

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

PCI64-HPDI32A

32-Bit
High-Speed Parallel
Digital Interface

User Manual

Preliminary

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

PREFACE

General Standards Corporation
Copyright (C) 2002 **General Standards Corporation**

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

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

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

General Standards Corporation 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 Corporation.**

RELATED PUBLICATIONS

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

PLX PCI 9656 Data Book

PLX Technology Inc.
390 Potrero Avenue
Sunnyvale, CA 4085
(408) 774-3735
http://www.plxtech.com/

EIA-422-A – Electrical Characteristics of Balanced Voltage Digital Interface Circuits
(EIA order number EIA-RS-422A)

EIA-485 – Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems
(EIA order number EIA-RS-485)

EIA Standards and Publications can be purchased from:

GLOBAL ENGINEERING DOCUMENTS
15 Inverness Way East
Englewood, CO 80112
Phone: (800) 854-7179
http://global.ihs.com/

IEEE P1386 - Standard Mechanic for a Common Mezzanine Card Family: CMC

IEEE P1386.1 - Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC
Sponsored by the Microprocessor & Microcomputer Standards Committee (MMSC) of the IEEE Computer Society

Copies of IEEE specifications available from:

Institute of Electrical and Electronics Engineers
Service Center
445 Hoes Lane
Piscataway, NJ 08855-1331 USA
http://www.ieee.org/

PCI Local Bus Specification Revision 2.1 June 1, 1995.

Copies of PCI specifications available from:

PCI Special Interest Group
NE 2575 Kathryn Street, #17
Hillsboro, OR 97124
http://www.pcisig.com/

TABLE of CONTENTS

CHAPTER 1: INTRODUCTION 4

1.0 FUNCTIONAL DESCRIPTION 4
Figure 1.0: HPDI32 Block Diagram 4
1.1 CABLE INTERFACE 6
Figure 1.1: HPDI32 Cable Interface 6
1.2 FIFOS 6

CHAPTER 2: PROGRAMMING 7

2.0 INITIALIZATION 7
2.1 RESETS 7
2.2 FIFOS 7
2.3 INTERRUPTS 8
2.4 DMA 8
Table 2.4: 7
2.4.1 PIO MODE 7
2.4.2 NON-DEMAND DMA 9
2.4.3 DEMAND MODE DMA 9
2.4.4 DMA DATA PACKING 9
2.5 CABLE INTERFACE SIGNALS 9
FIGURE 2.1: CABLE INTERFACE TIMING 11
2.6 GENERAL PURPOSE I/O 11

CHAPTER 3: LOCAL SPACE REGISTERS 12

3.0 LOCAL REGISTERS 12
Table 3.0: Local Register Map 10
3.1 FIRMWARE REVISION REGISTER 12
3.2 BOARD CONTROL REGISTER 13
3.3 BOARD STATUS REGISTER 14
3.4 TX ALMOST FLAG REGISTER 15
3.5 RX ALMOST FLAG REGISTER 15
3.6 FEATURES REGISTER 15
3.7 TX FIFO/RX FIFO 16
3.8 TX STATUS LENGTH COUNT 16
3.9 TX LINE VALID LENGTH COUNT 16
3.10 TX INVALID LENGTH COUNT 16
3.11 RX STATUS LENGTH COUNTER 17
3.12 RX LINE LENGTH COUNTER 17
3.13 INTERRUPT CONTROL REGISTER 17
3.14 INTERRUPT STATUS REGISTER 18
3.15 TRANSMIT CLOCK DIVIDER REGISTER 18
3.16 TRANSMIT FIFO SIZE 18
3.17 RECEIVE FIFO SIZE 18
3.18 TRANSMIT FIFO WORDS 18
3.19 RECEIVE FIFO WORDS 19
3.20 INTERRUPT EDGE/LEVEL REGISTER 19
3.21 INTERRUPT HI/LO REGISTER 19

CHAPTER 4: PCI INTERFACE 20

4.0 PCI INTERFACE REGISTERS 20

4.1 PCI CONFIGURATION REGISTERS 20
Table 4.1: PCI Configuration Registers 20
4.1.1 PCI CONFIGURATION ID REGISTER 21
4.1.2 PCI COMMAND REGISTER 21
4.1.3 PCI STATUS REGISTER 21
4.1.4 PCI REVISION ID REGISTER 22
4.1.5 PCI CLASS CODE REGISTER 22
4.1.6 PCI CACHE LINE SIZE REGISTER 23
4.1.7 PCI LATENCY TIMER REGISTER 23
4.1.8 PCI HEADER TYPE REGISTER 23
4.1.9 PCI BASE ADDRESS REGISTER FOR MEMORY ACCESS TO LOCAL/RUNTIME/DMA REGISTERS 23
4.1.10 PCI BASE ADDRESS REGISTER FOR I/O ACCESS TO LOCAL/RUNTIME/DMA REGISTERS 23
4.1.11 PCI BASE ADDRESS REGISTER FOR MEMORY ACCESS TO LOCAL ADDRESS SPACE 0 24
4.1.12 PCI SUBSYSTEM DEVICE/VENDOR ID REGISTER 24
4.1.13 PCI INTERRUPT LINE REGISTER 24
4.1.14 PCI INTERRUPT PIN REGISTER 24
4.1.15 PCI MIN_GNT REGISTER 24
4.1.16 PCI MAX_LAT REGISTER 24
4.2 LOCAL CONFIGURATION REGISTERS 26
Table 4.2: Local Configuration Registers 26
4.1.1 LOCAL ADDRESS SPACE 0 RANGE REGISTER FOR PCI TO LOCAL BUS 26
4.2.2 MODE/ARBITRATION REGISTER 27
4.2.3 BIG/LITTLE ENDIAN DESCRIPTOR REGISTER 27
4.2.4 LOCAL ADDRESS SPACE 0/EXPANSION ROM BUS REGION DESCRIPTOR REGISTER 27
4.3 RUNTIME REGISTERS 28
Table 4.3: Runtime Registers 28
4.3.1 INTERRUPT CONTROL /STATUS 28
4.3.2 SERIAL EEPROM CONTROL, PCI COMMAND CODES, USER I/O CONTROL, INIT CONTROL REGISTER 29
4.3.3 PCI PERMANENT CONFIGURATION ID REGISTER 30
4.3.4 PCI PERMANENT REVISION ID REGISTER 30
4.4 LOCAL DMA REGISTERS 31
Table 4.4: DMA Registers 31
4.4.1 DMA CHANNEL 0 MODE REGISTER 31
4.4.2 DMA CHANNEL 0 PCI ADDRESS REGISTER 32
4.4.3 DMA CHANNEL 0 LOCAL ADDRESS REGISTER 32
4.4.4 DMA CHANNEL 0 TRANSFER SIZE (BYTES) REGISTER 33
4.4.5 DMA CHANNEL 0 DESCRIPTOR POINTER REGISTER 33
4.4.6 DMA CHANNEL 0 COMMAND/STATUS REGISTER 33
4.4.7 DMA ARBITRATION REGISTER 33
4.4.8 DMA THRESHOLD REGISTER 33
4.5 MESSAGING QUEUE REGISTERS 33

CHAPTER 5: HARDWARE CONFIGURATION 35

5.0 HARDWARE JUMPERS (J1) 35
5.1 CABLE INTERFACE CONNECTOR 36
Table 5.0: Cable Pin-Out 36

CHAPTER 6: ORDERING INFORMATION 38

6.0 ORDERING INFORMATION 38
6.0.1 BUS INTERFACE 38
6.0.2 FIFO SIZE 38
6.1 CUSTOM APPLICATIONS 38

CHAPTER 1: INTRODUCTION

1.0 FUNCTIONAL DESCRIPTION

The PCI64-HPDI32A Board is a high-speed 32-bit/64-bit parallel digital interface card capable of transmitting or receiving data at up to 80 Mbytes per second over a LVDS interface. An optional PECL user interface is available. On-board transmit and receive FIFOs of up to 128k words deep buffer transfer data between the PCI/PMC bus and the cable interface. This allows the HPDI32A to maintain maximum bursts on the cable interface (at least up to the depth of the FIFOs) independent of the PCI bus interface. The on-board FIFOs can also used to buffer data between the cable interface and the PCI bus to maintain a sustained data throughput for real-time applications.

The HPDI32A interface is a half-duplex interface (board is either transmitting or receiving data, not simultaneous). The HPDI32A is easily set up to transfer data by initializing a few local registers. Once the data link is established, the data is transferred to/from the user application by simply reading or writing the on-board FIFOs. The board has an advanced PCI interface engine, which provides for increased data throughput via DMA. The board will support both standard and chained (scatter/gather) DMA.

The cable interface provides a flexible interface suited to most high speed parallel applications. All data to/from the board is synchronous with the transmit clock supplied by the transmitter. A Frame Valid signal indicates data is present on the interface, and programmable Line Valid and Status Valid signals provide additional interface capability. The interface also provides for data throttling by the receiving device – the receiver can hold off data the transmitting device until ready to receive. The HPDI32A interface is further programmable to allow the user to disable most of these standard interface functions and use the interface signals as discrete IO, including external interrupts.

