20 bits wide, the upper 12 bits are undefined.

NOTE: This register cannot be accessed using the currently released Windows NT Device Driver.

<b>D19..D0</b>	Tx FIFO Size. Number of D32 Words that the Tx FIFO Holds.
<b>D31..D20</b>	Undefined

**2.2.19 RX FIFO SIZE: (Offset 0x44 – RO)**

This Feature has been removed from the HPDI32A-DIPHASE2.

NOTE: This register cannot be accessed using the currently released Windows NT Device Driver.

<b>D19..D0</b>	Reserved.
<b>D31..D20</b>	Undefined

**2.2.20 TX FIFO WORD COUNT: (Offset 0x48 – RO)**

This counter tracks the number of D32 Words in the Tx FIFO. Every time a word is written to the FIFO, the count is incremented. When a D32 Word is read, the count is decremented. The Counter is only 20 bits wide, the upper 12 bits are undefined.

NOTE: This register cannot be accessed using the currently released Windows NT Device Driver.

<b>D19..D0</b>	Tx FIFO Word Count. Number of D32 Words currently in the Tx FIFO.
<b>D31..D20</b>	Undefined

**2.2.21 RX FIFO WORD COUNT: (Offset 0x4C – RO)**

This Feature has been removed from the HPDI32A-DIPHASE2.

NOTE: This register cannot be accessed using the currently released Windows NT Device Driver.

<b>D19..D0</b>	Reserved.
<b>D31..D20</b>	Undefined

**2.2.22 INTERRUPT EDGE REGISTER: (Offset 0x50 – RW)**

Thirty-two bit register to define interrupt source to individually define as edge or level sensitive. Bits are the same order as the interrupt source. Default is 0, Level Triggered Interrupts to maintain compatibility with existing devices.

NOTE: This register cannot be accessed using the currently released Windows NT Device Driver.

<b>D31..D0</b>	Interrupt Edge Select Individual Interrupt level or edge trigger select. 0 = Level Triggered Interrupts. 1 = Edge Triggered Interrupts.
----------------	--

**2.2.23 INTERRUPT HI/LO REGISTER: (Offset 0x54 – RW)**

Thirty-two bit register to define interrupt source as active hi or active lo. Bits are in same order as interrupt source. Default is 1, Active High Interrupts to maintain compatibility with existing devices.

NOTE: This register cannot be accessed using the currently released Windows NT Device Driver.

<b>D31..D0</b>	Interrupt Hi/Lo Select. Individual Interrupt Hi or Lo active level selection. 0 = Active Low Interrupt. 1 = Active Hi Interrupt
----------------	--



# 3 PROGRAMMING

## 3.1 INITIALIZATION

Initializing the PCI-HPDI32A-DIPHASE2 Card will generally need to be done only once by the software, unless the mode needs to change. The software is responsible for tracking any changes; for making all changes necessary to meet the needs of the application; and for making all the adjustments when requirements change. Most of the configuration status can be determined by reading the Board Control Register. Upon system reset, and also after a board reset is performed, the board will be in a state where the following initialization will apply:

1. all cable data transceivers will be in their default state, Tri-state off
2. all interrupts will be disabled;
3. the FIFOs will be empty;
4. the receive logic will be disabled.
5. the transmit logic will be disabled.
6. the board will be driving Cable Command 5 and Cable Command 6.
7. all RW registers will be set to 0 (Board Control Register bits are set to 0 on system reset)

## 3.2 RESETS

There are three (3) bits on this board that are used as resets to the local side. All three bits are located in the Board Control Register. These bits perform a reset when the software writes a 1 to them. After writing a 1, the software does not need to return to clear the bits, the bits operate as a self-timed pulse that will return to 0 after they have been asserted long enough to perform the reset(s). For further details on the resets refer to the PCI-HPDI32A-DIPHASE2 Register Map, See Table 2.1-1.

<b>D0</b>	<b>Board Reset</b> will reset the local logic, clear the FIFOs, and place the appropriate registers into a known state.
<b>D1</b>	<b>Tx FIFO Reset</b> will reset the Tx FIFOs.
<b>D2</b>	<b>Rx FIFO Reset</b> will reset the Rx FIFOs.

The FIFOs are reset by either a hardware system reset of the board, a software FIFO reset (Board Control Register bit 1 or bit 2), or a software board reset (Board Control Register bit 0).

### 3.3 FIFOs

The FIFOs flags are used to indicate the current fill level of the FIFO. There are four flags for Tx and four flags for Rx. These flags are labeled and defined as follows:

- Empty:
- Almost Empty:
- Almost Full:
- Full:

There are 0 true signals. The almost registers are used for programming the FIFOs, states are:

- Empty – 0
- Almost Empty – programmable level
- Almost Full – programmable level.
- Full – depth

Use bits 1 and 2 of the BCR a board reset will not program the FIFOs.

The FIFO Almost Registers are used to program the Almost Empty and Almost Full flag levels. The default is 0x01000008 which will give you levels of

- Almost Empty – 0x0008 – 8 D32 words above empty.
- Almost Full – 0x0100 – Almost Full is 256 D32 words below Full.

In addition there are 3 FIFO Event Flags that are part of the FIFO system.

- Rx FIFO Underflow – Read of the Rx FIFO while the Rx FIFO is Empty.
- Rx FIFO Overflow – Write to the Rx FIFO while the Rx FIFO is Full.
- Tx FIFO Overflow – Write to the Tx FIFO while the Tx FIFO is Full.

The Rx FIFO events are cleared by a Rx FIFO Reset. The Tx FIFO Event flag is cleared by a Tx FIFO reset.

### 3.4 INTERRUPTS

In order for this board to generate interrupts to the PCI bus, Bits 8, 11, and 16 of the PLX Interrupt Control/Status Register must be set to a 1. These bits must be set to a 1 in order for the interrupts to occur.

The next step in initializing the interrupt, is to specify which interrupts are to be allowed. The board allows the software to enable some interrupts and leave others disabled. This is accomplished by writing a 1 to the appropriate bits in the Interrupt Control Register (ICR). For example, to enable the interrupt for FIFO Almost Empty, the software will need to write a 1 to bit 13 of the ICR. This bit will not need to be changed again until the need to disable this specific interrupt. This enable can be a one time process which will allow many interrupts to occur.

Multiple interrupts from the same cause are prevented via the Interrupt Status Register (ISR). Writing to the ISR is the method by which the software acknowledges to the board that it has received the previous interrupt request and signals to the board that it may now generate any other interrupts that may occur. This register will need some attention from the software after each interrupt has occurred. Following the previous example, when this interrupt has occurred, the software will find that bit 13 of the ISR is now a 1, indicating that a FIFO Almost Empty interrupt has occurred. This bit will remain a 1 and will not allow any additional interrupts to be generated until the software performs a write to this register. To re-enable the FIFO Almost Empty interrupt, the software must write a 1 to bit 13. This will clear the occurrence of the interrupt.

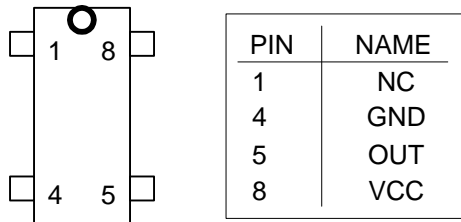
The enabling, latching, and clearing of the FIFO Almost Empty status bit will not effect the other bits of the register. This means that if the software receives the interrupt for FIFO Almost Empty (the only interrupt currently enabled) the software may very well find that the FIFO is now Empty, (indicated by bit 12 being a 1). Since this bit is not enabled as an interrupt, it is acting as a status bit. If it is enabled now, it will immediately generate an interrupt.

# 4 HARDWARE CONFIGURATION

## 4.1 THE ON-BOARD TRANSMIT CLOCK

The on-board oscillator, U11, is used as the transmit clock while in test mode. The oscillator is factory installed at 20MHz. This oscillator can be changed in the field by the end user. The maximum frequency supported by this clock is 32Mhz.

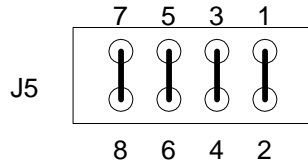
**Figure 4.1-1 Oscillator Pinout**



## 4.2 JUMPERS (J5)

J5 is a header consisting of four individual jumpers.

**Figure 4.2-1 Jumper J5**



**J5: 1 to J5: 2      EEPROM Configuration**

This jumper connects the PNP EEPROM to the PCI chipset for power-up configuration. This jumper is intended for factory use only and should always be installed.

**J5: 3 to J5: 4      FPGA Reload**

This jumper connects the FPGA Reload to the local PCI Reset. When this jumper is installed, the FPGA will reload when the PCI is reset. If the jumper is off, the FPGA will reload only at power-up. This jumper is intended for factory use only and should always be installed.

**J5: 5 to J5: 6      Board Jumper 0**

This jumper is Readable thru the Board Status Register and can be used to identify a particular board if multiple boards are installed in a system.

**J5: 7 to J5: 8      Board Jumper 1**

This jumper is Readable thru the Board Status Register and can be used to identify a particular board if multiple boards are installed in a system.

### 4.3 CABLE CONNECTOR

The 80-pin user connector (reference designator: P1) is manufactured by Robinson Nugent, the part number is P50E-080-P1-SR1-TG. The part number for the mate is P50E-080-S-TG, (50 mil. cabling is suggested for twisted pair), or P25E-080S-TG (25 mil. cabling may be used for multi-drop capability, but with loss of twisted pair).

**Figure 4.3-1 Cable Pinout**

Pin No.	Cable Signal Name	Pin No.	Cable Signal Name
1	CABLE CLOCK + / Not Used	41	CABLE D12 +
2	CABLE CLOCK - / Not Used	42	CABLE D12 -
3	Cable Command 0 + / Not Used	43	CABLE D13 +
4	Cable Command 0 - / Not Used	44	CABLE D13 -
5	Cable Diphase Data +	45	CABLE D14 +
6	Cable Diphase Data -	46	CABLE D14 -
7	Cable Command 2 + / Not Used	47	CABLE D15 +
8	Cable Command 2 - / Not Used	48	CABLE D15 -
9	Cable Command 3 + / Not Used	49	CABLE D16 +
10	Cable Command 3 - / Not Used	50	CABLE D16 -
11	Cable Command 4 + / Not Used	51	CABLE D17 +
12	Cable Command 4 - / Not Used	52	CABLE D17 -
13	Tx Transmitter Driving Diphase Data +	53	CABLE D18 +
14	Tx Transmitter Driving Diphase Data -	54	CABLE D18 -
15	Rx Receiver Running +	55	CABLE D19 +
16	Rx Receiver Running -	56	CABLE D19 -
17	CABLE D0 +	57	CABLE D20 +
18	CABLE D0 -	58	CABLE D20 -
19	CABLE D1 +	59	CABLE D21 +
20	CABLE D1 -	60	CABLE D21 -
21	CABLE D2 +	61	CABLE D22 +
22	CABLE D2 -	62	CABLE D22 -
23	CABLE D3 +	63	CABLE D23 +
24	CABLE D3 -	64	CABLE D23 -
25	CABLE D4 +	65	CABLE D24 +
26	CABLE D4 -	66	CABLE D24 -
27	CABLE D5 +	67	CABLE D25 +
28	CABLE D5 -	68	CABLE D25 -
29	CABLE D6 +	69	CABLE D26 +
30	CABLE D6 -	70	CABLE D26 -
31	CABLE D7 +	71	CABLE D27 +
32	CABLE D7 -	72	CABLE D27 -
33	CABLE D8 +	73	CABLE D28 +
34	CABLE D8 -	74	CABLE D28 -
35	CABLE D9 +	75	CABLE D29 +
36	CABLE D9 -	76	CABLE D29 -
37	CABLE D10 +	77	CABLE D30 +
38	CABLE D10 -	78	CABLE D30 -
39	CABLE D11 +	79	CABLE D31 +
40	CABLE D11 -	80	CABLE D31 -

# Appendix A PLX REGISTER PROGRAMMING

The following Appendix describes the registers of the PLX PCI Bus Interface that is used on the PMC-HPDI32A-CDC.