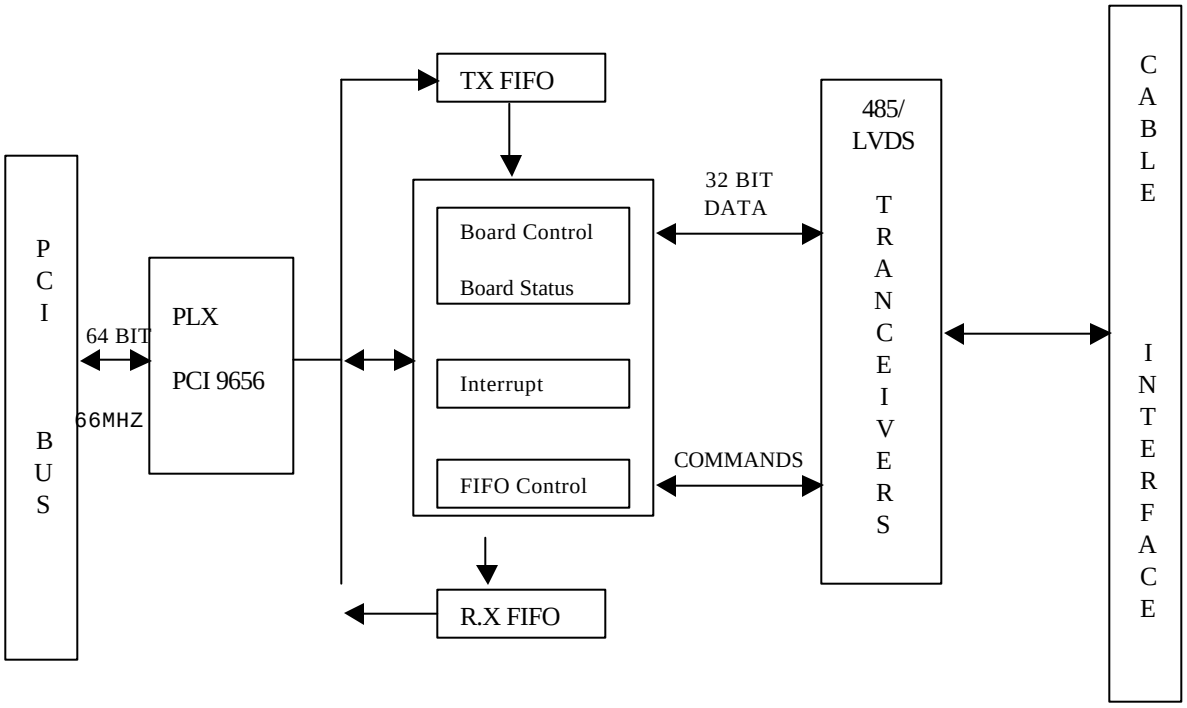


Figure 1.0: HPDI32 Block Diagram

1.1 CABLE INTERFACE

The cable interface consists of 32 bits of data, one clock, and 7 bi-directional signals. All cable interface signals are either differential RS485/RS422 or differential PECL, depending upon the version of the board. The clock and 7 bi-directional signals are used for controlling the data transfer and other pre-defined functions. Six of these signals may also be used for user IO or interrupt sources if the default function is not used.

Figure 1.1 shows the default HPDI32A interface control signals. All control signals and data are synchronous to the cable interface clock supplied by the transmitting device. The on-board transmit clock may be up to 20MHz. A Frame Valid signal indicates valid data is being transferred. Two optional programmable signals, Line Valid and Status Valid, can be used to further qualify the data. The Status Valid signal is used to flag a programmable number of words at the start of the cycle as status words. The Line Valid signal can be used to signify valid data within a frame. If the Line Valid signal is used, data is only recorded when the Line Valid signal is asserted. The standard cable interface also provides for a cable-throttling signal to pause data transmission. This signal is driven by the receiving device to indicate it is capable of receiving data. Section 2.5 provides a more detailed cable interface description.

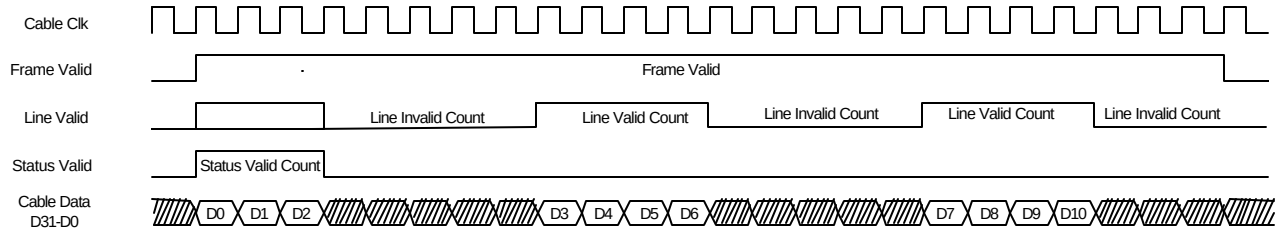


Figure 1.1: HPDI32A Cable Interface

1.2 FIFOs

The FIFOs on the PCI64-HPDI32A are used for buffering the transmit or receive data. This allows the data on cable interface to run independent of the PCI interface. There are two sets of FIFOs on the board: a set of four FIFOs for transmit data, and a second set of four for receive data. Each set consists of 32 bits of data and 4 status flags. 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. Four status flags accompany each set the FIFOs: Empty, Almost Empty, Almost Full, and Full. The Almost Empty and the Almost Full status flags can be programmed via software to assert most desired levels within the FIFOs. These programmable flags can be used for DMA control, throttling cable data, or to indicate when a desired amount of data has been received.

CHAPTER 2: PROGRAMMING

2.0 INITIALIZATION

Several functions on the HPDI32A board will generally be unchanged in a given application. These include interrupt setup, FIFO Almost Flag Values, and General Purpose IO direction setup. Therefore, initializing these functions board will generally need to be done only once by the software. However, if a Board Reset is performed, all registers will return to their default values. Software must reinitialize the board following a Board Reset.

2.1 RESETS

There are three bits in this Board Control Register that are used as resets to the local logic functions. These bits perform a reset when the software writes a '1'. The software does not need to clear the bits following a Reset since the reset bits are self-clearing

Board Reset	Board Control Register D0 – Setting this bit will reset the local logic, reset (clear) the FIFOs and program the default Almost Flag Values, and place all registers into a known state.
Tx FIFO Reset	Board Control Register D1 – Setting this bit will clear the Tx FIFOs and program the current Almost Flag Values.
Rx FIFO Reset	Board Control Register D2 – Setting this bit will clear the Rx FIFOs and program the current Almost Flag Values.

2.2 FIFOs

The FIFOs on the PCI64-HPDI32A are used for buffering the transmit or receive data. This allows the data on cable interface to run independent of the PCI interface, ensuring data will be transferred on the cable regardless of software overhead to the board (assuming the average receive throughput can be maintained). The board allows for different FIFO depths to be installed from 8k to 128k (32k standard). This allows the user to customize the board for specific real-time applications. The FIFO depth must be specified when ordering – consult factory for further information.

There are two sets of FIFOs on the board – one for transmit and one for receive. Each set has a width of 32 bits of data and provides four FIFO status flags. These flags indicate Empty, Full, Almost Empty, and Almost Full. Two of these flags, Almost Empty and Almost Full are user programmable. These programmable flags are used for Demand Mode DMA (see Section 3.3), or may be used for cable throttling (pause transmitter when Rx FIFO Almost Full) or to provide the user with a trigger at a specific FIFO level. The Almost Flags of the FIFOs are programmed with the Almost Register values during a FIFO Reset (See Resets, Section 3.1).

NOTE: The Almost Empty Flag value represents the number of words from empty. The Almost Full Flag value represents the number of words from Full –1, not the number of words from empty.

The board also provides FIFO Size registers for both transmit and receive FIFOs. These registers may be useful if writing software (especially drivers) which needs to support multiple FIFO depth. The board also tracks the number of words currently in each FIFO. This count may be useful when accessing the FIFOs to prevent underruns or overflows.

2.3 INTERRUPTS

The HPDI32A contains many possible interrupt sources. The PLX 9656 PCI Interface chip has several interrupt sources (DMA done is the most useful), and the local HPDI32 firmware has 16 potential interrupt sources. All local interrupts are passed through the PLX chip and will be requested on PCI INTA. In order for this board to generate interrupts to the PCI bus, they must be enabled in the PLX chip. Bits 8, 11, and 16 of the PLX Interrupt Control/Status Register must be set to a '1' in order for the interrupts to occur. The device driver typically handles enabling board interrupts, so the average user should not be concerned with accessing the PLX registers.

The 16 local interrupt sources allow for interrupts on Frame Start, Frame End, six cable inputs, and all FIFO flags. All interrupt sources are can be individually enabled, allowing the user 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 Rx FIFO Almost Empty Interrupt, the software will need to write a '1' to bit 13 of the ICR. This bit will should remain set until the need to disable this specific interrupt.

Each interrupt source is individually programmable as Rising Edge, Falling Edge, Level High, or Level Low triggering. All interrupts default to edge triggered to prevent a potential interrupt lockup seen with level interrupts (the level interrupt may keep re-interrupting indefinitely). Since interrupts are latched, edge triggered interrupts are more user friendly. The interrupts default at reset to the most common configuration, but this configuration may not suit all users. For example, one may wish to interrupt when the Tx FIFO is almost full to stop writing data to the FIFO, while another may wish to interrupt when it is not almost full to know when to resume writing to the FIFO. The HPDI32A provides the user the flexibility of interrupt configuration.

Interrupts are latched in the Interrupt Status Register (ISR). Since all interrupts are multiplexed onto a single interrupt request, the ISR provides the means of determining the unique interrupt source. The user should clear the interrupt by writing the specific bit back to the ISR as part of the interrupt service routine. From the previous example, when the interrupt has occurred, 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 user clears the interrupt. To re-enable the FIFO Almost Empty interrupt, the user must clear the interrupt by writing a '1' to bit 13 of the Interrupt Status Register.

2.4 DMA

Although the 33MHz PMC/PCI bus is capable of burst transfers up to 132Mbytes per sec, actual sustained throughput on the PCI bus will be much lower. The 66MHz PMC/PCI bus, capable of burst transfers up to 264MHz, will also sustain lower actual throughput on the PCI bus. This is due to many factors such as bus overhead, operating system overhead, application overhead, and possibly data storage overhead such as hard disk drive accesses. Since sustained PCI data rates will be typically slower than the maximum cable interface rate, DMA on the PCI bus is supported to make the PCI data transfers as fast as possible.

There are 3 methods the software application can move data to and from the HPDI32 board: PIO mode, Non-Demand DMA, and Demand Mode DMA. The two DMA modes are only supported to the on-board data FIFOs (DMA accesses to local registers are not supported, see table 2.4)

2.4.1 PIO MODE

In PIO mode, the user accesses the FIFOs through is single register reads and writes to the board. This is the slowest data transfer mode. In PIO mode, the user must check to make sure the Tx FIFO is not full prior to writing the Tx FIFOs, and check to ensure the Rx FIFO contains data before reading. The device driver should normally handle checking of this status prior to reading/writing the FIFO. See specific driver information for further details.

2.4.2 NON-DEMAND DMA

In Non-Demand DMA mode, the user specifies a DMA transfer size and initiates the transfer. Since there is no checking for data validity (no check to determine if Tx FIFO Full or Rx FIFO empty), the user must ensure the FIFOs can handle the entire transfer prior to transfer initiation. If the Tx FIFO becomes full or Rx FIFO Empty during a Non-Demand mode DMA transfer, the extra data will be discarded and no error indication is received – the DMA appears to have completed normally. Therefore, the user must use caution when using Non-Demand DMA.

2.4.3 DEMAND MODE DMA

Demand Mode DMA is similar to Non-Demand DMA, except the local logic will request the transfer based on FIFO status. After the user specifies a DMA transfer size and initiates the transfer, the local logic will request the data transfer if the Tx FIFO is not Full (transmit) or the Rx FIFO is not empty (Rx FIFO). For transmit, the logic will burst data into the FIFO until the Almost Full Flag is reached. The logic will then switch into a single transfer mode until the FIFO is filled (or transfer is complete). When the Tx FIFO becomes full, the logic will cease requesting data. When data is transmitted out of the FIFO (Tx FIFO no longer full), the request will again be asserted to re-fill the FIFO. This will resume until the transfer completes. When receive Demand DMA is initiated, the DMA is requested whenever data is in the Rx FIFO. On-board logic will run in burst mode until the Rx FIFO reaches the Almost Empty level, at which time the logic will switch into single word mode, where single words will be transferred until the FIFO goes empty. Once the Rx FIFO is empty, the logic ceases to request data until more data is received. The request is again asserted to empty the FIFO. This will continue until the transfer completes. Demand Mode DMA is the preferred DMA method.

2.4.4 DMA DATA PACKING

DMA also provides a means to pack 16-bit or 8-bit data into 32bit transfers. This is useful if only a 16-bit or 8-bit cable interface is used. In such cases, each data word uses the full 32-bit FIFO width (data is not packed in FIFO), but is packed for the PCI bus transfer. This will increase the effective PCI data throughput.

2.5 CABLE INTERFACE SIGNALS

Data is transferred to and from the HPDI32A board via an 80 pin cable interface consisting of 40 differential signals – 1 Clock, 7 Command /Control signals, and 32 Data bits. The seven Command/Control signals provide many cable protocol options. The default Cable Command signals are:

Command 0 – Frame Valid.	Provides an indication that data is currently being transferred on the cable. This signal is driven by the transmitter and must be asserted before data is recorded.
Command 1 – Line Valid.	Programmable signal to allow for multiple lines or rows within a frame. If Line Valid and Line Invalid counters are used during transmit, the signal will be and negated for Line Invalid Count and asserted for Line Valid Count, alternating for the entire frame. The Line Invalid Count begins after the Status Valid Count. Board Control Register Bit 6 controls the Line Valid state during the Status Word Count. If Line Valid is enabled but the Line Valid counter is zero (default), Line Valid will be asserted for the entire Frame. Line Valid may be disabled via the Board Control Register and used as discrete IO – GPIO0.
Command 2 – Status Valid.	Programmable signal to allow for status words at the start of frame. If Status Valid counters are used during transmit, the signal will be asserted for Status Valid Count at the beginning of every frame, and negated for the remainder of the frame. Status Valid may be disabled via the Board Control Register and used as discrete IO – GPIO1.
Command 3 – Rx Ready.	Provides a method for the receiving device to pause the data transfer. The receiving device drives this signal. This function is enabled by Board Control Bit D9. When the HPDI32A is in receive mode, this signal is negated when the Rx FIFO Almost Full flag is reached. Rx Ready may be disabled via the Board Control Register and used as discrete IO – GPIO2.
Command 4 – Tx Data Ready	Provides a signal to indicate data is present in the Tx FIFO. The transmitting device drives this signal Tx Data Ready may be disabled via the Board control register and used as discrete IO – GPIO3. This signal is not present in the PECL version of the card.
Command 5 – Tx Enabled	Provides a signal to indicate the HPDI32A is in transmit mode (Board Control Bit D4). If enabled, the HPDI32A card always drives this signal. Tx Enabled may be disabled via the Board Control Register and used as discrete IO – GPIO4.
Command 6 – Rx Enabled	Provides a signal to indicate the HPDI32A is in receive mode (Board Control Bit D5). If enabled, the HPDI32A card always drives this signal. Rx Enabled may be disabled via the Board Control Register and used as discrete IO – GPIO5.

Data is transferred using the Cable Clock. All transmit data and command/control signals are clocked on the rising edge of the TxCLK. The transmit clock on the HPDI32 is supplied by an on-board oscillator. This oscillator is socketed so the user can customize the clock interface speed. The board is shipped with a 20MHz oscillator standard, but may run up to 20MHz. To ensure maximum setup and hold times, all receive data and Command/Control signals are clocked on the falling edge of the RxClk. Figure 2.1 shows the data setup and hold times.

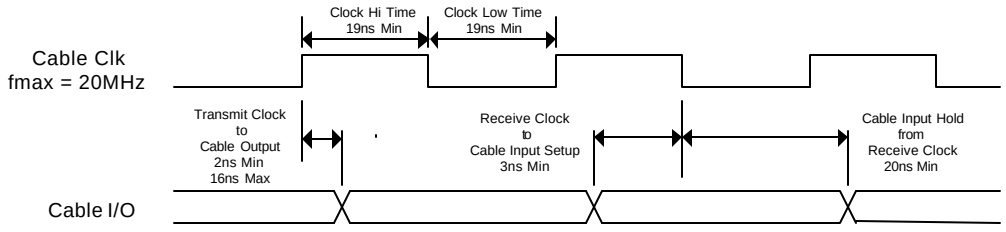


FIGURE 2.1: CABLE INTERFACE TIMING

2.6 GENERAL PURPOSE I/O

Since most users will not require all of the default Cable Command functions, six of the cable command signals can be changed from their default function and used as discrete IO. This allows users to control cable outputs, read cable inputs, or receive cable interrupts via simple software control. The General Purpose IO bits are controlled from the Board Control register. If a bit is set as a discrete output, the Board Control register also defines the output value. If a bit is setup as an input, the Cable Command state may be read through the Board Status register. To set a Cable Command as an interrupt source, set the signal as a discrete input, and setup the interrupt source via the interrupt registers.

The default configuration for the Cable Command bits is the factory defined cable functions. The user will need to re-initialize General Purpose IO setup following a board reset. If a Cable Command signal is changed from its default function, all on-board logic associated with that function is disabled.

CHAPTER 3: LOCAL SPACE REGISTERS

3.0 LOCAL REGISTERS

The Local Space registers control the transmission and reception of data to and from the board.

Table 3.0 Local Register Map

Offset Address	Size	Access*	Register Name	Value after Reset
0x00	D32	RO	Firmware Revision	0x800702XX
0x04	D32	RW	Board Control	0x00000000
0x08	D32	RC	Board Status	0x00X0CCXX
0x0C	D32	RW	Tx Programmable Almost	0x0010000F
0x10	D32	RW	Rx Programmable Almost	0x0010000F
0x14	D32	RO	Features	0x0000003F
0x18	D32	RW	Rx/Tx FIFOs	EMPTY
0x1C	D32	RW	Tx Status Length Count	0x00000000
0x20	D32	RW	Tx Line Valid Length Count	0x00000000
0x24	D32	RW	Tx Line Invalid Length Count	0x00000000
0x28	D32	RO	Rx Status Counter	0x00000000
0x2C	D32	RO	Rx Line Counter	0x00000000
0x30	D32	RW	Interrupt Control	0x00000000
0x34	D32	RC	Interrupt Status	0x00000000
0x38	D32	RW	Tx Clock Divider	0x00000000
0x3C			Reserved	
0x40	D32	RO	Tx FIFO Size	0x000XXX00
0x44	D32	RO	Rx FIFO Size	0x000XXX00
0x48	D32	RO	Tx FIFO Words	0x00000000
0x4C	D32	RO	Rx FIFO Words	0x00000000
0x50	D32	RW	Interrupt Edge/Level	0x0000FFFF
0x54	D32	RW	Interrupt Hi/Lo	0x0000CFFD

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

3.1 FIRMWARE REVISION REGISTER

(Offset 0x00000000)

This Register is used to determine the version of firmware that is programmed into the board. If the board logic is changed, the value in this register is changed.

D7:0 Firmware Revision – Incremented when firmware changes.
D15:8 PCB Revision – This field details the PCB version for this firmware version.
D23:16 Sub ID – This field reflects special variations of the HPDI32 board.
D30:24 Reserved

D31 Features Register present
A '1' indicates Features Register is implemented in specific firmware version.

3.2 BOARD CONTROL REGISTER

(Offset 0x00000004)

The Board Control Register is used to control miscellaneous functions on the HPDI32A. These include Resets, Enable data transmit/receive, and General Purpose IO.

D0 Board Reset
Writing a'1' to this bit will generate a pulse to reset the on-board logic and the FIFOs. This bit will clear automatically. The Rx/Tx FIFOs are also reset and reprogrammed with the default values following a board reset. To allow FIFO programming to occur, the user should wait about 10usec following Board Reset before the FIFOs are accessed.

D1 Tx FIFO Reset
Writing a '1' to this bit will generate a pulse that will reset the Tx FIFOs. Following the Reset, this bit will automatically clear itself. The Tx FIFO Reset also programs the Tx Almost Empty and Almost Full flags with the values contained in the Tx Almost Programming register (See Section 3.3).

D2 Rx FIFO Reset
Writing a '1' to this bit will generate a pulse that will reset the Rx FIFOs. Following the Reset, this bit will automatically clear itself. The Rx FIFO Reset also programs the Rx Almost Empty and Almost Full flags with the values contained in the Rx Almost Programming register (See Section 3.4).

D3 Reserved

D4 Transmit Enable
A '1' will enable data transmission from the Tx FIFOs to the Cable. When Transmit is enabled, the HPDI32A will drive Cable Data, TxClk, Frame Valid, Line Valid (if enabled), Status Valid (if enabled), and Tx Empty (if enabled).