## A.1 Configuration EEPROM

During configuration, the PLX will initialize itself from a EEPROM that is programmed at the factory. The following values are loaded into the PLX and are used to configure the board for operation.

**Table B Configuration EEPROM Contents.**

PCI Offset from base Addr	Size	Register Name	Value after Reset	PCI CFG Register Address
0x70	D32	Device ID / Vendor ID	0x9080/ 0x10B5	0x00
-	D32	Class Code / Revision ID	0x07800003	0x08
-	D32	Max / Min Latency / Int. Pin / Int. Line Routing Value	0x00010000	0x3C
0x78	D32	Mailbox 0 User defined	0x00000000	-
0x7c	D32	Mailbox 1 User defined	0x00000000	-
0x00	D32	Range PCI to Local Address Space 0	0xFFFFFE00	0x18
0x04	D32	Local Base Address (remap) Local Address Space 0	0x00000001	-
0x08	D32	Local Mode / Arbitration register	0x0000, Not Used	-
0x0C	D32	Big / little Endian descriptor	0x0000	-
0x10	D32	Range for PCI to local expansion ROM	0x0000, Not Used	-
0x14	D32	Local Base Address Remap PCI to Local Expansion ROM	0x0000, Not Used	-
0x18	D32	Bus region descriptors for PCI to Local Address Space 0	0x42000143	-
0x1C	D32	Range for direct Master to PCI	0x0000, Not Used	-
0x20	D32	Local Base Address for direct Master to PCI Memory	0x0000, Not Used	-
0x24	D32	Local Base Address for direct Master to PCI IO/CFG	0x0000, Not Used	-
0x28	D32	Base Address (remap) for direct Master to PCI	0x0000, Not Used	-
0x2C	D32	PCI Configuration Address Register for direct Master to PCI IO/CFG	0x0000, Not Used	-
-	D32	Subsystem ID / Subsystem Vendor ID	0x2400/ 0x10B5	0x2C
0xF0	D32	Range PCI to Local Address Space 1	0x0000, Not Used	-
0xF4	D32	Local Base Address Remap PCI to Local Address Space 1	0x0000, Not Used	-
0xF8	D32	Bus region descriptors for PCI to Local Address Space 1	0x0000, Not Used	-
-	D32	PCI Base Address for Local Expansion ROM	0x0000, Not Used	0x30

## A.2 PCI CONFIGURATION REGISTER BIT MAPS

The PCI-HPDI32-CDC Card complies with the **plug-n-play** concept. That is, at the time of power-up, an attempt will be made by the CPU to set up the board to meet the configuration requirements of the system. In doing this, the CPU will map the amount of I/O space requested by the PCI-HPDI32-CDC Card and return the configuration base address into the PCI Configuration Register (Offset 0x18) of this Board.

The configuration registers are usually programmed by the system BIOS during system startup.

**Table C Pci Configuration Register Mapping**

Size	Access		Register Name	Value after Reset	PCI CFG register address
	R	W			
D32	yes	no	Device ID/Vendor ID	0x9080/ 0x10B5	0x00
D32	yes	yes	Status/Command	0x00010000	0x04
D32	yes	no	Class Code/Revision ID	0x07800003	0x08
D32	yes	Yes 15..0	BIST/Header Type/Latency Timer/Cache Line Size	0x00000000	0x0C
D32	yes	Yes	PCI Base Address for Memory Mapped Runtime Registers	0x00000000	0x10
D32	yes	Yes	PCI Base Address for 1 for I/O Mapped Configuration Registers	0x00000001	0x14
D32	yes	Yes	PCI Base Address 2 for Local Address Space 0	0x00000000	0x18
D32	yes	Yes	PCI Base Address 3 for Local Address Space 1	0x00000000	0x1C
D32	yes	No	Unused Base Address	0x00000000	0x20
D32	yes	No	Unused Base Address	0x00000000	0x24
D32	yes	No	Cardbus CIS Pointer (Not Supported)	0x00000000	0x28
D32	yes	No	Subsystem ID/Subsystem Vendor ID	0x240001B5	0x2C
D32	yes	Yes	PCI Base Address to Local Expansion ROM	0x00000001	0x30
D32	yes	No	Reserved	0x00000000	0x34
D32	yes	No	Reserved	0x00000000	0x38
D32	yes	Yes 7..0	Max_lat/Min_Gnt/Interrupt Pin/Interrupt Line	0x00010000	0x3C

All registers may be written to or read from in byte, word, or Lword accesses.

### A.2.1 PCI Configuration ID Register (PCI Configuration Offset 0x00)

<b>D0..15</b>	Vendor ID- Identifies the manufacturer of the device. Defaults to the PCI SIG issued vendor ID of PLX (0x10B5) if no serial EEPROM is present and pin NB# (no local bus initialization) is asserted low.
<b>D16..31</b>	Device ID - Identifies the particular device. Defaults to the PLX part number for PCI interface chip (0x9080) if no serial EEPROM is present and pin NB#(no local bus initialization) is asserted low.



### A.2.2 PCI Command Register (PCI Configuration Offset 0x04)

<b>D0</b>	I/O Space A value of 1 allows the device to respond to I/O space accesses. A value of 0 disables the device from responding to I/O space accesses.
<b>D1</b>	Memory Space A value of 1 allows the device to respond to memory space accesses. A value of 0 disables the device from responding to memory space accesses.
<b>D2</b>	Master Enable. Controls a device's ability to act as a master on the PCI bus. A value of 1 allows the device to behave as a bus master. A value of 0 disables the device from generating bus master accesses. This bit must be set for the PCI 9080 to perform Direct Master or DMA cycles.
<b>D3</b>	Special Cycle. (This bit is not supported.)
<b>D4</b>	Memory Write/Invalidate. A value of 1 enables memory write/invalidate. A value of 0 disables memory write/invalidate.
<b>D5</b>	VGA Palette Snoop. (This bit is not supported.)
<b>D6</b>	Parity Error Response A value of 0 indicates that a parity error is ignored and operation continues. A value of 1 indicates that parity checking is enabled.
<b>D7</b>	Wait Cycle Control. Controls whether the device does address/data stepping. A 0 value indicates the device never does stepping. A value of 1 indicates that the device always does stepping. <b>Note:</b> <i>Hardcoded to 0.</i>
<b>D8</b>	SERR# Enable A value of 1 enables the SERR# driver. A value of 0 disables the driver.
<b>D9</b>	Fast Back-to-Back Enable. Indicates what type of fast back-to-back transfers a Master can perform on the bus. A value of 1 indicates that fast back-to-back transfers can occur to any agent on the bus. A value of 0 indicates fast back-to-back transfers can only occur to the same agent as the previous cycle.
<b>D10..15</b>	Reserved

### A.2.3 PCI Status Register (PCI Configuration Offset 0x06)

<b>D0..5</b>	Reserved
<b>D6</b>	If high, supports User definable features. This bit can only be written from the local side. It is read-only from the PCI side.
<b>D7</b>	Fast Back-to-Back Capable. When this bit is set to a 1, it indicates the adapter can accept fast back-to-back transactions. A 0 indicates the adapter cannot.
<b>D8</b>	Master Data Parity Error Detected This bit is set to a 1 when three conditions are met: <ol style="list-style-type: none"> <li>1. the PCI9080 asserted PERR# itself or observed PERR# asserted.</li> <li>2. the PCI9080 was the bus master for the operation in which the error occurred.</li> <li>3. the Parity Error Response bit in the Command Register is set.</li> </ol> Writing a 1 to this bit clears the bit to a 0.
<b>D9..10</b>	DEVSEL Timing. Indicates timing for DEVSEL# assertion. A value of 01 indicates a medium decode. <b>Note:</b> <i>Hardcode to 01.</i>
<b>D11</b>	Target Abort When this bit is set to a 1, this bit indicates the PCI9080 has signaled a target abort. Writing a 1 to this bit clears the bit (0).
<b>D12</b>	Received Target Abort When set to a 1, this bit indicates the PCI9080 has signaled a target abort. Writing a 1 to this bit clears the bit (0).
<b>D13</b>	Master Abort When set to a 1, this bit indicates the PCI9080 has generated a master abort signal. Writing a 1 to this bit clears the bit (0).
<b>D14</b>	Signal System Error When set to a 1, this bit indicates the PCI9080 has reported a system error on the SERR# signal. Writing a 1 to this bit clears the bit (0).
<b>D15</b>	Detected Parity Error When set to a 1, this bit indicates the PCI9080 has detected a PCI bus parity error, even if parity error handling is disabled (the Parity Error Response bit in the Command Register is clear). One of three conditions can cause this bit to be set: <ol style="list-style-type: none"> <li>1. the PCI9080 detected a parity error during a PCI address phase.</li> <li>2. the PCI9080 detected a data parity error when it was the target of a write.</li> <li>3. the PCI9080 detected a data parity error when performing a master read operation.</li> </ol> Writing a 1 to this bit clears the bit (0).

### A.2.4 PCI Revision ID Register (PCI Configuration Offset 0x08)

<b>D0..7</b>	Revision ID. The silicon revision of the PCI9080.
--------------	---

### A.2.5 PCI Class Code Register (PCI Configuration Offset 0x09..0B)

<b>D0..7</b>	Register level programming interface 0x00 = Queue Ports at 0x40 and 0x44. 0x01 = Queue Ports at 0x40 and 0x44, and Int Status and Int Mask at 0x30 and 0x34, respectively.
<b>D8..15</b>	Sub-class Code 0x80 = Other Communications device.
<b>D16..D23</b>	Base Class Code. 0x07 = Communications Device.

### A.2.6 PCI Cache Line Size Register (PCI Configuration Offset 0x0C)

<b>D0..7</b>	System cache line size in units of 32-bit words.
--------------	--

### A.2.7 PCI Latency Timer Register (PCI Configuration Offset 0x0D)

<b>D0..7</b>	PCI Latency Timer. Units of PCI bus clocks, the amount of time the PCI9080, as a bus master, can burst data on the PCI bus.
--------------	---

### A.2.8 PCI Header Type Register (PCI Configuration Offset 0x0E)

<b>D0..6</b>	Configuration Layout Type. Specifies the layout of bits 0x10 through 0x3F in configuration space. Only one encoding 0 is defined. All other encodings are reserved.
<b>D7</b>	Header Type. A 1 indicates multiple functions. A 0 indicates a single function.