D5 Receive Enable
A '1' will enable data reception into the Rx FIFOs from the Cable. When Receive enabled, the HPDI32A will drive Rx Ready (if enabled).

D6 Demand DMA Direction
A '1' will set the demand mode DMA direction to transmit. This bit is not used in half-duplex operation, but is added for future full-duplex operation

D7 Line Valid Hi On Status Valid Hi
A '1' will assert the Line Valid signal while Status Valid is asserted. This bit is unused if Line Valid is not used or Status Valid counter is not used.

D8 Start Transmit
A '1' will start data transmission from the Tx FIFO to the cable. This bit will automatically reset when the Tx FIFO is empty (no data to transmit).

D9 Cable Throttle Enable
A '1' will allow Rx Ready input (Tx Cable Command D3) to control the data transmission. If enabled (and transmit enabled), data will be transferred from the Tx FIFO to the cable as long as Rx Ready is asserted.

D16:D10 Reserved.

D17 Cable Command D1 Discrete Output
A '1' will set Cable Command D1 as a Discrete Output

D18 Cable Command D2 Discrete Output
A '1' will set Cable Command D2 as a Discrete Output

D19 Cable Command D3 Discrete Output
A '1' will set Cable Command D3 as a Discrete Output

D20 Cable Command D4 Discrete Output

A '1' will set Cable Command D4 as a Discrete Output

D21 Cable Command D5 Discrete Output
A '1' will set Cable Command D5 as a Discrete Output

D22 Cable Command D6 Discrete Output
A '1' will set Cable Command D6 as a Discrete Output

D24:D23 Reserved.

D25 Cable Command D1 Discrete Input/Output Value
If Cable Command D1 is set to a Discrete Output (D17=1), this bit determines the output value. Otherwise (D17=0), a '1' will set Cable Command D1 as a Discrete Input.

D26 Cable Command D2 Discrete Input/Output Value
If Cable Command D2 is set to a Discrete Output (D18=1), this bit determines the output value. Otherwise (D18=0), a '1' will set Cable Command D2 as a Discrete Input.

D27 Cable Command D3 Discrete Input/Output Value
If Cable Command D3 is set to a Discrete Output (D19=1), this bit determines the output value. Otherwise (D19=0), a '1' will set Cable Command D3 as a Discrete Input.

D28 Cable Command D4 Discrete Input/Output Value
If Cable Command D4 is set to a Discrete Output (D20=1), this bit determines the output value. Otherwise (D20=0), a '1' will set Cable Command D4 as a Discrete Input.

D29 Cable Command D5 Discrete Input/Output Value
If Cable Command D5 is set to a Discrete Output (D21=1), this bit determines the output value. Otherwise (D21=0), a '1' will set Cable Command D5 as a Discrete Input.

D30 Cable Command D6 Discrete Input/Output Value
If Cable Command D6 is set to a Discrete Output (D22=1), this bit determines the output value. Otherwise (D22=0), a '1' will set Cable Command D6 as a Discrete Input.

D31 Test Mode Enable (Factory Test Only)
A '1' will disable driving Tx Enabled (Cable Command D5) and Rx Enabled (Cable Command D6). This will prevent contention if two HPDI32A boards are cabled together.

3.3 BOARD STATUS REGISTER

(Offset 0x00000008)

The Board Status Register is used to check the most current status of on-board signals including the FIFO status flags and the Cable Command signals.

D0 Cable Command D0

D1 Cable Command D1

D2 Cable Command D2

D3 Cable Command D3

D4 Cable Command D4

D5 Cable Command D5

D6 Cable Command D6

D7 Start Transmit
A '1' indicates data transmission is in currently in progress

D8 Tx FIFO Empty Low
A '0' indicates the Tx FIFO is empty (Tx FIFO Empty Flag asserted).

D9 Tx FIFO Almost Empty Low
A '0' indicates the Tx FIFO is almost empty (Tx FIFO Almost Empty Flag asserted).

D10 Tx FIFO Almost Full Low
A '0' indicates the Tx FIFO is almost full (Tx FIFO Almost Full Flag asserted).

D11 Tx FIFO Full Low
A '0' indicates the Tx FIFO is full (Tx FIFO Full Flag asserted).

D12 Rx FIFO Empty Low

A ‘0’ indicates the Rx FIFO is empty (Rx FIFO Empty Flag asserted).

D13 Rx FIFO Almost Empty Low
A ‘0’ indicates the Rx FIFO is almost empty (Rx FIFO Almost Empty Flag asserted).

D14 Rx FIFO Almost Full Low
A ‘0’ indicates the Rx FIFO is almost full (Rx FIFO Almost Full Flag asserted).

D15 Rx FIFO Full Low
A ‘0’ indicates the Rx FIFO is full (Rx FIFO Full Flag asserted).

D16 Board Jumper 0 (PCI Only)
A ‘1’ will indicate the mini jumper on pins J2:1 to J2:2 is installed.

D17 Board Jumper 1 (PCI Only)
A ‘1’ will indicate the mini jumper on pins J2:3 to J2:4 is installed.
Note: Board jumpers can be used to distinguish between two HPDI32A cards in a system.

D20:18 Reserved

D21 Tx OverRun
A ‘1’ will indicate an attempt to write to the Rx FIFO when the FIFO was full has occurred. Since this bit is latched, it is cleared by writing a ‘1’ back to D21.

D22 Rx UnderRun
A ‘1’ will indicate an attempt to read from the Rx FIFO when the FIFO was empty has occurred. Since this bit is latched, it is cleared by writing a ‘1’ back to D22.

D23 Rx OverRun
A ‘1’ will indicate an attempt to write to the Rx FIFO when the FIFO was full has occurred. Since this bit is latched, it is cleared by writing a ‘1’ back to D23.

D24:31 Reserved

3.4 Tx ALMOST FLAG REGISTER

(Offset 0x0000000C)

This register is sets the programmed values for the Tx FIFO Almost Empty and Almost Full Flags. This value is programmed following a Tx FIFO Reset. A Board Reset will reset this register to the default value (0x0010000F) and then program this value.

D15:0 Almost Empty Flag Value

D31:16 Almost Full Flag Value (Number of available words remaining in FIFO – 1 when Flag asserted).

3.5 Rx ALMOST FLAG REGISTER

(Offset 0x00000010)

This register is sets the programmed values for the Rx FIFO Almost Empty and Almost Full Flags. This value is programmed following an Rx FIFO Reset. A Board Reset will reset this register to the default value (0x0010000F) and then program this value.

D15:0 Almost Empty Flag Value

D31:16 Almost Full Flag Value (Number of available words remaining in FIFO – 1 when Flag asserted).

3.6 FEATURES REGISTER

(Offset 0x00000014)

This Register indicates new features in each firmware version. This allows a driver to maintain compatibility across firmware and board revisions, while providing for new features to be added.

D0 Tx/Rx FIFO Size Registers Present
D1 Tx/Rx FIFO Words Registers Present
D2 Level/Edge Triggered Interrupts Supported
D3 General Purpose IO on Cable Command D6:2 Supported
D4 PLX DMA Channel 1 Supported
D5 Tx/Rx Underrun & Overrun flags
D31:6 Reserved

3.7 TX FIFO/RX FIFO
(Offset 0x00000018)

This register provides access to the transmit and receive FIFOs. Writing this register sets data for transmission, while reading gets incoming stored data from the Rx FIFO. The user must ensure that the Tx FIFO is not full when writing or data will be discarded. The user must also ensure that the Rx FIFO is not empty when reading or data returned will be indeterminate.

D31:0 FIFO Data

3.8 TX STATUS LENGTH COUNT
(Offset 0x0000001C)

This register holds the number of clocks the Status Valid signal will be asserted at the start of each transmit frame. This register is unused if Cable Command D2 is set as Discrete IO (instead of Status Valid).

D31:0 Tx Status Length Count Value

3.9 TX LINE VALID LENGTH COUNT
(Offset 0x00000020)

This register is contains the number of clocks that the Line Valid signal will be asserted after the Tx Line Invalid Length Counter has expired in a frame. The Line Valid signal will continue to alternate between the Tx Line Valid Length Counter and the Tx Line Invalid Length Counter until the end of the frame. This register is unused if Cable Command D1 is set as discrete IO (instead of Line Valid).

D31:0 Tx Line Valid Length Count Value

3.10 TX INVALID LENGTH COUNT
(Offset 0x00000024)

This register is contains the number of clocks that the Line Valid signal will be negated at the beginning of each transmit frame (following Status Valid count) or after the Tx Line Valid Length count has expired in a frame. The Line Valid signal will continue to pulse - alternating between the Tx Line Valid Length Counter and the Tx Line Invalid Length Counter until the end of the frame. This register is unused if Cable Command D1 is set as discrete IO (instead of Line Valid).

D15:0 Tx Line Invalid Length Count Value
D31:16 Reserved

3.11 RX STATUS LENGTH COUNTER
(Offset 0x00000028)

This register will count the number of Status Words (clocks where the Status Valid signal was asserted) received during the last frame. This register will reset at the start of the next frame.

If Cable Command D2 is set as Discrete IO (instead of Status Valid), this register will count the total number received words since Receive Enabled. This register will reset when receive is disabled.

D31:0 Rx Status Length Counter

3.12 RX LINE LENGTH COUNTER
(Offset 0x0000002C)

This register contains the number of Line Words (clocks where the Line Valid signal was asserted) during the last received frame. This register is unused if Cable Command D1 is set as discrete IO (instead of Line Valid).

D31:0 Rx Line Length Counter

3.13 INTERRUPT CONTROL REGISTER
(Offset 0x00000030)

The Interrupt Control Register enables the Local Interrupt Sources to generate a Local Interrupt request. See Section 3.3 for more detailed explanation of Interrupts.