### A.2.9 PCI Built-In Self Test (BIST) Register (PCI Configuration Offset 0x0F)

<b>D0..3</b>	A value of 0 means the device has passed its test. Non-zero values mean the device failed. Device specific failure codes can be encoded in the non-zero value.
<b>D4..5</b>	Reserved, Device returns 0.
<b>D6</b>	PCI writes a 1 to invoke BIST. Generates an interrupt to local bus. Local bus resets the bit when BIST is complete. Software should fail device if BIST is not complete after 2 seconds. Refer to Runtime registers for interrupt control/status.
<b>D7</b>	Return 1 if device supports BIST. Return 0 if the device is not BIST compatible.

**A.2.10 PCI Base Address Register for Memory Access to Runtime Registers (PCI Configuration Offset 0x010)**

<b>D0</b>	Memory space indicator A value of 0 indicates register maps into Memory space. A value of 1 indicates the register maps into I/O space <b>Note:</b> <i>Hardcoded to 0.</i>
<b>D1..2</b>	Location of register: 00 - Locate anywhere in 32 bit memory address space 01 - Locate below 1 Mbyte memory address space 10 - Locate anywhere in 64 bit memory address space 11 - Reserved <b>Note:</b> <i>Hardcoded to 0.</i>
<b>D3</b>	Prefetchable. A value of 1 indicates there are no side effects on reads. This bit has no effect on the operation of the PCI 9080. <b>Note:</b> <i>Hardcoded to 0.</i>
<b>D4..7</b>	Memory Base Address. Memory base address for access to Local, Runtime and DMA registers. (default is 256 bytes.) <b>Note:</b> <i>Hardcoded to 0.</i>
<b>D8..31</b>	Memory Base Address. Memory base address for access to Local, Runtime, and DMA registers.

**A.2.11 PCI Base Address Register for I/O Access to Runtime Registers (PCI Configuration Offset 0x14)**

<b>D0</b>	Memory space indicator A value of 0 indicates register maps into Memory space. A value of 1 indicates the register maps into I/O space. <b>Note:</b> <i>Hardcoded to 1</i>
<b>D1</b>	Reserved
<b>D2..7</b>	I/O Base Address. Base Address for I/O access to runtime registers. (Minimum Block Size = 128 bytes.) <b>Note:</b> <i>Hardcoded to 0</i>
<b>D8..31</b>	I/O Base Address. Base Address for I/O access to Local, Runtime, and DMA Registers.

**A.2.12 PCI Base Address Register for Memory Access to Local Address Space 0 (PCI Configuration Offset 0x18)**

<b>D0</b>	Memory space indicator A value of 0 indicates register maps into Memory space. A value of 1 indicates the register maps into I/O space. (Specified in Local Address Space 0 Range Register)
<b>D1..2</b>	Location of register (if memory space). Location values: 00 - Locate anywhere in 32 bit memory address space 01 - Locate below 1 Mbyte memory address space 10 - Locate anywhere in 64 bit memory address space 11 - Reserved (Specified in Local Address Space 0 Range Register.)
<b>D3</b>	Prefetchable (if memory space) A value of 1 indicates there are no side effects on reads. This bit reflects the value of bit 3 in the LASORR register and provides only status to the system. This bit has no effect on the operation of the PCI 9080. Prefetching features of this address space are controlled by the associated Bus Region Descriptor Register. (Specified in LASORR register.) If I/O Space, bit 3 is included in the base address.
<b>D4..31</b>	Memory Base Address Memory base address for access to Local Address Space 0.

**A.2.13 PCI Base Address Register for Memory Access to Local Address Space 1 (PCI Configuration Offset 0x1C)**

<b>D0</b>	Memory space indicator A value of 0 indicates register maps into Memory space. A value of 1 indicates the register maps into I/O space. (Specified in Local Address Space 1 Range Register.)
<b>D1..2</b>	Location of register (if memory space). Location values: 00 - Locate anywhere in 32 bit memory address space 01 - Locate below 1 Mbyte memory address space 10 - Locate anywhere in 64 bit memory address space 11 - Reserved (Specified in Local Address Space 1 Range Register.)
<b>D3</b>	Prefetchable (if memory space) A value of 1 indicates there are no side effects on reads. This bit reflects the value of bit 3 in the LAS1RR register and provides only status to the system. This bit has no effect on the operation of the PCI 9080. Prefetching features of this address space are controlled by the associated Bus Region Descriptor Register. (Specified in LAS1RR register.) If I/O Space, bit 3 is included in the base address.
<b>D4..31</b>	Memory Base Address Memory base address for access to Local Address Space 1.

**A.2.14 PCI Base Address Register (PCI Configuration Offset 0x20)**

<b>D0..31</b>	Reserved
---------------	----------

**A.2.15 PCI Base Address Register (PCI Configuration Offset 0x24)**

<b>D0..31</b>	Reserved
---------------	----------

**A.2.16 PCI Cardbus CIS Pointer Register (PCI Configuration Offset 0x28)**

<b>D0..31</b>	Cardbus Information Structure Pointer for PCMCIA. (Not supported.)
---------------	--

### A.2.17 PCI Subsystem Vendor ID Register (PCI Configuration Offset 0x2C)

<b>D0..15</b>	Subsystem Vendor ID (unique add-in board Vendor ID).
---------------	--

### A.2.18 PCI Subsystem ID Register (PCI Configuration Offset 0x2E)

<b>D0..15</b>	Subsystem ID (unique add-in board Device ID).
---------------	---

### A.2.19 PCI Expansion ROM Base Register (PCI Configuration Offset 0x30)

<b>D0</b>	Address Decode Enable A value of 1 indicates the device accepts accesses to the expansion ROM address. A value of 0 indicates the device does not accept accesses to expansion ROM space. Should be set to 1 by PCI host if expansion ROM is present.
<b>D1..10</b>	Reserved
<b>D11..31</b>	Expansion ROM Base Address (upper 21bits)

### A.2.20 PCI Interrupt Line Register (PCI Configuration Offset 0x3C)

<b>D0..7</b>	Interrupt Line Routing Value. Indicates which input of the system interrupt controller(s) to which the interrupt line of the device is connected.
--------------	--

### A.2.21 PCI Interrupt Pin Register (PCI Configuration Offset 0x3D)

<b>D0..7</b>	Interrupt Pin register. Indicates which interrupt pin the device uses. The following values are decoded: 0=No Interrupt Pin 1=INTA# 2=INTB# 3=INTC# 4=INTD# <b>Note:</b> PCI 9080 supports only one PCI interrupt pin (INTA#).
--------------	---

### A.2.22 PCI Min\_Gnt Register (PCI Configuration Offset 0x3E)

<b>D0..7</b>	Min_Gnt. Used to specify how long a burst period the device needs assuming a clock rate of 33 MHz. Value is multiple of ¼ use increments.
--------------	--

### A.2.23 PCI Max\_Lat Register (PCI Configuration Offset 0x3F)

<b>D0..7</b>	Max_Lat. Specifies how often the device must gain access to the PCI bus. Value is multiple of ¼ user increments.
--------------	---

### A.3 LOCAL CONFIGURATION REGISTERS

**Table D LOCAL CONFIGURATION REGISTERS**

PCI Offset from base Addr	Size	Access		Register Name	Value after Reset
		R	W		
0x00	D32	yes	yes	Range for PCI to Local Address Space 0	0xFFFFE00
0x04	D32	yes	yes	Local Base Address (Re-map) for PCI to Local Address Space 0	0x00000001
0x08	D32	yes	no	Mode Arbitration	
0x0C	D32	yes	no	Big/Little Endian Descriptor	
0x10	D32	yes	yes	Range for PCI to Local Expansion ROM	0xFFFF0000
0x14	D32	yes	Bit Dep.	Local Base Address (Re-map) for PCI to Local Expansion ROM and BREQo control	0x00000000
0x18	D32	yes	yes	Bus Region Descriptions for PCI Local Accesses	0x42000143 (Cx Mode)
0x1C	D32	yes	Bit Dep.	Range for Direct Master to PCI	0x00000000
0x20	D32	yes	Bit Dep.	Local Base Address for Direct Master to PCI Memory	0x00000000
0x24	D32	yes	Bit Dep.	Local Base Address for Direct Master to PCI Memory IO/CFG	0x00000000
0x28	D32	yes	Bit Dep.	PCI Base Address (Re-map) for Direct Master to PCI	0x00000000
0x2C	D32	yes	yes	PCI Configuration Address Register for Direct Master to PCI IO/CFG	0x00000000
0xF0	D32	yes	yes	Range for PCI to Local Address Space 1	0x00000000
0xF4	D32	yes	yes	Local Base Address (Remap) for PCI to Local Address Space 1	0x00000000
0xF8	D32	yes	yes	Local Bus Region Descriptor (Space 1) for PCI to Local Accesses	0x00000000

### A.3.1 Local Address Space 0 Range Register for PCI to Local bus (PCI Offset 0x00)

<b>D0</b>	Memory space indicator A value of 0 indicates Local address space 0 maps into PCI memory space. A value of 1 indicates address space 0 maps into PCI I/O space.
<b>D1..2</b>	If mapped into memory space, encoded as follows (D1 being the LSB): Meaning 00-locate anywhere in 32 bit PCI address space 01-locate below 1 Meg in PCI address space 10-locate anywhere in 64 bit PCI address space 11-reserved If mapped into I/O space, bit 1 must be a 0. Bit 2 is included with bits 3 through 31 to indicate decoding range.
<b>D3</b>	If mapped into memory space, a value of 1 indicates that reads are pre-fetchable (bit has no effect on the PCI9080, but it is used for system status). If mapped into I/O space, bit is included with bits [31:2] to indicate decoding range. If mapped into I/O space, bit is included with bits 2 through 31 to indicate decoding range
<b>D4..31</b>	Specifies which PCI address bits to use for decoding a PCI access to local bus space 0. Each bit corresponds to a PCI address bit. Bit 31 corresponds to Address bit 31. Write a value of 1 to all others (used in conjunction with PCI Configuration register 0x18). Default is 1 Meg.

### A.3.2 Local Address Space 0 Local Base Address (Re-map) Register for PCI to Local Bus (PCI Offset 0x04)

<b>D0</b>	Space 0 Enable A 1 value enables decoding of PCI addresses for Direct Slave access to local space 0. A value of 0 disables Decode. If this bit is set to 0, the PCI BIOS may not allocate (assign) the base address for Space 0. <b>Note:</b> <i>Must be set to 1 for any Direct Slave access to Space 0.</i>
<b>D1</b>	Reserved
<b>D2..3</b>	If local space 0 is mapped into memory space, bits are not used. If mapped into I/O space, bit is included with bits 4 through 31 for re-mapping.
<b>D4..31</b>	Re-map of PCI Address to Local Address Space 0 into a Local Address Space. The bits in this register re-map (replace) the PCI Address bits used in decode as the Local Address bits. <b>Note:</b> <i>Remap Address value must be multiple of Range (not the Range register)</i>



### A.3.3 Mode/Arbitration Register (PCI Offset 0x08)

<b>D0..7</b>	Local bus Latency Timer Number of local bus clock cycles before negating HOLD and releasing the local bus. This timer is also used with bit 27 to delay BREQ input to give up the local bus only when this timer expires.
<b>D8..15</b>	Local bus Pause Timer Number of local bus clock cycles before reasserting HOLD after releasing the local bus. <b>Note:</b> <i>Applicable only to DMA operation.</i>
<b>D16</b>	Local bus Latency Timer Enable A value of 1 enables latency timer.
<b>D17</b>	Local bus Pause Timer Enable A value of 1 enables pause timer.
<b>D18</b>	Local bus BREQ Enable A value of 1 enables local bus BREQ input. When the BREQ input is active, PCI 9080 negates HOLD and releases the local bus.
<b>D19..20</b>	DMA Channel Priority A value of 00 indicates a rotational priority scheme. A value of 01 indicates Channel 0 has priority. A value of 10 indicates Channel 1 has priority. A value of 11 is reserved.
<b>D21</b>	Local bus direct slave give up bus mode When set to 1, PCI 9080 negates HOLD and releases the local bus when the Direct Slave write FIFO becomes empty during a Direct Slave write or when the Direct Slave read FIFO becomes full during a Direct Slave read.
<b>D22</b>	Direct slave LLOCKo# Enable A value of 1 enables PCI Direct Slave locked sequences. A value of 0 disables Direct Slave locked sequences.
<b>D23</b>	PCI Request Mode A value of 1 causes PCI9080 to negate REQ when it asserts FRAME during a master cycle. A value of 0 causes PCI 9080 to leave REQ asserted for the entire bus master cycle.
<b>D24</b>	PCI Rev 2.1 Mode When set to 1, PCI 9080 operates in Delayed Transaction mode for Direct Slave Reads. PCI 9080 issues a RETRY and prefetches the read data.
<b>D25</b>	PCI Read No Write Mode A value of 1 forces a retry on writes if read is pending. A value of 0 allows writes to occur while read is pending.
<b>D26</b>	PCI Read with Write Flush Mode A value of 1 submits a request to flush a pending read cycle if a write cycle is detected. A value of 0 submits a request to not effect pending reads when a write cycle occurs. (PCI v2.1 compatible)
<b>D27</b>	Gate the Local Bus Latency Timer with BREQ If this bit is set to 0, PCI 9080 gives up the local bus during Direct Slave or DMA transfer after the current cycle (if enabled and BREQ is sampled). If this bit is set to 1, PCI 9080 gives up the local bus only (if BREQ is sampled) and the Local Bus Latency Timer is enabled and expires during Direct Slave or DMA transfer.
<b>D28</b>	PCI Read No Flush Mode A value of 1 submits request to not flush the read FIFO if PCI read cycle completes (Read Ahead mode). A value of 0 submits request to flush read FIFO if PCI read cycle completes.

<b>D29</b>	<p>Reads Device or Vendor ID</p> <p>If set to 0, reads from the PCI Configuration Register address 0x00 and returns the Device ID and Vendor ID.</p> <p>If set to 1, reads from the PCI Configuration Register address 0x00 and returns the Subsystem and Subsystem Vendor ID.</p>
<b>D30..31</b>	Reserved

### A.3.4 Big/Little Endian Descriptor Register (PCI Offset 0x0C)

<b>D0</b>	<p>Configuration Register Big Endian Mode</p> <p>A value of 1 specifies use of Big Endian data ordering for local accesses to the configuration registers.</p> <p>A value of 0 specifies Little Endian ordering. Big Endian mode can be specified for configuration register accesses by asserting the BIGEND# pin during the address phase of the access.</p>
<b>D1</b>	<p>Direct Master Big Endian Mode</p> <p>A value of 1 specifies use of Big Endian data ordering for Direct Master accesses.</p> <p>A value of 0 specifies Little Endian ordering. Big BIGEND# input pin during the address phase of the access.</p>
<b>D2</b>	<p>Direct Slave Address Space 0 Big Endian Mode</p> <p>A value of 1 specifies use of Big Endian data ordering for Direct Slave accesses to Local Address space 0.</p> <p>A value of 0 specifies Little Endian ordering.</p>
<b>D3</b>	<p>Direct Slave Address Expansion ROM 0 Big Endian Mode</p> <p>A value of 1 specifies use of Big Endian data ordering for Direct Slave accesses to Expansion ROM.</p> <p>A value of 0 specifies Little Endian ordering.</p>
<b>D4</b>	<p>Big Endian Byte Lane Mode</p> <p>A value of 1 specifies that in Big Endian mode, use byte lanes 31:16 for a bit local bus and byte lanes 31:24 for an 8 bit local bus.</p> <p>A value of 0 specifies that in Big Endian mode, byte lanes 15:0 be used for a 16 bit local bus byte lanes 7:0 for an 8 bit local bus.</p>
<b>D5</b>	<p>Direct Slave Address Space 1 Big Endian Mode</p> <p>A value of 1 specifies use of Big Endian data ordering for Direct Slave accesses to local Address Space 1.</p> <p>A value of 0 specifies Little Endian ordering.</p>
<b>D6</b>	<p>DMA Channel 1 Big Endian Mode</p> <p>A value of 1 specifies use of Big Endian data ordering for DMA Channel 1 accesses to the local Address Space.</p> <p>A value of 0 specifies Little Endian ordering.</p>
<b>D7</b>	<p>DMA Channel 0 Big Endian Mode</p> <p>A value of 1 specifies use of Big Endian data ordering for DMA Channel 0 accesses to the local Address Space.</p> <p>A value of 0 specifies Little Endian ordering.</p>
<b>D8..31</b>	Reserved

### A.3.5 Local Expansion ROM Range Register for PCI to Local Bus (PCI Offset 0x10)

<b>D0..10</b>	Reserved
<b>D11..31</b>	Specifies which PCI address bits will be used for decoding a PCI to local bus expansion ROM. Each of the bits corresponds to an Address bit. Bit 31 corresponds to Address bit 31. Write a value of 1 to all bits to be included in decode and a 0 to all others (Used in conjunction with PCI Configuration register 0x30). Default is 64 Kbytes.

### A.3.6 Local Expansion ROM Local Base Address (Re-map) register for PCI to Local Bus and BREQo Control (PCI Offset 0x14)

<b>D0..3</b>	Direct Slave BREQo (Backoff Requests Out) Delay Clocks. Number of local bus clocks in which a Direct Slave HOLD request is pending and a Local Direct Master access is in progress and not being granted the bus (HOLDA) before asserting BREQo. Once asserted, BREQo remains asserted until the PCI900 receives HOLDA (LSB = 8 or 64 clocks).
<b>D4</b>	Local Bus BREQo Enable A 1 value enables the PCI9080 to assert the BREQo output.
<b>D6..10</b>	Reserved
<b>D11..31</b>	Re-map of PCI Expansion ROM space into a Local address space. The bits in this register re-map (replace) the PCI address bits used in decode as the Local address bits.

### A.3.7 Local Address Space 0/Expansion ROM Bus Region Descriptor Register (PCI Offset 0x18)

<b>D0..1</b>	Memory Space 0 Local Bus Width A value of 00 indicates a bus width of 8 bits A value of 01 indicates a bus width of 16 bits A value of 10 or 11 indicates a bus width of 32 bits The bus width is forced to 16 bits for the Sx mode.
<b>D2..5</b>	Memory Space 0 Internal Wait States (data to data; 0-15 wait states).
<b>D6</b>	Memory Space 0 Ready Input Enable A 1 value enables Ready input. A value of 0 disables the Ready input.
<b>D7</b>	Memory Space 0 BTERM# Input Enable A 1 value enables BTERM# input. A value of 0 disables the BTERM# input.
<b>D8</b>	Memory Space 0 Prefetch Disable If mapped into memory space, A 0 enables read pre-fetching. A value of 1 disables prefetching. If prefetching is disabled, the PCI9080 will disconnect after each memory read.
<b>D9</b>	Expansion ROM Space Prefetch Disable. A 0 enables read prefetching. A 1 disables prefetching If prefetching is disabled, the PCI9080 will disconnect after each memory read.
<b>D10</b>	Read Prefetch Count Enable. When set to a 1 and memory prefetching is enabled, PCI 9080 prefetches up to the number of Lwords specified in the prefetch count. When set to 0, PCI 9080 ignores the count and continues prefetching until terminated by the PCI bus.
<b>D11..14</b>	Prefetch Counter Number of Lwords to prefetch during memory read cycles (0-15). A count of zero selects a prefetch of 16 Lwords.

<b>D15</b>	Reserved
<b>D16..17</b>	Expansion ROM Space Local Bus Width. A value of 00 indicates a bus width of 8 bits A value of 01 indicates a bus width of 16 bits A value of 10 or 11 indicates a bus width of 32 bits.
<b>D21..18</b>	Expansion ROM Space Internal Wait States (data to data; 0-15 wait states).
<b>D22</b>	Expansion ROM Space Ready Input Enable A 1 value enables Ready input. A value of 0 disables the Ready input.
<b>D23</b>	Expansion ROM Space BTERM# Input Enable A 1 value enables BTERM# input. A value of 0 disables the BTERM# input. If this bit is set to 1, PCI 9080 bursts four Lword maximum at a time.
<b>D24</b>	Memory Space 0 Burst Enable A 1 value enables bursting. A value of 0 disables bursting. If burst is disabled, the local bus performs continuous single cycles for burst PCI read/write cycles.
<b>D25</b>	Extra Long serial EEPROM. A value of 1 loads the Subsystem ID and Local Address Space 1 registers. A value of 0 indicates not to load them.
<b>D26</b>	Expansion ROM Space Burst Enable A 1 value enables bursting. A value of 0 disables bursting. If burst is disabled, the local bus performs continuous single cycles for burst PCI read/write cycles.
<b>D27</b>	Direct Slave PCI write mode A 0 indicates that the PCI9080 should disconnect when the Direct Slave write FIFO is full. A 1 indicates that the PCI9080 should de-assert TRDY when the write FIFO is full.
<b>D28..31</b>	PCI Target Retry Delay Clocks. Contains the value (multiplied by 8) of the # of PCI bus clocks after receiving a PCI-Local read or write access and not successfully completing a transfer. Only pertains to Direct Slave writes when bit 27 is set to 1.

### A.3.8 Local Range register for Direct Master to PCI (PCI Offset 0x1C)

<b>D0..15</b>	Reserved (64 KB increments)
<b>D16..31</b>	Specifies which local address bits to use for decoding a Local to PCI bus access. Each of the bits corresponds to an address bit. Bit 31 corresponds to Address bit 31. A value of 1 should be written to all bits that should be included in decode and a 0 to all others. This range is used for Direct Master memory, I/O, or configuration accesses.

### A.3.9 Local Bus Base Address register for Direct Master to PCI Memory (PCI Offset 0x20)

<b>D0..15</b>	Reserved
<b>D16..31</b>	Assigns a value to the bits which will be used to decode a Local to PCI memory access.