D0 IRQ Source 0 - Enable Frame Valid (default rising edge – Frame Start)
D1 IRQ Source 1 - Enable Frame Valid (default falling edge - Frame End)
D2 IRQ Source 2 - Enable Cable Command D1 (default rising edge)
D3 IRQ Source 3 - Enable Cable Command D2 (default rising edge)
D4 IRQ Source 4 - Enable Cable Command D3 (default rising edge)
D5 IRQ Source 5 - Enable Cable Command D4 (default rising edge)
D6 IRQ Source 6 - Enable Cable Command D5 (default rising edge)
D7 IRQ Source 7 - Enable Cable Command D6 (default rising edge)
D8 IRQ Source 8 - Enable Tx FIFO Empty (default rising edge – Tx FIFO Empty)
D9 IRQ Source 9 - Enable Tx FIFO Almost Empty (default rising edge Tx FIFO Almost Empty)
D10 IRQ Source 10 - Enable Tx FIFO Almost Full (default rising edge – Tx FIFO Almost Full)
D11 IRQ Source 11 - Enable Tx FIFO Full (default rising edge – Tx FIFO Full)
D12 IRQ Source 12 - Enable Rx FIFO Empty (default falling edge – Rx FIFO Not Empty)
D13 IRQ Source 13 - Enable Rx FIFO Almost Empty (default falling edge – Rx FIFO Not Almost Empty)
D14 IRQ Source 14 - Enable Rx FIFO Almost Full (default rising edge – Rx FIFO Almost Full)
D15 IRQ Source 15 - Enable Rx FIFO Full (default rising edge – Rx FIFO Full)
D31:16 Reserved

3.14 INTERRUPT STATUS REGISTER
(Offset 0x00000034)

The Interrupt Status Register will provide interrupt status for each of the interrupt sources. If an interrupt is enabled in the Interrupt Control register, the interrupt will be latched in this register until cleared via writing a '1' back to the respective bit. If an interrupt is not enabled, each bit will represent the current state of the interrupt (although the interrupt will not be latched and will not generate a Local Interrupt request).

D15:0 IRQ 15:0 Status
D31:16 Reserved

3.15 TRANSMIT CLOCK DIVIDER REGISTER
(Offset 0x00000038)

This register will allow the On-board Transmit clock to be slowed based on the value in this register. The value in this register is the clock divider value. If this register contains 0 (default) or 1, the On-board clock is used as the Transmit clock.

D15:0 Tx Clock Divide Value
D31:16 Reserved

3.16 TRANSMIT FIFO SIZE
(Offset 0x00000040)

This register contains the Transmit FIFO depth. This is the true FIFO depth, regardless of the FIFO data width. This value is calculated once on power-up.

D19:0 Tx FIFO Depth
D31:0 Reserved

3.17 RECEIVE FIFO SIZE
(Offset 0x00000044)

This register contains the Receive FIFO depth. This is the true FIFO depth, regardless of the FIFO data width. This value is calculated once on power-up.

D19:0 Rx FIFO Depth
D31:0 Reserved

3.18 TRANSMIT FIFO WORDS
(Offset 0x00000048)

This register will track the current number of words in the Transmit FIFO.

D19:0 Current Number of Words in Tx FIFO
D31:0 Reserved

3.19 RECEIVE FIFO WORDS
(Offset 0x0000004C)

This register will track the current number of words in the Receive FIFO.

D19:0 Current Number of Words in Rx FIFO
D31:0 Reserved

3.20 INTERRUPT EDGE/LEVEL REGISTER
(Offset 0x00000050)

This register, along with the Interrupt Hi/Lo Register, defines the interrupt source as Level Hi, Level Lo, Rising Edge, or Falling Edge triggered. A '1' in each respective bit will set the interrupt source to edge triggered; a '0' will set the interrupt source to level triggered.

D15:0 IRQ 15:0 Edge Trigger Enable
D31:16 Reserved

3.21 INTERRUPT HI/LO REGISTER
(Offset 0x00000054)

This register, along with the Interrupt Edge/Level Register, defines the interrupt source as Level Hi, Level Lo, Rising Edge, or Falling Edge triggered. A '1' in each respective bit will set the interrupt source to active Hi, a '0' will set the interrupt source to active Lo.

D15:0 IRQ 15:0 Active Hi Enable
D31:16 Reserved

CHAPTER 4: PCI INTERFACE

4.0 PCI INTERFACE REGISTERS

A PCI9656 I/O Accelerator from PLX Technology handles the PCI Interface. The PCI interface is compliant with the 5V, 33MHz PCI Specification 2.1. The PCI9656 provides dual DMA controllers for fast data transfers to and from the on-board FIFOs. Fast DMA burst accesses provide for a maximum burst throughput of 132MB/s to the PCI interface. To reduce CPU overhead during DMA transfers, the controller also implements Chained (Scatter/Gather) DMA, as well as Demand Mode DMA.

Since many features of the PCI9656 are not utilized in this design, it is beyond the scope of this document to duplicate the PCI9656 User's Manual. Only those features, which will clarify areas specific to the PCI64-HPDI32A, are detailed here. Please refer to the PCI9656 User's Manual (See Related Publications) for more detailed information. Note that the BIOS configuration and software driver will handle most of the PCI9656 interface. Unless the user is writing a device driver, the details of the PCI interface (Chapter 2) may be skipped.

4.1 PCI CONFIGURATION REGISTERS

The PCI device configuration for the PCI64-HPDI32A is fully PCI 2.1 compliant. Table 4.1 contains a list of the PCI configuration registers present in the PCI9656. An on-board configuration serial EEPROM initializes many of these registers.

Table 4.1: PCI Configuration Registers

PCI CFG Addr	Local Offset Addr	PCI/Local Writable	Register Name	Value after Reset
0x00	0x00	Local	Device ID/Vendor ID	0x965610B5
0x04	0x04	Y	Status/Command	0x02800017
0x08	0x08	Local	Class Code/Revision ID	0x0680003
0x0C	0x0C	Y[15:0], Local	BIST (Unused)/Header Type/Latency Timer/Cache Line Size	0x00002008
0x10	0x10	Y	PCI Base Addr 0 for Memory Mapped Local/Runtime/DMA Registers (PCIBAR0)	0x00000000
0x14	0x14	Y	PCI Base Addr 1 for I/O Mapped Local/Runtime/DMA Registers (PCIBAR1)	0x00000001
0x18	0x18	Y	PCI Base Addr 2 for Local Addr Space 0 (PCIBAR2)	0x00000000
0x1C	0x1C	Y	PCI Base Addr 3 for Local Addr Space 1 (PCIBAR3) (Unused)	0x00000000
0x20	0x20	Y	PCI Base Addr 4 for Local Addr Space 2	
0x24	0x24	Y	PCI Base Addr 5 for Local Addr Space 3	
0x28	0x28	N	Cardbus CIS Pointer (Not supported)	
0x2C	0x2C	N	Subsystem ID/Subsystem Vendor ID	0x90802400
0x30	0x30	Y	PCI Base Address to Local Expansion ROM (Unused)	0x00000000
0x34	0x34	Y[7:0], Local	Reserved Next-Cap Pointer	
0x38	0x38	Local	Reserved	
0x3C	0x3C	Y[7:0], Local	Max_Lat/Min_Gnt/Interrupt Pin/Interrupt Line	0x00000100
0x40	0x180	Y[31:8]	Power Management Capabilities/Next_Cap Pointer/Capability ID	

0x44	0x184	Y[15, 12:8], [1:0]	Data/PMCSR Bridge Support Extensions/Power Management Control/Status Register	
0x48	0x188	Y[23:16], Local [15:0]	Reserved/Control/Status Register/Next_Cap Pointer/Capability ID	
0x4C	0x18C	Y[31:16], Local [15:8]	F/VPD Address/Next_Cap Pointer/Capability ID	
0x50	0x190	Y	VPD Data	

Note: The Local Base Address for the PCI Configuration registers in Local Address Space is 0xC0000000. However, there should be no need for the user to access the PCI Configuration registers through Local Address Space.

4.1.1 PCI CONFIGURATION ID REGISTER
(Offset 0x00, Reset 0x965610B5)

D15:0 Vendor ID –– 0x10B5 = PLX Technology
D31:16 Device ID –– 0x9656 = PCI9656

4.1.2 PCI COMMAND REGISTER
(Offset 0x04, Reset 0x0017)

D0 I/O Space
A '1' allows the device to respond to I/O space accesses.
D1 Memory Space
A '1' allows the device to respond to memory space accesses.
D2 PCI Master Enable.
A '1' allows the device to behave as a PCI bus master.
Note: *This bit must be set for the PCI 9656 to perform DMA cycles*.
D3 Special Cycle. (*Not Supported*.)
D4 Memory Write/Invalidate.
A '1' enables memory write/invalidate.
D5 VGA Palette Snoop. (*Not Supported*.)
D6 Parity Error Response
A '0' indicates that a parity error is ignored and operation continues.
A '1' indicates that parity checking is enabled.
D7 Wait Cycle Control. Controls whether the device does address/data stepping.
A '0' indicates the device never does address/data stepping.
Note: *Hardcoded to 0.*
D8 SERR# Enable
A '1' allows the device to drive the SERR# line.
D9 Fast Back-to-Back Enable. Indicates what type of fast back-to-back transfers a Master can perform on the bus.
A '1' indicates fast back-to-back transfers can occur to any agent on the bus.
A '0' indicates fast back-to-back transfers can only occur to the same agent as the previous cycle.
D15:10 Reserved

4.1.3 PCI STATUS REGISTER

(Offset 0x06, Reset 0x0280)

D3:0 Reserved

D4 New Capability Functions Support
Writing a '1' supports New Capabilities Functions. If enabled, the first New Capapbility Function ID is located at PCI Configuration offset [40h]. Can be written only from the Local Bus. Read-only from the PCI Bus.

D5 66 MHz-Capable
If set to '1', this device supports a 66 MHz PCI clock environment.

D6 User Definable Features Supported
A '1' indicates UDF are supported.
Note: *User Definable Features are Not Implemented*

D7 Fast Back-to-Back Capable.
A '1' indicates the adapter can accept fast back-to-back transactions.

D8 Master Data Parity Error Detected
A '1 indicates the following three conditions are met:
1. PCI9656 asserted PERR# itself or observed PERR# asserted.
2. PCI9656 was bus master for the operation in which the error occurred.
3. Parity Error Response bit in the Command Register is set.

Writing a '1' to this bit clears the bit.

D10:9 DEVSEL Timing. Indicates timing for DEVSEL# assertion.
A value of '01' indicates a medium decode.
Note: *Hardcode to 01.*

D11 Target Abort
A '1' indicates the PCI9656 has signaled a target abort.
Writing a '1' to this bit clears the bit.