### A.3.10 Local Base Address for Direct Master to PCI IO/CFG Register (PCI Offset 0x24)

<b>D0..15</b>	Reserved
<b>D16..31</b>	Assigns a value to the bits to be used for decoding a Local to PCI I/O or configuration access. This base address is used for Direct Master I/O and configuration accesses.

### A.3.11 PCI Base Address (Re-map) register for Direct Master to PCI Memory (PCI Offset 0x28)

<b>D0</b>	Direct Memory Access Enable A value of 1 enables decode of Direct Master Memory accesses. A value of 0 disables decode of Direct Master Memory accesses.
<b>D1</b>	Direct Master I/O Access Enable A value of 1 enables decode of Direct Master I/O accesses. A value of 0 disables decode of Direct Master I/O accesses.
<b>D2</b>	LOCK# Input Enable A 1 value enables LOCK# input. A value of 0 disables the LOCK# input.
<b>D3</b>	Direct Master Red Prefetch Size control 00=PCI 9080 continues to prefetch read data from the PCI bus until Direct Master access is finished. This may result in an additional four unneeded Lwords being prefetched from the PCI bus. 01= Prefetch up to four Lwords from the PCI bus 10= Prefetch up to eight Lwords from the PCI bus 11= Prefetch up to 16 Lwords from the PCI bus. If PCI memory prefetch is not wanted, performs a Direct Master single cycle. The direct master burst reads must not exceed the programmed limit.
<b>D4</b>	Direct Master PCI read mode A value of 0 indicates that the PCI9080 should release the PCI bus when the read FIFO becomes full. A value of 1 indicates that the PCI9080 should keep the PCI bus and de-assert IRDY when the read FIFO becomes full.
<b>D5..8,10</b>	Programmable Almost Full flag. When the number of entries in the 32 word direct master write FIFO exceed this value, the output pin DMPAF# is asserted low
<b>D9</b>	Write and Invalidate Mode. When set to 1, PCI 9080 waits for 8 or 16 Lwords to be written from the local bus before starting PCI accesses. When set , all local Direct Master to PCI write accesses must be 8 or 16 Lwords bursts. Use in conjunction with (PCI 0x04)
<b>D10</b>	Direct Master Prefetch Limit. If set to 1, don't prefetch past 4K (4098 bytes) boundaries.
<b>D13</b>	I/O Remap Select. When set to 1, forces PCI address bits [31:16] to all zeros. When set to 0, uses bits [31:16] of this register as PCI address bits [31:16].
<b>D14..15</b>	Direct Master Write Delay. This register is used to delay the PCI bus request after direct master burst write cycle has started. Values: 00=No delay; start the cycle immediately 01=Delay 4 PCI clocks 10=Delay 8 PCI clocks 11=Delay 16 PCI clocks

<b>D16..31</b>	Re-map of Local to PCI space into a PCI address space. The bits in this register re-map (replace) the Local address bits used in decode as the PCI address bits. This PCI Remap address is used for Direct Master memory and I/O accesses.
----------------	---

### A.3.12 PCI Configuration Address Register for Direct Master to PCI IO/CFG (PCI Offset 0x2C)

<b>D0..1</b>	Configuration Type (00=Type 0, 01=Type 1)
<b>D2..7</b>	Register Number. If different register read/write is needed, this register value must be programmed and a new PCI configuration cycle must be generated.
<b>D8..10</b>	Function Number
<b>D11..15</b>	Device Number
<b>D16..23</b>	Bus Number
<b>D24..30</b>	Reserved
<b>D31</b>	Configuration Enable. A value of 1 allows Local to PCI I/O accesses to be converted to a PCI configuration cycle. The parameters in this table are used to generate the PCI configuration address.

### A.3.13 Local Address Space 1 Range Register for PCI to Local Bus (PCI Offset 0xF0)

<b>D0</b>	Memory Space Indicator A value of 0 indicates Local Address Space 1 maps into PCI memory space. A value of 1 indicates Local Address Space 1 maps into PCI I/O space.
<b>D1..2</b>	Encoded for Memory Space If mapped into memory space, encoding is as follows: 00-Locate anywhere in 32 bit PCI address space 01-Locate below 1 MB in PCI address space 10-Locate anywhere in 64 bit PCI address space 11-Reserved If mapped into I/O space, bit 1 must be set to 0. Bit 2 is included with bits [31:3] to indicate decoding range.
<b>D3</b>	If mapped into memory space, a value of 1 indicates reads are prefetchable (bit has no effect on the operation of the PCI 9080, but is for system status). If mapped into I/O space, bit is included with bits [31:2] to indicate decoding range.
<b>D4..31</b>	Specifies which PCI address bits to use for decoding a PCI access to local bus space 1. Each of the bits corresponds to a PCI address bit. Bit 31 corresponds to Address bit 31. Write a value of 1 to all bits that must be included in decode and a 0 to all others (Used in conjunction with PCI Configuration Register). Default is 1 MB.

### A.3.14 Local Address Space 1 Local Base Address (Remap) Register (PCI Offset 0xF4)

<b>D0</b>	Space 1 Enable A value of 1 enables decoding of PCI addresses for Direct Slave access to local space 1. A value of 0 disables decoding. If this bit is set to 0, the PCI BIOS may not allocate (assign) the base address for Space 1. <b>Note:</b> <i>Must be set to 1 for any Direct Slave access to Space 1.</i>
<b>D1</b>	Reserved
<b>D2..3</b>	If local space 1 is mapped into memory space, bits are not used. If mapped I/O space, bit is included with bits [31:4] for remapping.
<b>D4..31</b>	Remap of PCI Address to Local Address space 1 into a Local Address Space. The bits in this register remap (replace) the PCI Address bits used in decode as the Local Address bits.

### A.3.15 Local Address Space 1 Bus Region Descriptor Register (PCI Offset 0xF8)

<b>D0..1</b>	Memory Space 1 Local Bus Width A value of 00 indicates bus width of 8 bits A value of 01 indicates bus width of 16 bits A value of 10 indicates bus width of 32 bits
<b>D2..5</b>	Memory space 1 Internal Wait States (data to data; 0-15 wait states).
<b>D6</b>	Memory space 1 Ready Input Enable A value of 1 enables BTERM# input. A value of 0 disables Ready input.
<b>D7</b>	Memory space 1 BTERM# Input Enable A value of 1 enables BTERM# input. A value of 0 disables BTERM# input. If this bit is set to 0, PCI 9080 bursts four Lword maximum at a time.
<b>D8</b>	Memory space 1 Burst Enable A value of 1 enables bursting. A value of 0 disables bursting. If burst is disabled, the local bus performs continuous single cycle for burst PCI read/write cycles.
<b>D9</b>	Memory space 1 Prefetch Disable If mapped into memory space A value of 0 enables read prefetching. A value of 1 disables read prefetching. If prefetching is disabled, PCI 9080 disconnects after each memory read.
<b>D10</b>	Read Prefetch Count Enable When set to 1 and memory prefetching is enabled, PCI 9080 prefetches up to the number of Lwords specified in the prefetch count. When set to 0, PCI 9080 ignores the count and continues prefetching until terminated by the PCI bus.
<b>D11..14</b>	Prefetch Counter Number of Lwords to prefetch during memory read cycles (0-15).
<b>D15..31</b>	Reserved

## A.4 RUNTIME REGISTERS

**Table E RUN TIME REGISTERS**

PCI Offset from base Addr	Size	Access		Register Name	Value after reset
		R	W		
0x40	D32	yes	yes	Mailbox Register 0	0x00000000
0x44	D32	yes	yes	Mailbox Register 1	0x00000000
0x48	D32	yes	yes	Mailbox Register 2	0x00000000
0x4C	D32	yes	yes	Mailbox Register 3	0x00000000
0x50	D32	yes	yes	Mailbox Register 4	0x00000000
0x54	D32	yes	yes	Mailbox Register 5	0x00000000
0x58	D32	yes	yes	Mailbox Register 6	0x00000000
0x5C	D32	yes	yes	Mailbox Register 7	0x00000000
0x60	D32	yes	yes	PCI to Local Doorbell Register	0x00000000
0x64	D32	yes	yes	Local to PCI Doorbell Register	0x00000000
0x68	D32	yes	Bit Dep.	Interrupt Control/Status	0x00000000
0x6C	D32	yes	Bit Dep.	EEPROM Control, PCI Command Codes, User I/O Control, Init Control	0x001767E
0x70	D32	yes	Bit Dep.	Device ID/Vendor ID	0x00
0x74	D32	yes	Bit Dep.	Unused/Revision ID	0x00
0x70	D32	yes	Bit Dep.	Mailbox Register 0 (1)	0x00
0x74	D32	yes	Bit Dep.	Mailbox Register 1 (1)	0x00

### A.4.1 Mailbox Register 0 (PCI Offset 0x40)

<b>D0..31</b>	32 bit mailbox register
---------------	-------------------------

### A.4.2 Mailbox Register 1 (PCI Offset 0x44)

<b>D0..31</b>	32 bit mailbox register
---------------	-------------------------

### A.4.3 Mailbox Register 2 (PCI Offset 0x48)

<b>D0..31</b>	32 bit mailbox register
---------------	-------------------------



**A.4.4 Mailbox Register 3 (PCI Offset 0x4C)**

<b>D0..31</b>	32 bit mailbox register
---------------	-------------------------

**A.4.5 Mailbox Register 4 (PCI Offset 0x50)**

<b>D0..31</b>	32 bit mailbox register
---------------	-------------------------

**A.4.6 Mailbox Register 5 (PCI Offset 0x54)**

<b>D0..31</b>	32 bit mailbox register
---------------	-------------------------

**A.4.7 Mailbox Register 6 (PCI Offset 0x58)**

<b>D0..31</b>	32 bit mailbox register
---------------	-------------------------

**A.4.8 Mailbox Register 7 (PCI Offset 0x5C)**

<b>D0..31</b>	32 bit mailbox register
---------------	-------------------------

**A.4.9 PCI to Local Doorbell Register Description (PCI Offset 0x60)**

<b>D0..31</b>	<p>Doorbell register.</p> <p>A PCI master can write to this register and it will generate a local interrupt to the local processor. The local processor can then read this register to determine which doorbell bit was asserted. The PCI master sets a doorbell by writing a 1 to a particular bit. The local processor can clear a doorbell bit by writing a 1 to that bit position.</p>
---------------	--

**A.4.10 Local to PCI Doorbell Register Description (PCI Offset 0x64)**

<b>D0..31</b>	<p>Doorbell register.</p> <p>The local processor can write to this register and it will generate a PCI interrupt. A PCI master can then read this register to determine which doorbell bit was asserted. The local processor sets a doorbell by writing a 1 to a particular bit. The PCI master can clear a doorbell bit by writing a 1 to that bit position.</p>
---------------	---

### A.4.11 Interrupt Control /Status (PCI Offset 0x68)

<b>D0</b>	Enable Local Bus LSERR#. A value of 1 enables PCI 9080 to assert LSERR# interrupt output when PCI bus Target Abort or Master Abort status bit is set in the PCI Status configuration register.
<b>D1</b>	Enable Local Bus SERR# when PCI parity error occurs during PCI 9080 Master Transfer or PCI 9080 Slave access or Outbound Free List FIFO Overflow Init.
<b>D2</b>	Generate PCI Bus SERR#. When this bit is set to 0, writing a 1 generates a PCI bus SERR#.
<b>D3</b>	Mailbox Interrupt Enable. A value of 1 enables a Local Interrupt to be generated when the PCI bus writes to Mailbox register 0-3. To clear the Local Interrupt, the Local master must read the Mailbox. Used in conjunction with Local interrupt enable.
<b>D4..7</b>	Reserved
<b>D8</b>	PCI Interrupt Enable. A value of 1 enables PCI interrupts.
<b>D9</b>	PCI Doorbell Interrupt Enable. A value of 1 enables doorbell interrupts. Used in conjunction with PCI interrupt enable. Clearing the doorbell interrupt bits that caused the interrupt also clears the interrupt.
<b>D10</b>	PCI Abort Interrupt Enable A value of 1 enables a master abort or master detect of a target abort to generate a PCI interrupt. Used in conjunction with PCI interrupt enable. Clearing the abort status bits also clears the PCI interrupt.
<b>D11</b>	PCI Local Interrupt Enable. A value of 1 enables a local interrupt input to generate a PCI interrupt. Use in conjunction with PCI Interrupt enable. Clearing the local bus cause of the interrupt also clears the interrupt.
<b>D12</b>	Retry Abort Enable. A value of 1 enables PCI 9080 to treat 256 Master consecutive retries to a Target as a target abort. A value of 0 enables PCI 9080 to attempt Master Retries indefinitely. <b>Note:</b> <i>for diagnostic purposes only.</i>
<b>D13</b>	Value of 1 indicates PCI doorbell interrupt is active.
<b>D14</b>	Value of 1 indicates PCI abort interrupt is active.
<b>D15</b>	Value of 1 indicates local interrupt is active (LINTi#).
<b>D16</b>	Local Interrupt Output Enable. A value of 1 enables local interrupt output.
<b>D17</b>	Local Doorbell Interrupt Enable. A value of 1 enables doorbell interrupts. Used in conjunction with Local interrupt enable. Clearing the local doorbell interrupt bits that caused the interrupt also clears the interrupt.
<b>D18</b>	Local DMA Channel 0 Interrupt Enable. A value of 1 enables DMA Channel 0 interrupts. Used in conjunction with Local interrupt enable. Clearing the DMA status bits also clears the interrupt.
<b>D19</b>	Local DMA Channel 1 Interrupt Enable. A value of 1 enables DMA Channel 1 interrupts. Used in conjunction with Local interrupt enable. Clearing the DMA status bits also clears the interrupt.
<b>D20</b>	Value of 1 indicates local doorbell interrupt is active.
<b>D21</b>	Value of 1 indicates DMA Ch 0 interrupt is active.
<b>D22</b>	Value of 1 indicates DMA Ch 1 interrupt is active.

<b>D23</b>	Value of 1 indicates BIST interrupt is active. BIST (Built-In Self Test) interrupt is generated by writing 1 to bit 6 of the PCI Configuration BIST register. Clearing bit 6 clears the interrupt. Refer to the BIST Register for a description of self test.
<b>D24</b>	Value of 0 indicates a Direct master was the bus master during a master or Target abort. (Not valid until abort occurs.)
<b>D25</b>	Value of 0 indicates a DMA CH 0 was the bus master during a master or Target abort. (Not valid until abort occurs.)
<b>D26</b>	Value of 0 indicates a DMA CH 1 was the bus master during a master or Target abort. (Not valid until abort occurs.)
<b>D27</b>	Value of 0 indicates a Target Abort was generated by the PCI 9080 after 256 consecutive Master retries to a Target. (Not valid until abort occurs.)
<b>D28</b>	Value of 1 indicates PCI wrote data to the Mailbox #0. Enabled only if MBOXINTENB is enabled (bit 3 high).
<b>D29</b>	Value of 1 indicates PCI wrote data to the Mailbox #1. Enabled only if MBOXINTENB is enabled (bit 3 high).
<b>D30</b>	Value of 1 indicates PCI wrote data to the Mailbox #2. Enabled only if MBOXINTENB is enabled (bit 3 high).
<b>D31</b>	Value of 1 indicates PCI wrote data to the Mailbox #3. Enabled only if MBOXINTENB is enabled (bit 3 high).

**A.4.12 Serial EEPROM Control, PCI Command Codes, User I/O Control, Init Control Register (PCI Offset 0x6C)**

<b>D0..3</b>	PCI Read Command Code for DMA This PCI command is sent out during DMA read cycles.
<b>D4..7</b>	PCI Write Command Code for DMA This PCI command is sent out during DMA write cycles.
<b>D8..11</b>	PCI Memory Read Command Code for Direct Master This PCI command is sent out during Direct Master read cycles.
<b>D12..15</b>	PCI Memory Write Command Code for Direct Master This PCI command is sent out during Direct Master write cycles.
<b>D16</b>	General Purpose Output A value of 1 will cause the USERO output to go high. A value of 0 will cause the output to go low.
<b>D17</b>	General Purpose Input A value of 1 indicates that USERI input pin is high. A value of 0 indicates that USERI pin is low.
<b>D18..23</b>	Reserved
<b>D24</b>	Serial EEPROM clock for Local or PCI bus reads or writes to serial EEPROM. Toggling this bit generates a serial EEPROM clock. (Refer to the manufacturer's data sheet for the particular EEPROM being used.)
<b>D25</b>	Serial EEPROM chip select. For Local or PCI bus reads or writes to serial EEPROM Setting this bit to a 1 provides the EEPROM chip select.
<b>D26</b>	Write bit to serial EEPROM. For writes, this output bit is the input to the serial EEPROM. It is clocked into the serial EEPROM by the serial EEPROM clock.
<b>D27</b>	Read serial EEPROM data bit. For reads, this input bit is the output of the serial EEPROM. It is clocked out of the serial EEPROM by the serial EEPROM clock.
<b>D28</b>	Serial EEPROM present A 1 in this bit indicates that an EEPROM is present.
<b>D29</b>	Reload Configuration Registers When this bit is 0, writing a 1 causes the PCI9080 to reload the local configuration registers from the serial EEPROM.
<b>D30</b>	PCI Adapter Software Reset A value of 1 written to this bit will hold the local bus logic in the PCI9080 reset and LRESETO# asserted. The contents of the PCI configuration registers and Shared Run Time registers will not be reset. Software Reset can only be cleared from the PCI bus. (Local bus remains reset until this bit is cleared.)
<b>D31</b>	Local Init Status. Value of 1 indicates local init done. Responses to PCI accesses will be RETRYs until this bit is set. While Input NB# is asserted low this bit will be forced to 1.

#### A.4.13 PCI Permanent Configuration ID Register (PCI Offset 0x70)

<b>D0..15</b>	Permanent Vendor ID. Identifies device manufacturer. <b>Note:</b> <i>Hardcoded to the PCI SIG issued vendor ID of PLX (10B5).</i>
<b>D16..31</b>	Permanent Device ID. Identifies the particular device. <b>Note:</b> <i>Hardcoded to the PLX part number for PCI interface chip PCI 9080.</i>

#### A.4.14 PCI Permanent Revision ID Register (PCI Offset 0x74)

<b>D0..15</b>	Permanent Revision ID. <b>Note:</b> <i>Hardcoded to the silicon revision of the PCI 9080.</i>
---------------	--

## A.5 LOCAL DMA REGISTERS

**Table F DMA REGISTERS**

PCI Offset from base Addr	Size	Access		Register Name	Value after Reset
		R	W		
0x80	D32	yes	yes	DMA Ch 0 Mode	0x00000003 (Cx and Jx modes)
0x84	D32	yes	yes	DMA Ch 0 PCI Address	0x00000000
0x88	D32	yes	yes	DMA Ch 0 Local Address	0x00000000
0x8C	D32	yes	yes	DMA Ch 0 Transfer Byte Count	0x00000000
0x90	D32	yes	yes	DMA Ch 0 Descriptor Pointer	0x00000000
0x94	D32	yes	yes	DMA Ch 1 Mode	0x00000003 (Cx and Jx modes)
0x98	D32	yes	yes	DMA Ch 1 PCI Address	0x00000000
0x9C	D32	yes	yes	DMA Ch 1 Local Address	0x00000000
0xA0	D32	yes	yes	DMA Ch 1 Transfer Byte Count	0x00000000
0xA4	D32	yes	yes	DMA Ch 1 Descriptor Pointer	0x00000000
0xA8	D32		Bit Dep.	Reserved/DMA Ch 1 Command/Status Register/DMA Ch 0 Command/Status	0x00000010
0xAC	D32	yes	Bit Dep.	Mode/ Arbitration Register	0x00000000
0xB0	D32	yes	Bit Dep.	DMA Threshold Register	0x00000000

### A.5.1 DMA Channel 0 Mode Register (PCI Offset 0x80)

<b>D0..1</b>	Local DMA Bus Width	A value of 00 indicates a DMA bus width of 8 bits. A value of 01 indicates DMA bus width of 16 bits. A value of 10 or 11 indicates a DMA bus width of 32 bits.
<b>D2..5</b>	Internal Wait States (data to data)	
<b>D6</b>	Ready Input Enable	A value of 1 enables Ready input. A value of 0 disables the Ready input.
<b>D7</b>	Bterm Input Enable	A value of 1 enables Bterm input. A value of 0 disables Bterm input.
<b>D8</b>	Local Burst Enable	A value of 1 enables bursting. A value of 0 disables bursting.