D12 Received Target Abort
A '1' indicates the PCI9656 has received a target abort.
Writing a '1' to this bit clears the bit.

D13 Master Abort
A '1' indicates the PCI9656 has generated a master abort signal.
Writing a '1' to this bit clears the bit.

D14 Signal System Error
A '1' indicates the PCI9656 has reported a system error on the SERR# signal.
Writing a '1' to this bit clears the bit.

D15 Detected Parity Error
A '1' indicates the PCI9656 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:
1. PCI9656 detected a parity error during a PCI address phase.
2. PCI9656 detected a data parity error when it was the target of a write.
3. PCI9656 detected a data parity error when performing a master read.

Writing a '1' to this bit clears the bit.

4.1.4 PCI REVISION ID REGISTER
(Offset 0x08)

D7:0 Revision ID - The silicon revision of the PCI9656.

4.1.5 PCI CLASS CODE REGISTER
(Offset 0x09-0B, Reset=0x068000)

D7:0 Register level programming interface

0x00 = Queue Ports at 0x40 and 0x44.
0x01 = Queue Ports at 0x40 and 0x44, Int Status and Int Mask at 0x30 and 0x34

D15:8 Sub-class Code - 0x80 = Other bridge device.

D23:16 Base Class Code. - 0x06 = Bridge Device

4.1.6 PCI CACHE LINE SIZE REGISTER
(Offset 0x0C, Reset 0x00)

D7:0 System cache line size in units of 32-bit words.

4.1.7 PCI LATENCY TIMER REGISTER
(Offset 0x0D, Reset 0x00)

D7:0 PCI Latency Timer. Units of PCI bus clocks, the amount of time the PCI9656, as a bus master, can burst data on the PCI bus.

4.1.8 PCI HEADER TYPE REGISTER
(Offset 0x0E, Reset 0x00)

D6:0 Configuration Layout Type = 0

D7 Header Type = 0.

4.1.9 PCI BASE ADDRESS REGISTER FOR MEMORY ACCESS TO LOCAL/RUNTIME/DMA REGISTERS
(Offset 0x010, Reset 0x00000000)

D0 Memory Space Indicator
A '0' indicates register maps into Memory space.
Note: *Hardcoded to 0.*

D2:1 Location of Register:
00 - Locate anywhere in 32-bit memory address space
Note: *Hardcoded to 0.*

D3 Prefetchable.
Note: *Hardcoded to 0.*

D7:4 Memory Base Address.
Default Size = 256 bytes.
Note: *Hardcoded to 0.*

D31:8 Memory Base Address.
Memory base address for access to Local, Runtime, and DMA registers.

Note: *PCIBAR0 is Memory Mapped Base Address of PCI9656 Registers*

4.1.10 PCI BASE ADDRESS REGISTER FOR I/O ACCESS TO LOCAL/RUNTIME/DMA REGISTERS
(Offset 0x14, Reset 0x00000001)

D0 Memory Space Indicator
A '1' indicates the register maps into I/O space.
Note: *Hardcoded to 1.*

D1 Reserved

D7:2 I/O Base Address.
Default Size = 256 bytes.
Note: *Hardcoded to 0.*

D31:8 I/O Base Address.

Base Address for I/O access to Local, Runtime, and DMA Registers.

Note: *PCIBAR1 is I/O Mapped Base Address of PCI9656 Registers*

4.1.11 PCI BASE ADDRESS REGISTER FOR MEMORY ACCESS TO LOCAL ADDRESS SPACE 0
(Offset 0x18, Reset 0x00000000)

D0	Memory Space Indicator A '0' indicates register maps into Memory space. (Specified in Local Address Space 0 Range Register - LAS0RR.)
D2:1	Location of register (if memory space). Location values: 00 - Locate anywhere in 32-bit memory address space (Specified in Local Address Space 0 Range Register - LAS0RR.)
D3	Prefetchable A '0' indicates reads are not prefetchable. (Specified in Local Address Space 0 Range Register - LAS0RR
D31:4	Memory Base Address Memory base address for access to Local Address Space 0.

4.1.12 PCI SUBSYSTEM DEVICE/VENDOR ID REGISTER
(Offset 0x2C, Reset 0x908010B5)

D15:0	Subsystem Vendor ID – 0x10B5 = PLX Technology
D31:16	Subsystem Device ID – 0x2705 = General Standards Corporation PCI64-HPDI32A.

4.1.13 PCI INTERRUPT LINE REGISTER
(Offset 0x3C, Reset 0x00)

D7:0	Interrupt Line Routing Value. Indicates which input of the system interrupt controller(s) to which the interrupt line of the device is connected.

4.1.14 PCI INTERRUPT PIN REGISTER
(Offset 0x3D, Reset 0x01)

D7:0	Interrupt Pin register. Indicates which interrupt pin the device uses. The following values are decoded (the PCI 9656 supports only INTA#): 0=No interrupt pin 1=INTA# 2=INTB# 3=INTC# 4=INTD#

4.1.15 PCI MIN_GNT REGISTER
(Offset 0x3E, Reset 0x00)

D7:0	Minimum Grant Specifies how long a burst period device needs, assuming a clock rate of 33 MHz. Value is a multiple of ¼ microsecond increments.

4.1.16 PCI MAX_LAT REGISTER
(Offset 0x3F, Reset 0x00)

D7:0	Maximum Latency

Specifies how often the device must gain access to the PCI Bus. Value is a multiple of ¼ microsecond increments.

4.2 LOCAL CONFIGURATION REGISTERS

The Local Configuration registers give information on the Local side implementation. Since Local Expansion ROM, Local Address Space 1, and Direct Master accesses are not implemented on the PCI64-HPDI32A, the descriptions of these registers have been omitted. Most of the Local Configuration Registers are preloaded from the configuration Serial EEPROM at system reset.

Table 4.2: Local Configuration Registers

PCI CFG Addr	Local Offset Addr	PCI/Local Writable	Register Name	Value after Reset
0x00	0x80	Y	Range for PCI to Local Address Space 0	0xFFFFF000
0x04	0x84	Y	Local Base Address (Remap) for PCI to Local Address Space 0 (Unused)	0x00000000
0x08	0x88	Y	Mode/Arbitration Register	0x00000000
0x0C	0x8C	Y	Big/Little Endian Descriptor	0x00000000
0x10	0x90	Y	Range for PCI to Local Expansion ROM (Unused)	0x00000000
0x14	0x94	Y	Local Base Address (Re-map) for PCI to Local Expansion ROM and BREQo control (Unused)	0x00000000
0x18	0x98	Y	Local Bus Region Descriptions for PCI Local Accesses	0x00000000
0x1C	0x9C	Y	Range for Direct Master to PCI (Unused)	0x00000000
0x20	0xA0	Y	Local Base Address for Direct Master to PCI Memory (Unused)	0x00000000
0x24	0xA4	Y	Local Base Address for Direct Master to PCI Memory IO/CFG (Unused)	0x00000000
0x28	0xA8	Y	PCI Base Address (Re-map) for Direct Master to PCI (Unused)	0x00000000
0x2C	0xAC	Y	PCI Configuration Address Register for Direct Master to PCI IO/CFG (Unused)	0x00000000
0xF0	0x170	Y	Range for PCI to Local Address Space 1 (Unused)	0x00000000
0xF4	0x174	Y	Local Base Address (Remap) for PCI to Local Address Space 1 (Unused)	0x00000000
0xF8	0x178	Y	Local Bus Region Descriptor (Space 1) for PCI to Local Accesses (Unused)	0x00000000

4.1.1 LOCAL ADDRESS SPACE 0 RANGE REGISTER FOR PCI TO LOCAL BUS (PCI 0x00, Reset 0xFFFFF000)

D0 Memory Space Indicator
A '0' indicates register maps into Memory space.

D2:1 Location of register (if memory space). Location values:
00 - Locate anywhere in 32-bit memory address space
01 - Locate below 1 MB in PCI Address space
10 - Locate anywhere in 64-bit PCI Address space
11 - Reserved

D3 Prefetchable
A '0' indicates reads are not prefetchable.

D31:4 Specifies which PCI address bits will be used to decode a PCI access to Local Address Space 0. A '1' indicates bit is included in address decode.

Local Address Space 0 value 0xFFFFF000 maps a 4kbyte range.
Since entire Local Address Space can be mapped into 4kb range, the remap register is not used.

4.2.2 MODE/ARBITRATION REGISTER
(PCI 0x08)

D7:0	Local bus Latency Timer (Unused)
D15:8	Local bus Pause Timer (Unused)
D16	Local bus Latency Timer Enable (Unused)
D17	Local bus Pause Timer Enable (Unused)
D18	Local bus BREQ Enable (Unused)
D20:19	DMA Channel Priority 00 = Rotational priority 01 = Channel 2 priority 10 = Channel 1 priority 11 = Reserved
D21	Local bus direct slave give up bus mode A value of 1 indicates local bus will be released when PCI9080 write FIFO empty or read FIFO full.
D22	Direct slave LLOCKo# Enable (Unused)
D23	PCI Request Mode
D24	PCI Rev 2.1 Mode
D25	PCI Read No Write Mode
D26	PCI Read with Write Flush Mode
D27	Gate the Local Bus Latency Timer with BREQ (Unused)
D28	PCI Read No Flush Mode
D29	Reads Device/Vendor ID or SubDevice/SubVendor ID
D30	FIFO Full Status Flag.
D31	BIGEND#/WAIT# Input/Output Select (M mode only).

4.2.3 BIG/LITTLE ENDIAN DESCRIPTOR REGISTER
(PCI 0x0C)

Since local bus is little endian, all bits should be left zero

4.2.4 LOCAL ADDRESS SPACE 0/EXPANSION ROM BUS REGION DESCRIPTOR REGISTER
(PCI 0x18, Reset 0x40030143)

D1:0	Memory Space 0 Local Bus Width 11 indicates 32-bit local bus
D5:2	Memory Space 0 Internal Wait States A ‘0’ indicates no wait states required
D6	Memory Space 0 Ready Input Enable A ‘1’ indicates Local Ready input enabled.
D7	Memory Space 0 Bterm Input Enable (Unused)
D8	Memory Space 0 Prefetch Disable (Unused)
D9	Expansion ROM Space Prefetch Disable (Unused)
D10	Read Prefetch Count Enable (Unused)
D14:11	Prefetch Counter (Unused)
D15	Reserved
D17:16	Expansion ROM Space Local Bus Width (Unused)

D21:18 Expansion ROM Space Internal Wait States (Unused)
D22 Expansion ROM Space Ready Input Enable (Unused)
D23 Expansion ROM Space Bterm Input Enable (Unused)
D24 Memory Space 0 Burst Enable
D25 Extra Long Load from Serial Enable
D26 Expansion ROM Space Burst Enable (Unused)
D27 Direct Slave PCI Write Mode
D28:31 PCI Target Retry Delay Clocks

4.3 RUNTIME REGISTERS

The Runtime registers consist of mailbox registers, doorbell registers, and a general-purpose control register. The mailbox and doorbell registers serve no purpose on the PCI64-HPDI32A.

Table 4.3: Runtime Registers

PCI CFG Addr	Local Offset Addr	PCI/Local Writable	Register Name	Value after Reset
0x40	0xC0	Y	Mailbox Register 0 (Unused)	0x00000000
0x44	0xC4	Y	Mailbox Register 1 (Unused)	0x00000000
0x48	0xC8	Y	Mailbox Register 2 (Unused)	0x00000000
0x4C	0xCC	Y	Mailbox Register 3 (Unused)	0x00000000
0x50	0xD0	Y	Mailbox Register 4 (Unused)	0x00000000
0x54	0xD4	Y	Mailbox Register 5 (Unused)	0x00000000
0x58	0xD8	Y	Mailbox Register 6 (Unused)	0x00000000
0x5C	0xDC	Y	Mailbox Register 7 (Unused)	0x00000000
0x60	0xE0	Y	PCI to Local Doorbell Register (Unused)	0x00000000
0x64	0xE4	Y	Local to PCI Doorbell Register (Unused)	0x00000000
0x68	0xE8	Y	Interrupt Control/Status	0x00000000
0x6C	0xEC	Y	General Purpose Control	0x00000000
0x70	0xF0	N	Permanent Device ID/ Permanent Vendor ID	0x10B59656
0x74	0xF4	N	Permanent Revision ID	0x0000000X
0x78	0xC0	Y	Mailbox Register 0 (Unused)	0x00000000
0x7C	0xC4	Y	Mailbox Register 1 (Unused)	0x00000000

4.3.1 INTERRUPT CONTROL /STATUS
(PCI 0x68, Reset 0x00000000)

D0 Enable Local bus LSERR# (Unused)
D1 Enable Local bus LSERR# on a PCI parity error (Unused)
D2 Generate PCI Bus SERR#
D3 Mailbox Interrupt Enable (Unused)
D4 Power Management Interrupt Enable
D5 Power Management Interrupt
D6 Direct Master Write/Direct Slave Read Local Data Parity Check Error Enable
D7 Direct Master Write/Direct Slave Read Local Data Parity Check Error Status
D8 PCI Interrupt Enable
D9 PCI Doorbell Interrupt Enable (Unused)

D10 PCI Abort Interrupt Enable
D11 PCI Local Interrupt Enable
Local Interrupt must be enabled for USC/FIFO interrupts.
D12 Retry Abort Enable (Unused)
D13 PCI Doorbell Interrupt Status.
D14 PCI Abort Interrupt Status
D15 PCI Local Interrupt Status
D16 Local Interrupt Output Enable
D17 Local Doorbell Interrupt Enable (Unused)
D18 Local DMA Channel 0 Interrupt Enable
D19 Local DMA Channel 1 Interrupt Enable
D20 Local Doorbell Interrupt Status
D21 DMA Channel 0 Interrupt Status
D22 DMA Channel 1 Interrupt Status
D23 BIST Interrupt Status
D24 A '0' indicates a Direct Master was bus master during a Master or Target abort.
D25 A '0' indicates that DMA CH0 was bus master during a Master or Target abort.
D26 A '0' indicates that DMA CH1 was bus master during a Master or Target abort.
D27 A '0' indicates that a Target Abort was generated by the PCI9080 after 256 consecutive Master retries to a Target.
D28 Reading a '1' indicates the PCI Bus wrote data to MBOX0. Enabled only if the Mailbox Interrupt Enable bit is set (INTCSR[3]=1).
D29 Reading a '1' indicates the PCI Bus wrote data to MBOX1. Enabled only if the Mailbox Interrupt Enable is set (INTCSR[3]=1).
D30 Reading a '1' indicates the PCI Bus wrote data to MBOX2. Enabled only if the Mailbox Interrupt Enable bit is set (INTCSR[3]=1).
D31 Reading a '1' indicates the PCI Bus wrote data to MBOX3. Enabled only if the Mailbox Interrupt Enable bit is set (INTCSR[3]=1).

4.3.2 SERIAL EEPROM CONTROL, PCI COMMAND CODES, USER I/O CONTROL, INIT CONTROL REGISTER (PCI 0x6C, Reset 0x0x001767E)

D3:0 PCI Read Command Code for DMA
D7:4 PCI Write Command Code for DMA
D11:8 PCI Memory Read Command Code for Direct Master (Unused)
D15:12 PCI Memory Write Command Code for Direct Master (Unused)
D16 General Purpose Output (Unused)
D17 General Purpose Input (Unused)
D18 Writing a '1' selects USERi to be an input to the chip.
Writing a '0' selects LLOCKi# as an input. Enables the user to select between The USERi and LLOCKi# functions when USERi is chosen to be an output. The select bit(s) for the pin is DMAMODE0[12] and/or DMAMODE1[12].
D19 Writing a '1' selects USERo to be an output from the chip.
Writing a '0' selects LLOCKo# as an output. Enables the user to select between the USERo and LLOCKo# functions when USERo is chosen to be an output. The select bit(s) for the pin is DMAMODE0[12] and/or DMAMODE1[12].
D20 LINTo#Interrupt Status
D21 TEA#/LSERR# Interrupt Status
D23:22 Reserved
D24 Serial EEPROM clock for Local or PCI bus reads or writes to Serial EEPROM.
D25 Serial EEPROM chip select
D26 Write bit to serial EEPROM

D27 Read serial EEPROM data bit
D28 Serial EEPROM present
D29 Reload Configuration Registers
D30 PCI Adapter Software Reset
D31 Local Init Status
A '1' indicates Local initialization done.

4.3.3 PCI PERMANENT CONFIGURATION ID REGISTER
(PCI 0x70, Reset 0x10B59080)

D15:0 Permanent Vendor ID (0x10B5)
D31:16 Permanent Device ID (0x9656)

4.3.4 PCI PERMANENT REVISION ID REGISTER
(PCI 0x74)

D7:0 Permanent Revision ID

4.4 LOCAL DMA REGISTERS

The Local DMA registers are used to setup the DMA transfers to and from the on-board FIFOs. Since the PCI64-HPDI32A is half-duplex (data is only transferred in one direction at a time), only DMA Channel 0 is used.

Table 4.4: DMA Registers

PCI CFG Addr	Local Offset Addr	PCI/Local Writable	Register Name	Value after Reset
0x80	0x100	Y	DMA Channel 0 Mode Register	0x00000003
0x84	0x104	Y	DMA Channel 0 PCI Address Register	0x00000000
0x88	0x108	Y	DMA Channel 0 Local Address Register	0x00000000
0x8C	0x10C	Y	DMA Channel 0 Transfer Byte Count Register	0x00000000
0x90	0x110	Y	DMA Channel 0 Descriptor Pointer Register	0x00000000
0x94	0x114	Y	DMA Channel 1 Mode Register (Unused)	0x00000003
0x98	0x118	Y	DMA Channel 1 PCI Address Register (Unused)	0x00000000
0x9C	0x11C	Y	DMA Channel 1 Local Address Register (Unused)	0x00000000
0xA0	0x120	Y	DMA Channel 1 Transfer Byte Count Register (Unused)	0x00000000
0xA4	0x124	Y	DMA Channel 1 Descriptor Pointer Register (Unused)	0x00000000
0xA8	0x128	Y	DMA Channel 1Command/Status Register DMA Channel 0 Command/Status Register	0x00000010
0xAC	0x12C	Y	DMA Mode/ Arbitration Register	0x00000000
0xB0	0x130	Y	DMA Threshold Register	0x00000000

4.4.1 DMA CHANNEL 0 MODE REGISTER (PCI 0x80)

D1:0 Local Bus Width
00 = 8 bit DMA transfer width
01 = 16 bit DMA transfer width
10/11 = 32 bit DMA transfer width

D5:2 Internal Wait States (Unused)

D6 Ready Input Enable
Note: This bit should always be set to '1' (Ready Input Enabled)

D7 Bterm# Input Enable (Unused)
Note: This bit should always be set to '0' (BTERM# Disabled)

D8 Local Burst Enable
Note: If Burst enabled, the user must ensure FIFO will not become empty (read) or full (write) during the burst access. For Demand Mode DMA, this means the Almost Empty/Almost Full flags should be set to a value of at least 8.

D9 Chaining Enable
A '1' indicates chaining mode is enabled.
For chaining mode, the DMA source address, destination address and byte count are loaded from memory in PCI Space.

D10 Done Interrupt Enable
A '1' enables interrupt when DMA done.

Note: If DMA clear count mode is enabled, the interrupt won't occur until the byte count is cleared.

D11 Local Addressing Mode
A '1' indicates local addresses LA [31:2] to be held constant.
Note: This bit should always be set to '1' (no address increment)

D12 Demand Mode Enable
A '1' causes the DMA controller to operate in Demand Mode.
In Demand Mode, the DMA controller transfers data when its DREQ# input is asserted. The DMA controller transfers Lwords (32bits) of data. This may result in multiple transfers for an 8 or 16 bit bus.

D13 Memory Write and Invalidate Mode for DMA Transfers.
When set to a '1', the PCI 9656 performs Memory Write and Invalidate cycles to the PCI bus. The PCI 9656 supports Memory Write and Invalidates sizes of 8 or 16 Lwords. Size is specified in the System Cache Line Size bits (PCICLSR[7:0]). If a size other than 8 or 16 is specified, the PCI 9656 performs Write transfers rather than Memory Write and Invalidate transfers. Transfers must start and end at cache line boundaries.