<b>D9</b>	<p>Chaining                      A 1 value indicates non-chaining mode is enabled.                      For chaining mode, the DMA source address, destination address and byte count are loaded from memory in PCI or Local Address Spaces.                      A 0 value indicates non-chaining mode is enabled.</p>
<b>D10</b>	<p>Done Interrupt Enable                      A 1 value enables interrupt when done.                      A 0 value disables interrupt when done.                      If DMA clear count mode is enabled, the interrupt won't occur until the byte count is cleared.</p>
<b>D11</b>	<p>Local Addressing Mode                      A 1 value indicates local addresses LA [31:2] to be held constant.                      A 0 value indicates local address is incremented.</p>
<b>D12</b>	<p>Demand Mode                      A value of 1 causes the DMA controller to operate in demand mode.                      In demand mode the DMA controller transfers data when it's DREQ# input is asserted. It asserts DACK# to indicate that the current local bus transfer is in response to the DREQ# input. The DMA controller transfers Lwords (32bits) of data. This may result in multiple transfers for an 8 or 16 bit bus.</p>
<b>D13</b>	<p>Write and Invalidate mode for DMA transfers.                      When set to 1, PCI 9080 performs Write and Invalidate cycles to the PCI bus. PCI 9080 supports Write and Invalidate sizes of 8 or 16 Lwords. The size is specified in the PCI Cache Line Size Register. If a size other than 8 or 16 is specified, PCI 9080 performs write transfers rather than Write and Invalidate transfers. Transfers must start and end at the Cache Line Boundaries.</p>
<b>D14</b>	<p>DMA EOT (End Of Transfer) Enable.                      A Value of 1 enables EOT# input pin.                      A Value of 0 disables EOT# input pin.</p>
<b>D15</b>	<p>DMA Stop Data Transfer Mode                      A Value of 0 sends a BLAST to terminate DMA transfer.                      A Value of 1 indicates an EOT asserted or DREQ# negated during demand mode DMA terminates the DMA transfer.</p>
<b>D16</b>	<p>DMA Clear Count Mode.                      When set to 1, the byte count in each chaining descriptor, if it is in local memory, is cleared when the corresponding DMA transfer is complete.  <b>Note:</b> <i>If chaining descriptor is in PCI memory, the count is not cleared.</i></p>
<b>D17</b>	<p>DMA Channel 0 Interrupt Select                      A Value of 1 routes the DMA Channel 0 interrupt to the PCI interrupt.                      A Value of 0 routes the DMA Channel 0 interrupt to the local bus interrupt.</p>
<b>D18..31</b>	Reserved

### A.5.2 DMA Channel 0 PCI Address Register (PCI Offset 0x84)

<b>D0..31</b>	PCI Address Register. This indicates where in the PCI memory space the DMA transfers (reads or writes) will start from.
---------------	---

### A.5.3 DMA Channel 0 Local Address Register (PCI Offset 0x88)

<b>D0..31</b>	Local Address Register. This indicates where in the local memory space the DMA transfers (reads or writes) will start from.
---------------	---

### A.5.4 DMA Channel 0 Transfer Size (Bytes) Register (PCI Offset 0x8C)

<b>D0..22</b>	DMA Transfer Size (Bytes). Indicates number of bytes to be transferred during DMA operation.
<b>D23..31</b>	Reserved

### A.5.5 DMA Channel 0 Descriptor Pointer Register (PCI Offset 0x90)

<b>D0</b>	Descriptor Location. A Value of 1 indicates PCI address space. A Value of 0 indicates Local Address Space.
<b>D1</b>	End of Chain A 1 value indicates end of chain. A 0 value indicates not end of chain descriptor. (Same as Nonchaining Mode)
<b>D2</b>	Interrupt after Terminal Count A 1 value causes an interrupt to be generated after the terminal count for this descriptor is reached. A 0 value disables interrupts from being generated.
<b>D3</b>	Direction of transfer A 1 value indicates transfers from local bus to PCI bus. A 0 value indicates transfers from PCI to local bus.
<b>D4..31</b>	Next Descriptor Address. Quad word aligned (Bit[3:0] = 0000).

### A.5.6 DMA Channel 1 Mode Register (PCI Offset 0x94)

<b>D0..1</b>	Local Bus Width A value of 00 indicates a bus width of 8 bits. A value of 01 indicates bus width of 16 bits. A value of 10 or 11 indicates a DMA bus width of 32 bits. The bus width is forced to 16 bits for the Sx mode
<b>D2..5</b>	Internal Wait States (data to data).
<b>D6</b>	Ready Input Enable A 1 value enables Ready input. A 0 value disables the Ready input.
<b>D7</b>	Bterm Input Enable A 1 value enables Bterm input. A value of 0 disables the Bterm input. If this bit is set to 0, PCI 9080 bursts four Lword maximum at a time.
<b>D8</b>	Local Burst Enable A 1 value enables Local bursting input. A value of 0 disables Local bursting. If burst is disabled, the local bus performs continuous single cycles for burst PCI read/write cycles.
<b>D9</b>	Chaining A 1 value indicates chaining mode enabled. For chaining mode, the DMA source address, destination address and byte count are loaded from memory in PCI or Local address spaces. A 0 value indicates non-chaining mode.
<b>D10</b>	Done Interrupt Enable A 1 value enables interrupt when done. A 0 value disables the interrupt when done. If DMA Clear Count Mode is enabled, the interrupt won't occur until the byte count is cleared.
<b>D11</b>	Local Addressing Mode A 1 value indicates local addresses LA[31:2] to be held constant. A 0 value indicates local addresses is incremented.



<b>D12</b>	<p>Demand Mode</p> <p>A value of 1 causes the DMA controller to operate in demand mode. In demand mode the DMA controller transfers data when its DREQ# input is asserted. It asserts DACK# to indicate that the current local bus transfer is in response to the DREQ# input. The DMA controller transfers Lwords (32bits) of data. This may result in multiple transfers for an 8 or 16 bit bus.</p>
<b>D13..31</b>	Reserved

#### A.5.7 DMA Channel 1 PCI Data Address Register (PCI Offset 0x98)

<b>D0..31</b>	PCI Data Address Register. This indicates where in the PCI memory space the DMA transfers (reads or writes) will start from.
---------------	--

#### A.5.8 DMA Channel 1 Local Data Address Register (PCI Offset 0x9C)

<b>D0..31</b>	Local data Address Register. This indicates where in the local memory space the DMA transfers (reads or writes) will start from.
---------------	--

#### A.5.9 DMA Channel 1 Transfer Size (bytes) register (PCI Offset 0xA0)

<b>D0..22</b>	DMA Transfer Size (Bytes). Indicates number of bytes to be transferred during DMA operation.
<b>D23..31</b>	Reserved

#### A.5.10 DMA Channel 1 Descriptor Pointer Register (PCI Offset 0xA4)

<b>D0</b>	<p>Descriptor Location.</p> <p>A 1 value indicates PCI address space.</p> <p>A 0 value indicates Local address space.</p>
<b>D1</b>	<p>End of Chain</p> <p>A 1 value indicates end of chain.</p> <p>A 0 value indicates not end of chain descriptor. (Same as Nonchaining Mode)</p>
<b>D2</b>	<p>Interrupt after Terminal Count</p> <p>A 1 value causes an interrupt to be generated after the terminal count for this descriptor is reached.</p> <p>A 0 value disables interrupts from being generated.</p>
<b>D3</b>	<p>Direction of transfer</p> <p>A 1 value indicates transfers from local bus to PCI bus.</p> <p>A 0 value indicates transfers from PCI bus to local bus.</p>
<b>D4..31</b>	Next Descriptor Address. Quad word aligned (bits [3:0] =0000).

### A.5.11 DMA Command/Status Register (PCI Offset 0xA8)

<b>D0</b>	Channel 0 Enable A 1 value enables the channel to transfer data. A 0 value disables the channel from starting a DMA transfer and if in the process of transferring data suspend transfer (Pause).
<b>D1</b>	Channel 0 Start Writing a 1 to this bit causes the channel to start transferring data if the channel is enabled.
<b>D2</b>	Channel 0 Abort Writing a 1 to this bit causes the channel to abort the current transfer. The channel enable bit must be cleared. The channel complete bit is set when the abort has completed.
<b>D3</b>	Clear Interrupt. Writing a 1 to this bit clears channel 0 interrupts.
<b>D4</b>	Channel 0 Done A 1 value indicates this channels transfer is complete. A 0 value indicates the channel transfer is not complete.
<b>D5..7</b>	User Defined

### A.5.12 DMA Channel 1 Command/Status Register 0 (PCI Offset 0xA8)

<b>D0</b>	Channel 1 Enable A 1 value enables the channel to transfer data. A 0 value disables the channel from starting a DMA transfer and if in the process of transferring data suspend transfer (Pause).
<b>D1</b>	Channel 1 Start Writing a 1 to this bit causes the channel to start transferring data if the channel is enabled.
<b>D2</b>	Channel 1 Abort Writing a 1 to this bit causes the channel to abort the current transfer. The channel enable bit must be cleared. The channel complete bit is set when the abort has completed.
<b>D3</b>	Clear Interrupt. Writing a 1 to this bit clears channel 1 interrupts.
<b>D4</b>	Channel 1 Done A 1 value indicates this channel's transfer is complete. A 0 value indicates the channel transfer is not complete.
<b>D5..7</b>	Reserved

### A.5.13 DMA Arbitration Register 1 (PCI Offset 0xAC)

<b>D0..7</b>	Local bus Latency Timer Number of local bus clock cycles before negating HOLD and releasing the local bus. This timer is also used with bit 27 to delay BREQ input to give up the local bus only when this timer expires.
<b>D8..15</b>	Local bus Pause Timer Number of local bus clock cycles before reasserting HOLD after releasing the local bus. <b>Note:</b> <i>Applicable only to DMA operation.</i>
<b>D16</b>	Local bus Latency Timer Enable A value of 1 enables latency timer.
<b>D17</b>	Local bus Pause Timer Enable A value of 1 enables pause timer.
<b>D18</b>	Local bus BREQ Enable A value of 1 enables local bus BREQ input. When the BREQ input is active, PCI 9080 negates HOLD and releases the local bus.
<b>D19..20</b>	DMA Channel Priority A value of 00 indicates a rotational priority scheme. A value of 01 indicates Channel 0 has priority. A value of 10 indicates Channel 1 has priority. A value of 11 is reserved.
<b>D21</b>	Local bus direct slave give up bus mode When set to 1, PCI 9080 negates HOLD and releases the local bus when the Direct Slave write FIFO becomes empty during a Direct Slave write or when the Direct Slave read FIFO becomes full during a Direct Slave read.
<b>D22</b>	Direct slave LLOCKo# Enable A value of 1 enables PCI Direct Slave locked sequences. A value of 0 disables Direct Slave locked sequences.
<b>D23</b>	PCI Request Mode A value of 1 causes PCI9080 to negate REQ when it asserts FRAME during a master cycle. A value of 0 causes PCI 9080 to leave REQ asserted for the entire bus master cycle.
<b>D24</b>	PCI Rev 2.1 Mode When set to 1, PCI 9080 operates in Delayed Transaction mode for Direct Slave Reads. PCI 9080 issues a RETRY and prefetches the read data.
<b>D25</b>	PCI Read No Write Mode A value of 1 forces a retry on writes if read is pending. A value of 0 allows writes to occur while read is pending.
<b>D26</b>	PCI Read with Write Flush Mode A value of 1 submits a request to flush a pending read cycle if a write cycle is detected. A value of 0 submits a request to not effect pending reads when a write cycle occurs. (PCI v2.1 compatible)
<b>D27</b>	Gate the Local Bus Latency Timer with BREQ If this bit is set to 0, PCI 9080 gives up the local bus during Direct Slave or DMA transfer after the current cycle (if enabled and BREQ is sampled). If this bit is set to 1, PCI 9080 gives up the local bus only (if BREQ is sampled) and the Local Bus Latency Timer is enabled and expires during Direct Slave or DMA transfer.
<b>D28</b>	PCI Read No Flush Mode A value of 1 submits request to not flush the read FIFO if PCI read cycle completes (Read Ahead mode). A value of 0 submits request to flush read FIFO if PCI read cycle completes.

<b>D29</b>	Reads Device or Vendor ID If set to 0, reads from the PCI Configuration Register address 0x00 and returns the Device ID and Vendor ID. If set to 1, reads from the PCI Configuration Register address 0x00 and returns the Subsystem and Subsystem Vendor ID.
<b>D30..31</b>	Reserved

#### A.5.14 DMA Threshold Register 1 (PCI Offset 0xB0)

<b>D0..3</b>	DMA Channel 0 PCI to Local Almost Full (C0PLAF): # of Full Entries (minus 1) in FIFO before Requesting Local Bus for Writes. (C0PLAF+1) + (C0PLAE+1) should be <= FIFO Depth of 16
<b>D4..7</b>	DMA Channel 0 Local to PCI Almost Empty (C0LPAE): # of Empty Entries (minus 1) in FIFO before Requesting Local Bus for Reads. (C0LPAF+1) + (C0LPAE+1) should be <= FIFO Depth of 16
<b>D8..11</b>	DMA Channel 0 Local to PCI Almost Full (C0LPAF): # of Full Entries (minus 1) in FIFO before requesting PCI bus for Writes.
<b>D12..15</b>	DMA Channel 0 PCI to Local Almost Empty (C0PLAE): # of Empty Entries (minus 1) in FIFO before Requesting PCI Bus for Reads.
<b>D16..19</b>	DMA Channel 1 PCI to Local Almost Full (C0PLAF): # of Full Entries (minus 1) in FIFO before Requesting Local Bus for Writes. (C0PLAF+1) + (C0PLAE+1) should be <= FIFO Depth of 16
<b>D20..23</b>	DMA Channel 1 Local to PCI Almost Empty (C0LPAE): # of Empty Entries (minus 1) in FIFO before Requesting Local Bus for Reads. (C0LPAF+1) + (C0LPAE+1) should be <= FIFO Depth of 16
<b>D24..27</b>	DMA Channel 1 Local to PCI Almost Full (C0LPAF): # of Full Entries (minus 1) in FIFO before requesting PCI bus for Writes.
<b>D28..31</b>	DMA Channel 1 PCI to Local Almost Empty (C0PLAE): # of Empty Entries (minus 1) in FIFO before Requesting PCI Bus for Reads.

## A.6 MESSAGING QUEUE REGISTERS

Table G MESSAGING QUEUE REGISTERS

PCI Offset from base Addr	Size	Access		Register Name	Value after reset
		R	W		
0x30	D32	yes	yes	Outbound Post Queue Interrupt Status	0x00000000
0x34	D32	yes	yes	Outbound Post Queue Interrupt Mask	0x00000000
0x40	D32	yes	yes	Inbound Queue Port	0x00000000
0x44	D32	yes	yes	Outbound Queue Port	0x00000000
0xC0	D32	yes	yes	Messaging Unit Configuration	0x00000000
0xC4	D32	yes	yes	Queue Base Address	0x00000000
0xC8	D32	yes	yes	Inbound Free Head Pointer	0x00000000
0xCC	D32	yes	yes	Inbound Free Trail Pointer	0x00000000
0xD0	D32	yes	yes	Inbound Post Head Pointer	0x00000000
0xD4	D32	yes	yes	Inbound Post Tail Pointer	0x00000000
0xD8	D32	yes	yes	Outbound Free Head Pointer	0x00000000
0xDC	D32	yes	yes	Outbound Free Tail Pointer	0x00000000
0xE0	D32	yes	yes	Outbound Post Head Pointer	0x00000000
0xE4	D32	yes	yes	Outbound Post Tail Pointer	0x00000000
0xE8	D32	yes	yes	Queue Status/Control Register	0x00000000

### A.6.1 Outbound Post List FIFO Interrupt Status Register (PCI Offset 0x30)

<b>D0..2</b>	Reserved
<b>D3</b>	Outbound Post List FIFO Interrupt. This bit is set when the Outbound Post List FIFO is not empty. This bit is not affected by the interrupt mask bit.
<b>D4..31</b>	Reserved

### A.6.2 Outbound Post List FIFO Interrupt Status Register (PCI Offset 0x34)

<b>D0..2</b>	Reserved
<b>D3</b>	Outbound Post List FIFO Interrupt Mask. Interrupt is masked when this bit is set.
<b>D4..31</b>	Reserved

### A.6.3 Inbound Queue Port Register (PCI Offset 0x40)

<b>D0..31</b>	<p>Value written by PCI master is stored into the Inbound Post List FIFO, which is located in local memory at the address pointed to by the Queue Base Address + FIFO Size + Inbound Post Head Pointer. From the time of the PCI write until the local memory write and update of the Inbound Post Queue Head Pointer, further accesses to this register result in a retry. A local interrupt is generated when the Inbound Post List FIFO is not empty.</p> <p>When the port is read by the PCI master, the value is read from the Inbound Free List FIFO, which is located in local memory at the address pointed the by The Queue Base Address + Inbound Free Tail Pointer. If FIFO is empty, a value of FFFFFFFFh is returned..</p>
---------------	---

### A.6.4 Outbound Queue Port Register (PCI Offset 0x44)

<b>D0..31</b>	<p>Value written by PCI master is stored into the Outbound Free List FIFO, which is located in local memory at the address pointed to by the Queue Base Address + (3*FIFO Size) + Outbound Free Head Pointer. From the time of the PCI write until the local memory write and update of the Outbound Free Head Pointer, further accesses to this register result in a retry. If FIFO fills up, a local LSERR interrupt is generated.</p> <p>When the port is read by the PCI master, the value is read from the Outbound Post List FIFO, which is located in local memory at the address pointed to by the Queue Base address + (2*FIFO Size) + Outbound Trail Pointer. If FIFO is empty, a value of FFFFFFFFh is returned. A PCI interrupt is generated if Outbound Post List FIFO is not empty.</p>
---------------	---

### A.6.5 Messaging Queue Configuration Register (PCI Offset 0xC0)

<b>D0</b>	<p><b>Queue Enable</b> Value of 1 allows accesses to the Inbound and Outbound Queue ports. If cleared to 0, writes are accepted but ignored and reads return FFFFFFFF. All pointer initialization and frame allocation should be completed before enabling this bit.</p>
<b>D1..5</b>	<p><b>Circular FIFO Size</b> Defines the size of one of the circular FIFOs. Each of the four FIFOs are the same size. Each FIFO entry is one 32 bit word.</p>

FIFO Size Encoding

	Max entries per FIFO	FIFO Size	Total FIFO Memory
00001	4K entries	16 KB	64 KB
00010	8K entries	32 KB	128 KB
00100	16K entries	64 KB	256 KB
01000	32K entries	128 KB	512 KB
10000	64K entries	256 KB	1 MB

<b>D6..31</b>	Reserved
---------------	----------

### A.6.6 Queue Base Address Register (PCI Offset 0xC4)

<b>D0..19</b>	Reserved
<b>D20..31</b>	<p><b>Queue Base Address</b> Local Memory base address of the Inbound and Outbound Queues (four contiguous and equal size FIFOs). Queue base address must be aligned on a 1 MB boundary.</p>

### A.6.7 Inbound Free Head Pointer Register (PCI Offset 0xC8)

<b>D0..1</b>	Reserved
<b>D2..19</b>	Inbound Free Head Pointer Local memory Offset for Inbound Free List FIFO. This register is initialized as (0*FIFO Size) and maintained by the local CPU software.
<b>D20..31</b>	Queue Base Address

### A.6.8 Inbound Free Head Tail Register (PCI Offset 0xCC)

<b>D0..1</b>	Reserved
<b>D2..19</b>	Inbound Free Tail Pointer Local Memory Offset for Inbound Free List FIFO. This register is initialized as (0*FIFO Size) by the local CPU software. It is maintained by the MU hardware and is incremented modulo the FIFO size.
<b>D20..31</b>	Queue Base Address

### A.6.9 Inbound Post Head Pointer Register (PCI Offset 0xD0)

<b>D0..1</b>	Reserved
<b>D2..19</b>	Inbound Post Head Pointer Local Memory Offset for Inbound Post List FIFO. This register is initialized as (1*FIFO Size) by the local CPU software. It is maintained by the MU hardware and is incremented modulo the FIFO size.
<b>D20..31</b>	Queue Base Address

### A.6.10 Inbound Post Tail Pointer Register (PCI Offset 0xD4)

<b>D0..1</b>	Reserved
<b>D2..19</b>	Inbound Post Tail Pointer Local Memory Offset for Inbound Post List FIFO. This register is initialized as (1*FIFO Size) by the local CPU software.
<b>D20..31</b>	Queue Base Address

### A.6.11 Outbound Free Head Pointer Register (PCI Offset 0xD8)

<b>D0..1</b>	Reserved
<b>D2..19</b>	Outbound Free Head Pointer Local Memory Offset for Outbound Free List FIFO. This register is initialized as (3*FIFO Size) by the local CPU software. It is maintained by the MU hardware and is incremented modulo the FIFO size.
<b>D20..31</b>	Queue Base Address

### A.6.12 Outbound Free Tail Pointer Register (PCI Offset 0xDC)

<b>D0..1</b>	Reserved
<b>D2..19</b>	Outbound Free Tail Pointer Local Memory Offset for Outbound Free List FIFO. This register is initialized as (3*FIFO Size) by the local CPU software.
<b>D20..31</b>	Queue Base Address

### A.6.13 Outbound Post Head Pointer Register (PCI Offset 0xE0)

<b>D0..1</b>	Reserved
<b>D2..19</b>	Outbound Post Head Pointer Local Memory Offset for Outbound Post List FIFO. This register is initialized as (2*FIFO Size) by the local CPU software.
<b>D20..31</b>	Queue Base Address

### A.6.14 Outbound Post Tail Pointer Register (PCI Offset 0xE4)

<b>D0..1</b>	Reserved
<b>D2..19</b>	Outbound Post Tail Pointer Local Memory Offset for Outbound Post List FIFO. This register is initialized as (2*FIFO Size) and maintained by the MU hardware and is incremented modulo the FIFO size.
<b>D20..31</b>	Queue Base Address

### A.6.15 Queue Status/Control Register (PCI Offset 0xE8)

<b>D0</b>	I(2)O Decode Enable When this bit is set, Mailbox registers 0 and 1 are replaced by the Inbound and Outbound Queue Port Registers and redefines Space 1 as PCI Base Address 0 to be accessed by PCIBAR0. Former Space 1 registers F0, F4, and F8 should be programmed to configure their shared I(2)O memory space, defined as PCI Base Address 0.
<b>D1</b>	Queue Local Space Select When this bit is set to 0, use Local Address Space 0 bus region descriptor for queue accesses. When this bit is set to 1, use Local Address Space 1 bus region descriptor for queue accesses.
<b>D2</b>	Outbound Post List FIFO Prefetch Enable When this bit is set, prefetching occurs from the Outbound Post List FIFO if not empty.
<b>D3</b>	Inbound Free List FIFO Prefetch Enable When this bit is set, prefetching occurs from the Inbound Free List FIFO if not empty.
<b>D4</b>	Inbound Post List FIFO Interrupt Mask When this bit is set, interrupt is masked.
<b>D5</b>	Inbound Post List FIFO Interrupt This bit is set when the Inbound Post List FIFO if not empty. This bit is not affected by the Interrupt Mask bit.
<b>D6</b>	Outbound Free List FIFO Overflow Interrupt Mask When this bit is set, interrupt is masked
<b>D7</b>	Outbound Free List FIFO Overflow Interrupt This bit is set when the Outbound Free List FIFO becomes full. A local SERR (NMI) interrupt is generated if enabled in the Interrupt Control/Status Register. Writing 1 clears the interrupt.
<b>D8..31</b>	Unused