D14 DMA EOT (End of Transfer) Enable (Unused)

D15 DMA Stop Data Transfer Mode
A'0' sends a BLAST to terminate DMA transfer
Note: This bit should always be set to '0'.

D16 DMA Clear Count Mode (Unused)

D17 DMA Channel 0 Interrupt Select
A '1' routes the DMA Channel 0 interrupt to the PCI interrupt.
Note: This bit should always be set to '1'.

D18 DAC Chain Load
When set to a '1', enables the descriptor to load the PCI Dual Address Cycle value. Otherwise, it uses the contents of the register.

D19 EOT# END Link

D20 Valid Mode Enable
Value of '0' indicates the Valid bit (DMASIZ0[31]) is ignored. Value of a '1' Indicates the DMA descriptors are processed only when the Valid bit is set (DMASIZ0[31]). If the Valid bit is set, the transfer count is '0', and the descriptor is not the last descriptor in the chain. The DMA controller then moves to the next descriptor in the chain.

D21 Valid Stop Control
Value of '0' indicates the DMA Chaining controller continuously polls a descriptor with the Valid bit set to 0 (invalid descriptor) if the Valid Mode Enable bit is set (bit [20]=1). Value of a '1' indicates the Chaining controller stops polling when the Valid bit with a value of '0' is detected (DMASIZ0[31]=0). In this case, the CPU must restart the DMA controller by setting the Start bit (DMACSR0[1]=1). A pause sets the DMA Done register.

D31:22 Reserved

4.4.2 DMA CHANNEL 0 PCI ADDRESS REGISTER (PCI 0x84)

D31:0 PCI Address Register

4.4.3 DMA CHANNEL 0 LOCAL ADDRESS REGISTER (PCI 0x88)

D31:0 Local Address Register

Note: Should be set to Local FIFO offset 0x18

4.4.4 DMA CHANNEL 0 TRANSFER SIZE (BYTES) REGISTER (PCI 0x8C)

D22:0	DMA Transfer Size
D30:23	Reserved
D31	Valid

4.4.5 DMA CHANNEL 0 DESCRIPTOR POINTER REGISTER (PCI 0x90)

D0	Descriptor Location A '1' indicates PCI address space. **Note:** This bit should always be set to '1' if Chained DMA enabled.
D1	End of Chain
D2	Interrupt after Terminal Count
D3	Direction of transfer A '1' indicates transfers from local bus to PCI bus (Read Receive FIFO) A '0' indicates transfers from local bus to PCI bus (Write Transmit FIFO)
D31:4	Next Descriptor Address

4.4.6 DMA CHANNEL 0 COMMAND/STATUS REGISTER (PCI 0xA8)

D0	Channel 0 Enable
D1	Channel 0 Control
D2	Channel 0 Abort
D3	Clear Interrupt
D4	Channel 0 Done
D7:5	Reserved

4.4.7 DMA ARBITRATION REGISTER (PCI 0xAC)

Same as Mode /Arbitration Register (MARBR) (PCI 0x08 – See Section 2.2.2)

4.4.8 DMA THRESHOLD REGISTER (PCI 0xB0)

D3:0	DMA Channel 0 PCI to Local Almost Full (C0PLAF)
D7:4	DMA Channel 0 Local to PCI Almost Empty (C0LPAE)
D11:8	DMA Channel 0 Local to PCI Almost Full (C0LPAF)
D15:12	DMA Channel 0 PCI to Local Almost Empty (C0PLAE)
D19:16	DMA Channel 1 PCI to Local Almost Full (C1PLAF) (Unused)
D23:20	DMA Channel 1 Local to PCI Almost Empty (C1LPAE) (Unused)
D27:24	DMA Channel 1 PCI to Local Almost Full (C1LPAF) (Unused)
D31:28	DMA Channel 1 PCI to Local Almost Empty (C1PLAE) (Unused)

4.5 MESSAGING QUEUE REGISTERS

Messaging queue registers are not used on the PCI64-HPDI32A.

CHAPTER 5: HARDWARE CONFIGURATION

5.0 HARDWARE JUMPERS (J1)

The EEPROM jumper header (J1) is a 4x2 header which contains four mini jumpers installed (PCI factory default) or two jumpers installed (PMC factory default). These jumpers (described below) should remain installed for most applications.

To locate the hardware jumpers, view the top side of the card with the cable interface connector to the right. The jumper block is located about two inches to the right of the connector in the center of the card. The jumpers will number top to bottom, with Jumper 1 toward the top and Jumper 4 toward the bottom.

PCI Only:

Two general purpose jumpers are used on the PCI version only. These jumpers are present on the PMC version, but are not connected. They may be connected on a future PMC version. These jumpers may be read from the Board Status register – installed = 1, removed = 0. If more than one HPDI32A board is installed in a system, Board Jumpers 0/1 can provide an addressing method to distinguish between the cards via software.

Board Jumper 1 (Jumper 1 closest to top of card).
Board Jumper 0 (Jumper 2 second from top of card).

PCI/PMC:

FPGA Reload (Jumper 3 -third from top of card)

This jumper connects the PCI reset signal to the FPGA reload. This forces the FPGAs to reload firmware upon a PCI reset. This allows a recovery mechanism of the FGPA firmware if a catastrophic failure occurs.

PLX Default Configuration (Jumper 4 - closest to bottom of card)

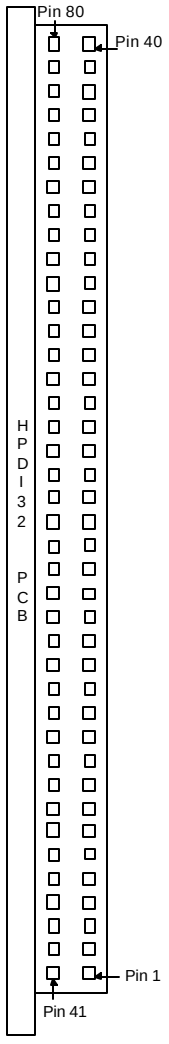
This jumper connects the PLX controller chip to the On-board Configuration EEPROM. When installed, the PLX will preload registers based on Serial EPROM data. This information is necessary for correct PCI configuration and driver installation. If the Configuration EEPROM becomes corrupted, PCI configuration may lockup due to invalid parameters. **Removing this jumper will force the PLX into a default configuration that should allow PCI configuration to proceed (OS will boot, and user can take action to reprogram Configuration EEPROM). This jumper should only be removed following factory consultation. The board will not function correctly if this jumper is removed.**

5.1 CABLE INTERFACE CONNECTOR

The 80-pin cable interface connector (reference designator: P1) on the HPDI32A board is manufactured by Robinson Nugent - part number P50E-080-P1-SR1-TG. The mating part number is Robinson Nugent P50E-080-S-TG (50 mil. cabling is suggested for twisted pair), or Robinson Nugent P25E-080S-TG (25 mil. cabling may be used for multi-drop capability, but with loss of twisted pair). Table 5.0 shows the cable pinout.

Table 5.0: Cable Pin-Out

Pin No.	Cable Signal Name	Pin No.	Cable Signal Name
1	CABLE CLK +	41	CABLE D12 +
2	CABLE CLK -	42	CABLE D12 -
3	FRAME VALID +	43	CABLE D13 +
4	FRAME VALID -	44	CABLE D13 -
5	LINE VALID / GPIO0 +	45	CABLE D14 +
6	LINE VALID / GPIO0 -	46	CABLE D14 -
7	STATUS VALID / GPIO1 +	47	CABLE D15 +
8	STATUS VALID / GPIO1 -	48	CABLE D15 -
9	RX READY / GPIO2 +	49	CABLE D16 +
10	RX READY / GPIO2 -	50	CABLE D16 -
11	TX DATA READY / GPIO3 +	51	CABLE D17 +
12	TX DATA READY / GPIO3 -	52	CABLE D17 -
13	TX ENABLED / GPIO4 +	53	CABLE D18 +
14	TX ENABLED / GPIO4 -	54	CABLE D18 -
15	RX ENABLED / GPIO5 +	55	CABLE D19 +
16	RX ENABLED / GPIO5 +	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 -

CHAPTER 6: ORDERING INFORMATION

6.0 ORDERING INFORMATION

Since the HPDI32A is designed to fit a variety of high-speed digital interface needs, there are several options that must be specified when ordering the HPDI32A board. Please consult our sales department with your application requirements to decide on the correct ordering options.

6.0.1 BUS INTERFACE

RS485/422 Interface
The RS485/RS422 interface provides for synchronous bus clock speeds up to 26MHz (104 Mbytes per sec). This is the standard interface option.

PECL Interface
The PECL (Positive ECL) interface.

LVDS and TTL versions of the HPDI32A board are planned. Please consult factory to inquire on current status.

6.0.2 FIFO SIZE

The HPDI32A can accept FIFOs with depths ranging from 2k words to 128k words. Larger FIFO depth is important for faster interfaces to reduce the risk of software overhead. Standard configuration of the HPDI32A contains 32k word deep FIFOs.

Since the HPDI32A contains separate transmit and receive FIFOs, it is possible to have different size transmit and receive FIFOs. Please consult factory for different FIFO options.

A version of the HPDI32A is planned which will accommodate 256k word and 512k word deep FIFOs. Please consult factory to inquire on current status.

6.0.3 Interface Cable

General Standards Corporation can provide an interface cable for the HPDI32A board. This cable is twisted pair for increased noise immunity. Several standard cable lengths are offered, or the cable length can be custom ordered to the user's needs. Versions of the cable are available with connectors on both ends, or the cable may be ordered with a single connector to allow the user to adapt the other end for a specific application. Please consult factory for more information on cabling options and pricing.

6.1 CUSTOM APPLICATIONS

Although the HPDI32A board provides extensive flexibility to accommodate most user applications, custom interfaces exist, which may not exactly conform to the HPDI32A interface standard. General Standards Corporation has worked with many customers to provide customized versions based on the HPDI32A board. Please consult our sales department with your specifications to inquire about a custom application.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP