PMC-HPDI32B-COS

User's Manual

32 Bit Change Of State Board
with RS-422 / RS-485 Transceivers

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

Revision 0

PREFACE

Revision History

1. Rev NR – Jan 2017 – Original rev from PCI-HPDI32A-COS manual.

Additional copies of this manual or other **General Standards Corporation** literature may be obtained from:

General Standards Corporation
8302A Whitesburg Drive
Huntsville, Alabama 35802
Telephone: (256) 880-8787
Fax: (256) 880-8788
URL: www.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 right of any rights of others.

General Standards Corporation assumes no responsibility 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.

RELATED PUBLICATIONS

PLX PCI 9080 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/

PCI Local Bus Specification Revision 2.2 December 18, 1998

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

1.0 General Description

The PMC-HPDI32B-COS is a 32-bit parallel Change of State Detection board featuring high speed RS485 interface transceivers. The Change of State detection is can be customized to ignore specific bits and to provide an interrupt after a specified number of change of state events have occurred. The board will also function as a simple Logic Analyzer with a configurable trigger value and mask. The sample rate for both Change of Stete and Logic Analyzer is programmable from 20MHz down to 300Hz.

Received values are buffered in a 8K deep on-board FIFO which is easily accessed by the integrated a DMA Controller. 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.

Features:

- Selectable Change-Of-State or Logic Analyzer Operation.
- Bitwise Programmable Mask for COS Detection and Logic Analyzer Trigger
- 20 MHz Base Sample rate with Programmable Clock Divider.
- Programmable Event Counter for Change Of State Events.
- Programmable Logic Analyzer Trigger
- Output Data Port that can be driven onto the cable on byte wide boundaries.
- A variety of device drivers are available, including VxWorks, Windows, and Linux

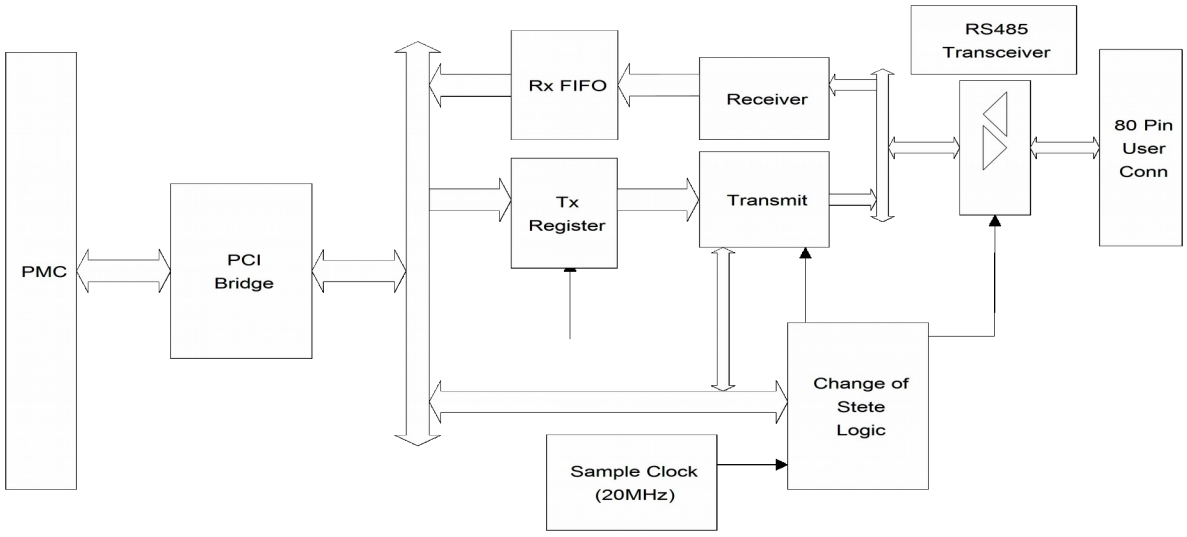


Figure 1-1 Block Diagram of PMC-HPDI32B-COS

1.1 Change Of State Detection

The PMC-HPDI32B-COS provides a simple interface to provide Change of State functionality.

1.2 Logic Analyzer

The PMC-HPDI32B-COS provides a simple interface to provide Change of State functionality.

1.3 Sample Clock Divider

The Clock Divider is capable of providing a divided down clock enable for operation at slower data rates.

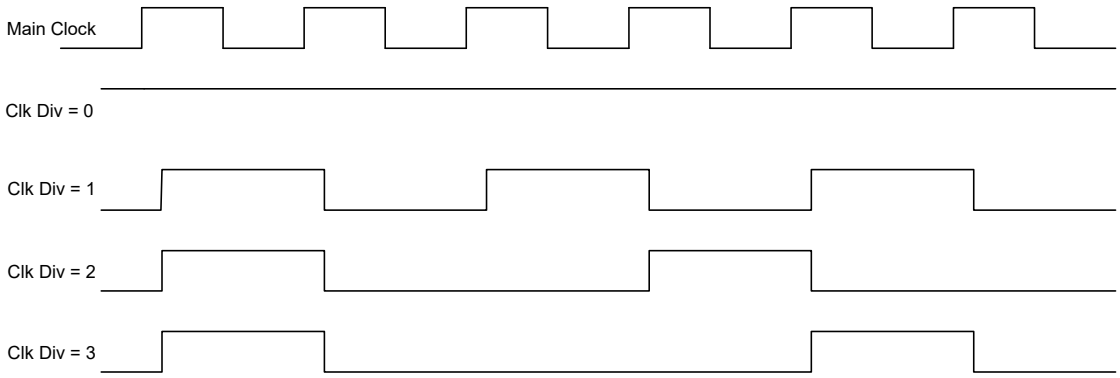


Figure Sample Clock Divider-1 Clock Division Operation

1.4 Receive FIFOs

The FIFO on the PMC-HPDI32B-COS are used for buffering the receive data.. The Receive FIFO consists of 32 bits of data and 4 status flags. The only configuration currently supported is the –32K board ordering option, which is 8K x 32bit words in the Receive FIFO. The receive FIFOs are loaded by the cable receive control logic and read by either the CPU or the DMA. The 4 status flags that accompany the Receive 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.

Data is transferred to/from the serial interface through Transmit and Receive FIFOs. Each of the four serial channels has an independent Transmit FIFO and a Receive FIFO for a total of eight separate on-board FIFOs. These FIFOs can vary from 4k bytes to 32 kbytes (based on ordering option). FIFOs allow data transfer to continue to/from the IO interface independent of PCI interface transfers and software overhead. The required FIFO size may depend on several factors including data transfer size, required throughput rate, and the software overhead (which will also vary based on OS). Generally, faster baud rates (greater than 500kbps) will require deeper FIFOs. Deeper FIFOs help ensure no data is lost for critical systems.

The HPDI32B-COS provides access to complete FIFO status to optimize data transfers. In addition to Empty and Full indicators, each FIFO has a programmable Almost Empty Flag and a programmable Almost Full Flag. These FIFO flags may be used as interrupt sources to monitor FIFO fill levels. In addition, real-time FIFO counters showing the exact number of words in the FIFO are also provided for each FIFO. By utilizing these FIFO counters, data transfers can be optimized to efficiently send and receive data.

1.5 RS422 / RS-485 Transceivers

The COS data is transferred over the user interface using high-speed RS422/RS485 transceivers. Removable on-board termination resistors allow the system interface to be easily customized.

1.6 PMC/PCI Interface

The control interface to the HPDI32B-COS is through the PMC/PCI interface. An industry standard PCI9080 bridge chip from PLX Technology is used to implement PCI Specification 2.1. The PCI9080 provides the 32bit, 33MHz (132MBit/sec) interface between the PCI bus and the Local 32 bit bus. It also provides for high-speed DMA transfers to efficiently move data to and from the board.

1.7 Connector Interface

The PMC-HPDI32B-COS provides a user IO interface through the 80-pin IO connector. The connector layout allows twisted pair cabling to be used with the differential data signals. Section 5 details pinout information.

CHAPTER 2: LOCAL SPACE REGISTERS

2.0 Register Map

The HPDI32B-COS is accessed through two sets of registers – PCI Registers, and GSC Firmware Registers. The GSC Firmware Registers are referred to as Local Space Registers and are described below. The PCI registers are discussed in Chapter 3.

2.1 GSC Firmware Registers

The GSC Firmware Registers provide the primary control/status for the HPDI32B-COS board. The following table shows the GSC Firmware Registers.

Offset Address	Size	Access*	Register Name	Default Value (Hex)
0x0000	D32	Read Only	Firmware Revision	00100202
0x0004	D32	Read/Write	Board Control	00000000
0x0008	D32	Read Only	Board Status	0000CEXX
0x000C	D32	Read Only	Event Counter	00000000
0x0010	D32	Read/Write	Rx Almost Full/Empty	00080008
0x0014	D32	Read/Write	Sample Clock Divider	00000000
0x0018	D32	Read Only	Rx Data FIFO	XXXXXXXX
0x0018	D32	Write Only	Tx Data Output	00000000
0x001C	D32	Read Only	Rx Data Input	XXXXXXXX
0x0020	D32	Read/Write	Logic Analyzer Trigger	00000000
0x0024	D32	Read/Write	Data Mask	FFFFFFFF
0x0028	D32	Read/Write	Event Counter Initial Value	00000000
0x002C	D32	Read/Write	Rx Word Count	00000000
0x0030	D32	Read/Write	Interrupt Control	00000000
0x0034	D32	Read/Write	Interrupt Status	00000000
0x0038-0x0040	---	--	RESERVED	--------
0x0044	D32	Read Only	Rx FIFO Size	00002000
0x0048	---	--	RESERVED	--------
0x004C	D32	Read Only	Rx FIFO Word Count	00000000
0x0050	D32	Read Only	Interrupt Edge/Level	FFFF7777
0x0054	D32	Read/Write	Interrupt High/Low	FFFFFFFF
0x0058-0x00FC	---	--	RESERVED	--------

2.1.1 Firmware Revision: Local Offset 0x0000

The Firmware ID register provides version information about the firmware on the board. This is useful for technical support to identify the firmware version.

D31..D24	Reserved - 0x00
D23..D16	Board Identifier 0x10 - Identifies the HPDI32B-COS
D15..D8	Reserved - 0x02
D7..D0	Firmware Revision FW Revision 0x02

2.1.2 Board Control: Local Offset 0x0004

The Board Control Register defines the general control functions for the board.

D17-D31	Reserved.
D16	COS / Logic Analyzer Mode. 0 = Change of State Mode 1 = Logic Analyzer Mode
D15	Tx Data Byte 3 Enable 1 = Drive Cable Data D31-D24.
D14	Tx Data Byte 2 Enable 1 = Drive Cable Data D23-D16.
D13	Tx Data Byte 1 Enable 1 = Drive Cable Data D15-D8.
D12	Tx Data Byte 0 Enable 1 = Drive Cable Data D7-D0.
D11..D8	Reserved
D7	Loopback 1 - Allow Enable Tx and Start Rx Testing
D6	Disable Output Status Lines 0 = Drive Cable Command D5 and D6
D5	Start Rx 1 = Start the Receiver
D4	Enable Tx 1 = Drive Output Register Data to the cable.
D3	Reserved
D2	FIFO Reset 1 = Reset Rx FIFO. Note: This bit auto clears after FIFO reset.
D1	Reserved
D0	Board Reset 1 = Reset Board logic and FIFO. Note : This bit auto clears after reset

2.1.3 Board Status: Local Offset 0x0008

The Board Status Register gives general overall status for a board. The Board Jumpers (D1:D0) are physical jumpers which can be used to distinguish between boards if multiple HPDI32B boards are present in a system.

D31..D27	Reserved
D26	COS Event Detected 1 = Receiver has detected a COS event. .
D25	Event Counter EQ Zero 1 = Event Counter has Decremented to Zero.
D24	Receive Running 1 = Receiver has been started and has not become Almost Full.
D23	Logic Analyzer Triggered 1 = Logic Analyzer has detected the Trigger.
D22	Receive Stopped 1 = FIFO is Almost Full and the Receiver has Stopped.
D21	Reserved
D20	Rx FIFO Overflow 1 = Attempt to write FIFO while Full has occurred (latched) Note: Writing '1' to this bit will clear
D19	Rx FIFO Underflow 1 = Attempt to read FIFO while Empty has occurred (latched) Note: Writing '1' to this bit will clear
D18	Reserved
D17 - D16	Board Jumper 1 - 0 1 = Board Jumper present
D15	FIFO Full L 0 = FIFO is Full
D14	FIFO Almost Full L 0 = FIFO is Almost Full
D13	FIFO Almost Empty L 0 = FIFO is Almost Empty
D12	FIFO Empty L 0 = FIFO is Empty
D11 - D7	Reserved
D6 - D0	Rx Cable Command D6 - D0 Current state of Cable Command Signals

2.1.4 COS Event Count: Local Offset 0x000C (Read Only)

The Event Counter decrements every time a COS event is detected until it reaches zero. The Event Counter is loaded from the Event Count Initial Value Register (0x0028) when the receiver is started. This register gives shows the current value of the Event Counter while the receiver is running..

D31..D0	Event Count Current value of COS Event Counter Decrements from Event Counter Initial Value to Zero while the Receiver is running

2.1.5 FIFO Almost Flags: Local Offset 0x0010

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

The Almost Flag Registers are used to set the Almost Full and Almost Empty Flags for the receive FIFOs. The Almost Full/Empty Flags may be read as status bits in the Channel Control/Status Register, and are also edge-triggered interrupt sources to the Interrupt Register.

D31..D16	Almost Full Value Number of words from FIFO Full when the Almost Full Flag will be asserted (i.e. FIFO contains {FIFO Size – Almost Full Value} words or more.)
D15..D0	Almost Empty Value Number of words from FIFO Empty when the Almost Empty Flag will be asserted

2.1.6 Sample Clock Divider: Local Offset 0x0014

The Sample Clock Divider provides a divided down clock enable for COS and Logic Analyzer operation. The Rx Data signals are sampled based on an on-board clock - typically 20MHz. A value of Zero will cause full speed board operation at the full Receiver Clock speed, Typically 20 Mhz. A value of 1 would provide 1 non-Enabled state between Clock Enables. A value of 2 would provide 2 non-Enabled states between clock enables, and so on. See Figure 1.2-1 Clock Division Operation.

Include the following in the GSC Standard Windows Device Driver to access this register:

```
# define SPL_TX_CLOCK_DIVISION_REG          5
```

D31..D16	Reserved
D15..D0	Sample Clock Divider Register

2.1.7 Rx FIFO Data / Tx Data Output : Local Offset 0x0018

The Rx FIFO Data Register allows the user to read received data stored in the Rx FIFO. The Tx Output Value allows data to be driven onto the Cable Data signals (for testing). The same register is used to access both the Receive FIFO (reads). and the Transmit Output (writes).

D31..D0	Read = Rx FIFO Data Write = Tx Output Data

2.1.8 Rx Data Input: Local Offset 0x001C (Read Only)

The Rx Data Input will read the current state of the Cable Data signals.

D31..D0	Rx Input Data The current state of the Rx Cable Data Signals

2.1.9 Logic Analyzer Trigger Word : Local Offset 0x0020

The Masked Cable Data will be compared to the Masked contents of this register to determine when to trigger the Logic Analyzer function. All Masked bits, '1' in the mask, must match for the Logic Analyzer to Trigger. All non-masked bits in the mask, '0' in the mask, are considered Don't Cares. A Mask of 0x00000000 is all Don't cares and will trigger on the First word when the Receiver is started regardless of the Logic Analyzer Trigger. A mask of 0x0ffffffff will force all 32 bits from the Cable to compare to the 32 bits in the Logic Analyzer Trigger Register. Default is '0' . This register has replaced the Tx Row Valid Counter.

D31..D0	Logic Analyzer Trigger Word

2.1.10 Rx Data Mask Word : Local Offset 0x0024

When the board is in COS mode, this Mask Register will be AND'ed bit for bit with the incoming cable data. Only bits with a '1' in the Mask will be used to determine if a COS has occurred. All bits with a '0' in the Mask will be ignored while determining COS. The Un-Masked cable Data is what will be loaded into the FIFO.

When the board is in LA mode, this mask register will be AND'ed bit for bit with the incoming cable data and the LA Trigger register. The AND'ed results must all match exactly for the LA trigger to occur. All bits where the mask is '1' will be compared. All bits where the mask is a '0' will be considered 'Don't Cares' for determining the Trigger Event.

D31..D0	COS and Logic Analyzer Data Mask.Word .

Note:: A LA mask of all zero's will be considered All 'Don't Cares' and will trigger as soon as the Receiver is started. A COS Mask of all zero's will ignore all bits and NO COS changes will ever be detected.

2.1.11 Event Counter Initial Value : Local Offset 0x0028

When the Receiver is started in either COS or Logic Analyzer modes then this register will be used to load a 32 bit down counter. Every time a Change of State is detected using the COS Mask Register then this counter will be Decremented. When the Counter reaches Zero the Counter will be halted and the Rx Event Counter EQ0 status will be sent to the Board Status and Interrupt Control Registers. The Interrupt Control Register can be used to generate an interrupt if enabled by the host processor. Reading this location will read the contents of the Register, NOT the contents of the Counter. Default is ''0'.

NOTE: This counter will still count Change of States even if the board is collecting data in Logic Analyzer mode..

D31..D0	Event Counter Initial Value

2.1.12 Receive Word Count: Local Offset 0x002C

When the Receiver is started in either COS or Logic Analyzer modes then this counter will be reset to '0'. Every time a word is loaded into the Receive FIFO then this counter will be incremented. Default is ''0'.

D31..D0	Counter 0..31

	The number of D32 Words received in this message. It is reset to 0 when the Receiver is started.

2.1.13 FIFO Size: Local Offset 0x0044

The FIFO Size Register displays the size of the mplemented data FIFOs. This value, along with the FIFO Count Registers, may be used to determine the amount of data which can be safely transferred without over-running (or under-running) the FIFOs.

D31..D0	FIFO Depth of on-board FIFO

2.1.14 FIFO Word Count: Local Offset 0x0048

The FIFO Word Count Register displasy the current number of words in each FIFO. This value, along with the FIFO Size Register, may be used to determine the amount of data which can be safely transferred without over-running (or under-running) the FIFOs.

D31..D0	The number of D32 Words currently in the Receive FIFO

2.1.15 Interrupt Registers

There are 32 on-board interrupt sources (in addition to PLX interrupts) which may be individually enabled. Four interrupt registers control the on-board interrupts – Interrupt Control, Interrupt Status, Interrupt Edge/Level, and Interrupt Hi/Lo. The 32 Interrupt sources are:

IRQ #	Source	Default Level
IRQ0	Cable Command 1 Rising Edge	Rising Edge
IRQ1	Cable Command 1 Falling Edge	Rising Edge
IRQ2	Cable Command 2	Level Hi
IRQ3	Cable Command 3	Level Hi
IRQ4	Cable Command 4	Level Hi
IRQ5	Cable Command 5	Level Hi
IRQ6	Cable Command 6	Level Hi
IRQ11-7	Unused	----
IRQ12	FIFO Not Empty	Level Hi
IRQ13	FIFO Not Almost Empty	Level Hi
IRQ14	FIFO Almost Full	Level Hi
IRQ15	FIFO Full	Level Hi
IRQ16	FIFO Underflow	Level Hi
IRQ17	FIFO Overflow	Level Hi
IRQ18	Unused	----
IRQ19	Receiver Stopped	Level Hi
IRQ20	Logic Analyzer Trigger	Level Hi
IRQ21	Receiver Running	Level Hi
IRQ22	Event Counter Equal Zero	Level Hi
IRQ23	COS Event Detected	Level Hi
IRQ31-24	Unused	----

For all interrupt registers, the IRQ source (IRQ31:IRQ0) will correspond to the respective data bit (D31:D0) of each register. (D0 = IRQ0, D1 = IRQ1, etc.)

All Interrupt Sources share a single interrupt request back to the PCI9080 PLX chip.

2.1.15.1 Interrupt Control: Local Offset 0x0060

The Interrupt Control register individually enables each interrupt source. A '1' enables each interrupt source; a '0' disables. An interrupt source must be enabled for an interrupt to be generated.

2.1.15.2 Interrupt Status/Clear: Local Offset 0x0064

The Interrupt Status Register shows the status of each respective interrupt source. If an interrupt source is enabled in the Interrupt Control Register, a '1' in the Interrupt Status Register indicates the respective interrupt has occurred. The interrupt source will remain latched until the interrupt is cleared, either by writing to the Interrupt Status/Clear Register with a '1' in the respective interrupt bit position, or the interrupt is disabled in the Interrupt Control register. If an interrupt source is not asserted or the interrupt is not enabled, writing a '1' to that bit in the Interrupt Status/Clear Register will have no effect on the interrupt.

If the interrupt source is a level triggered interrupt, the interrupt status may still be '1' even if the interrupt is disabled. This indicates the interrupt condition is true, regardless of whether the interrupt is enabled. Likewise, if a level interrupt is enabled and the interrupt source is true, the interrupt status will be reasserted immediately after clearing the interrupt, and an additional interrupt will be requested.

2.1.15.3 Interrupt Edge/Level: Local Offset 0x0050

This register defines each interrupt source as a level or edge triggered. All interrupt sources are level triggered by default.

2.1.15.4 Interrupt Hi/Lo: Local Offset 0x0054

The Interrupt Hi/Lo Register define each interrupt source as active hi / active lo (Level) or rising edge / falling edge (Edge).

Receiver Operation:

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

- COS or Logic Analyzer Operation.
- COS Mask Register – Also Functions as the Logic Analyzer Mask Register.
- Software Programmable Event Counter.
- Logic Analyzer Trigger Register.
- Interrupts available on Software Selectable Options.
- Logic Analyzer Triggered.
- COS Detected.
- FIFO Status Flags.
- Event Counter = 0.
- COS or Logic Analyzer Stopped.

The Receiver will also provide a count of the number of D32 Words received and loaded into the FIFO. both the COS and Logic Analyzer functions will run until the Receive FIFO status is detected as Almost Full. The Almost Full flag is programmable under software control.

When the board is running in Change of State mode, it will compare the Masked inputs from the cable to the Masked results stored in an onboard register. When a difference is detected, the COS Receiver will store the Changed word into the Receive FIFO, Increment the Number of Words Received Counter, and Decrement the Event Counter. When the Event Counter has decremented to Zero, This status is available to the host processor and can be used to generate an interrupt to the host processor. The COS receiver will continue to collect data until the Receive FIFO becomes Almost Full, or until stopped by the host processor by removing the Start Receiver bit in the board control register. When stopped by the Receive FIFO becoming Almost Full, the receiver will generate a Receiver Stopped Status that is readable thru the Board Status Register and can be used to generate an interrupt to the host processor.

When the board is running in Logic Analyzer mode, it will compare the Masked inputs from the Cable to the Masked Logic Analyzer Trigger Register. If all Masked bits match then the board will enter the LA Triggered State and Begin collecting data. The Cable data word that generated the Trigger will be the first word stored into the Receive FIFO. The Logic Analyzer will store every data word on every clock divided cycle until the FIFO becomes Almost Full or until stopped by the host processor by removing the Enable Receiver bit in the Board Control Register. When stopped by the Receive FIFO becoming Almost Full, the receiver will generate a Receiver Stopped Status that is readable thru the Board Status Register and can be used to generate an interrupt to the host processor.

Rx Command Logic Analyzer Mode. – BCR D16

When this bit is a ‘1’ the board will function in Logic Analyzer mode. When this bit is a ‘0’ then this board will function in COS mode. `Default is ‘0’.

Rx Clock Divider Register – Offset 0x014 – 16 Bits.

The Receive Clock Divider Register will provide a divided down clock enable for COS and Logic Analyzer operation. A value of Zero will cause full speed board operation at the full Receiver Clock speed, Typically 20 Mhz. A value of 1 would provide 1 non-Enabled state between Clock Enables. A value of 2 would provide 2 non-Enabled states between clock enables, and so on. See Figure 1.2-1 Clock Division Operation. In order to access this register using the Standard Windows NT Device Driver you must include the following Definition in your Source Code.

```
#define SPL_TX_CLOCK_DIVISION_REG          5
```

You can then access this register using this register Define.

Rx Receive Cable Data Port- Offset 0x01C – 32 Bits – Read Only.

This register will allow the Host processor to read the current state of the Receive Cable Data which is the data the Receiver has just clocked in from the Data Cable. A Floating Cable will read thru the Rs-422 Transceivers as a logic '1'. This register has replaced the Tx Status Block Length Counter.

Rx Logic Analyzer Trigger - Offset 0x020 – 32 Bits

The Masked Cable Data will be compared to the Masked contents of this register to determine when to trigger the Logic Analyzer function. All Masked bits, '1' in the mask, must match for the Logic Analyzer to Trigger. All non-masked bits in the mask, '0' in the mask, are considered Don't Cares. A Mask of 0x00000000 is all Don't cares and will trigger on the First word when the Receiver is started regardless of the Logic Analyzer Trigger. A mask of 0x0ffffffff will force all 32 bits from the Cable to compare to the 32 bits in the Logic Analyzer Trigger Register. Default is '0' . This register has replaced the Tx Row Valid Counter.

Rx Data Mask Register - Offset 0x024 – 32 Bits

When the board is in COS mode, this Mask Register will be AND'ed bit for bit with the incoming cable data. Only bits with a '1' in the Mask will be used to determine if a COS has occurred. All bits with a '0' in the Mask will be ignored while determining COS. The Un-Masked cable Data is what will be loaded into the FIFO.

When the board is in LA mode, this mask register will be AND'ed bit for bit with the incoming cable data and the LA Trigger register. The AND'ed results must all match exactly for the LA trigger to occur. All bits where the mask is '1' will be compared. All bits where the mask is a '0' will be considered 'Don't Cares' for determining the Trigger Event.

Default is '0x0ffffffff'. This register has replaced the Tx Line Invalid Counter.

NOTE: A LA mask of all zero's will be considered All 'Don't Cares' and will trigger as soon as the Receiver is started. A COS Mask of all zero's will ignore all bits and NO COS changes will ever be detected.

Rx Event Counter Register - Offset 0x028 – 32 Bits

When the Receiver is started in either COS or Logic Analyzer modes then this register will be used to load a 32 bit down counter. Every time a Change of State is detected using the COS Mask Register then this counter will be Decremented. When the Counter reaches Zero the Counter will be halted and the
Rx Event Counter EQ0 status will be sent to the Board Status and Interrupt Control Registers. The Interrupt Control Register can be used to generate an interrupt if enabled by the host processor. Reading this location will read the contents of the Register, NOT the contents of the Counter. Default is ''0'. This Counter has replaced the Rx Status Block Length Counter.

The Ability to Read back the Counter has been added with FW Revision 01. The Actual Value in the counter can be read back thru Register Offset 0x0C, which was the TX FIFO ALMOST FLAGS.

NOTE: This counter will still count Change of States even if the board is collecting data in Logic Analyzer mode.

Rx Number of Words Received - Offset 0x02C – 32 Bits – Read Only

When the Receiver is started in either COS or Logic Analyzer modes then this counter will be reset to ‘0’. Every time a word is loaded into the Receive FIFO then this counter will be incremented. Default is ‘’0’. This counter is also known as the Rx Row Length Counter.

Rx Receiver Stopped Operation Notes.

When the Receiver is running in either COS or LA modes and the Receive FIFO goes Almost Full then all data collection will Stop and the Receiver will enter the Receiver Stopped state. Due to Timing Delays from the FIFO, Thru the FPGA, and Back to the FIFO again, the Receiver is not able to Stop Instantly. When the board is operating at high speed this will cause extra words to be loaded into the FIFO. This is most noticeable in Logic Analyzer mode, so this discussion will deal with Stopping when in Logic Analyzer Mode.

It takes 2 clock cycles for the Receiver to Actually shut down when running at full speed. When the Clock Division register is 0x0000, 2 extra words will be loaded into the FIFO. When the Clock Division register is 0x0001, 1 extra word will be loaded into the FIFO. When the Clock Division Register is 0x02 or higher then no extra words will be loaded into the Receive FIFO. The extra words will be counted in the Number of Words Received Counter.

For Example: If the Default FIFO Almost Full setting of 0x08 is used with the Default FIFO’s, 32Kbytes (0x08000) in Depth,

Table -1 Clock Division Vs Number of Words Loaded into FIFO

Clock Division	Number of Words Loaded into Receive FIFO
0	0x07ffa
1	0x07ff9
2	0x07ff8
3	0x07ff8
All other Values	0x07ff8

When the Board is operating in COS, Change of State, mode the same conditions apply, but the receiver will only try to load data into the FIFO if a Change of State were detected during the 2 clock cycle shutdown time. The Number of Words Received Counter will always contain the correct value and should be checked if the board has to Stop because the FIFO has become Almost Full.

Rx Event Counter - Offset 0x0C – 32 Bits

When the Receiver is started in either COS or Logic Analyzer modes then this register will be used to load a 32 bit down counter. Every time a Change of State is detected using the COS Mask Register then this counter will be Decremented. When the Counter reaches Zero the Counter will be halted and the
Rx Event Counter EQ0 status will be sent to the Board Status and Interrupt Control Registers. The Interrupt Control Register can be used to generate an interrupt if enabled by the host processor. When the counter has reached Zero, it will remain at Zero until the Receiver is stopped. Reading this location will read the contents of the Counter, to read the Register that this counter is loaded from, read location 0x028. Default is ‘’0’. This Counter has replaced the Rx FIFO ALMOST FLAGS.

The Ability to Read back the Counter has been added with FW Revision 01. The Actual Value in the counter can be read back thru Register Offset 0x0C, which was the TX FIFO ALMOST FLAGS.

NOTE: This counter will still count Change of States even if the board is collecting data in Logic Analyzer mode.

Rx COS Event Detected - BCR D26

The Rx COS Event Detected status bit will be set whenever the Receiver is running and Detects a masked event that is stored into the FIFO. Once set, this status flag will remain stay set until there is a Write to the Interrupt Status Register. This status bit can be used to generate Interrupts thru the Interrupt Control Register and the Interrupt Status Register bit D23.

Rx Event Count EQ Zero - BCR D25

The Rx Event Count EQ Zero status bit will be set whenever the Event Counter is Zero. When the Receiver is not running, then the Event Counter will be continuously loaded from the Contents of the Event Counter Register. When the Counter is Non Zero, this status bit will be a Zero. When the Counter has been decremented to Zero, then this status bit will be Active and the Counter will be held at Zero until the receiver is stopped. Writing to the Interrupt Status Register will not affect this bit. This status bit can be used to generate Interrupts thru the Interrupt Control Register and the Interrupt Status Register bit D22.

INITIALIZATION

Initializing the PMC-HPDI32B-COS 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:

all cable data transceivers will be in their default state, Tri-state off
all interrupts will be disabled;
the FIFOs will be empty;
the receive logic will be disabled.
the transmit logic will be disabled.
the board will be driving Cable Command 5 and Cable Command 6.

RESETS

There are two bits located in the Board Control Register on this board that are used as resets to the COS logic. 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 the reset is performed. For further details on the resets refer to the PMC-HPDI32B-COS Register Map, See Table 2.1-1.

D0	**Board Reset** will reset the local logic, clear the FIFOs, and place the appropriate registers into a known state.
D2	**FIFO Reset** 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 2), or a software board reset (Board Control Register bit 0).

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 to program the FIFO Almost Flags. 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 0x00080008 which will give you levels of
Almost Empty – 0x0008 – 8 D32 words above empty.
Almost Full – 0x0008 – Almost Full is 8 D32 words below Full.

In addition there are 2 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 – – Not Used.

The Rx FIFO events are cleared by a Rx FIFO Reset.

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.

3.0 Introduction

This section addresses common programming questions when developing an application for the SIO4. General Standards has developed software libraries to simplify application development. These libraries handle many of the low-level issues described below, including Resets, FIFO programming, and DMA. These libraries may default the board to a "standard" configuration (one used by most applications), but still provide low-level access so applications may be customized. The following sections describe the hardware setup in detail for common programming issues.

3.1 Resets

Each serial channel provides control for three unique reset sources: a USC Reset, a Transmit FIFO Reset, and a Receive FIFO Reset. All three resets are controlled from the GSC Channel Control/Status Registers. In addition, a Board Reset has been implemented in the Board Control Register. This board reset will reset all local registers to their default state as well as reset all FIFOs and USCs (all channels will be reset).

It is important to realize that since each Zilog Z16C30 chip contains two serial channels, a USC Reset to either channel will reset the entire chip (both channels affected). Due to the limitation of a USC Reset to affecting two channels, it is recommended that a single USC Channel be Reset via the RTReset bit of the USC Channel Command/Address Register (CCAR).

The FIFO resets allow each individual FIFO (Tx and Rx) to be reset independently. Setting the FIFO reset bit will clear the FIFO immediately.

3.2 FIFOs

Deep transmit and receive FIFOs are the key to providing four high speed serial channels without losing data. Several features have been implemented to help in managing the on-board FIFOs. These include FIFO flags (Empty, Full, Almost Empty and Almost Full) presented as both real-time status bits and interrupt sources, and individual FIFO counters to determine the exact FIFO fill level. DMA of data to/from the FIFOs provides for fast and efficient data transfers.

A single memory address is used to access both transmit and receive FIFOs for each channel. Data written to this memory location will be written to the transmit FIFO, and data read from this location retrieves data from the receive FIFO. Individual resets for the FIFOs are also provided in the Channel Control/Status Register.

3.2.1 FIFO Flags

Four FIFO flags are present from each on-board FIFO: FIFO Empty, FIFO Full, FIFO Almost Empty, and FIFO Almost Full. These flags may be checked at any time from the Channel Control/Status Register. Note these flags are presented as active low signals ('0' signifies condition is true). The Empty and Full flags are asserted when the FIFO is empty or full, respectively. The Almost Empty and Almost Full flags are software programmable such that they may be asserted at any desired fill level. This may be useful in determining when a data transfer is complete or to provide an indicator that the FIFO is in danger of overflowing and needs immediate service.

The Almost Flag value represents the number of bytes from each respective "end" of the FIFO. The Almost Empty value represents the number of bytes from empty, and the Almost Full value represents the number of bytes from full (NOT the number of bytes from empty). For example, the default value of "0x0007 0007" in the FIFO Almost Register means that the Almost Empty Flag will indicate when the FIFO holds 7 bytes or fewer. It will transition as the 8th byte is read or written. In this example, the Almost Full Flag will indicate that the FIFO contains (FIFO Size – 7) bytes or more. For the standard 32Kbyte FIFO, an Almost Full value of 7 will cause the Almost Full flag to be asserted when the FIFO contains 32761 (32k – 7) or more bytes of data .

The values placed in the FIFO Almost Registers take effect immediately, but should be set while the FIFO is empty (or the FIFO should be reset following the change). Note that this is a little different than the method for FIFO Flag programming which has previously been implemented on SIO4 boards. No FIFO programming delay is necessary.

3.2.2 FIFO Counters

The FIFO Size and FIFO count registers can be used to determine the exact amount of data in a FIFO as well as the amount of free space remaining in a FIFO. The size of each FIFO is auto-detected following a board reset. Real-time FIFO counters report the exact number of data words currently in each FIFO. By utilizing this information, the user can determine the exact amount of data which can safely be transferred to the transmit FIFOs or transferred from the receive FIFO. This information should help streamline data transfers by eliminating the need to continuously check empty and full flags, yet still allow larger data blocks to be transferred.

3.2.3 FIFO Size

In some applications, 4Kbyte FIFOs may be all that is required to implement a serial interface. This typically includes baud rates slower than 500kbps, or applications where the transfer size is limited to less than 4k bytes at a time (and an effective throughput rate less than 500kpbs). For these applications, a PMC-HPDI32B-COS board should be adequate. For faster applications, deeper external FIFO are required to ensure no data will be lost. Please contact General Standards if you have any questions about determining which FIFO size may be necessary for a specific application.

3.6 Loopback Modes

For normal operation, the Cable Transceiver Enable bit of the Pin Source Register will turn on the cable transceivers, and the DTE/DCE Mode bit will set the transceiver direction. These bits must be set before any data is transmitted over the user interface.

Additionally, there are several ways to loopback data to aid in debug operations. Data may be physically looped back externally by connecting one channel to another. For DB25 cable applications, this simple loopback method will requires a gender changer to connect one channel to another. One channel will be set to DTE mode, the other to DCE mode. Data sent from one channel will be received on the other.

An External Loopback mode (External Loopback bit set in the Pin Source Register) is also provided to loop back data on the same channel without requiring any external cabling. In this mode, the DTE/DCE mode will control the location for the transmit signals (TxC, TXD, RTS), and the receive signals will use these same signals as the receive inputs. Since signals are transmitted and received through the transceivers, this mode allows the setup to be verified (including signal polarity) without any external connections. Since external signals could interfere with loopback operation, all cables should be disconnected when running in external loopback mode.

An Internal Loopback Mode is also provided which loops back on the same channel internal to the board. This provides a loopback method which does not depend on DTE/DCE mode or signal polarity. This can remove cable transceiver and signal setup issues to aid in debugging.. If the Cable Transceivers are enabled, the transmit data will still appear on the appropriate transmit pins (based on DTE/DCE Mode setting). The Pin Status register will not reflect internally looped back signals, only signals to/from the transceivers.

3.9 General Purpose IO

Unused signals at the cable may be used for general purpose IO. The Pin Source and Pin Status Registers provide for simple IO control of all the cable interface signals. For outputs, the output value is set using the appropriate field in the Pin Source Register. All inputs can be read via the Pin Status register.

3.10 Interrupts

The HPDI32B-COS has a number of interrupt sources which are passed to the host CPU via the PCI Interrupt A. Since there is only one physical interrupt source, the interrupts pass through a number of “levels” to get multiplexed onto this single interrupt. The interrupt originates in the PCI9080 PCI Bridge, which combines the internal PLX interrupt sources (DMA) with the local space interrupt. The driver will typically take care of setting up and handling the PCI9080 interrupts. The single Local Interrupt is made up of the interrupt sources described in Section 2.1.10. In addition, the Zilog USC contains a number of interrupt sources which are combined into a single Local Interrupt. The user should be aware that interrupts must be enabled at each level for an interrupt to occur. For example, if a USC interrupt is used, it must be setup and enabled in the USC, enabled in the GSC Firmware Interrupt Control Register, and enabled in the PCI9080. In addition, the interrupt must be acknowledged and/or cleared at each level following the interrupt.

3.11 PCI DMA

The PCI DMA functionality allows data to be transferred between host memory and the HPDI32B-COS onboard FIFOs with the least amount of CPU overhead. The PCI9080 bridge chip handles all PCI DMA functions, and the device driver should handle the details of the DMA transfer. (Note: DMA refers to the transfer of Data from the on-

board FIFOs over the PCI bus. This should not be confused with the DMA mode of the USC – transfer of data between the USC and the on-board FIFOs. This On-Board DMA is setup by the driver and should always be enabled).

There are two PCI DMA modes – Demand Mode DMA and Non-Demand Mode DMA. Demand Mode DMA refers to data being transferred on demand. For receive, this means data will be transferred as soon as it is received into the FIFO. Likewise, for transmit, data will be transferred to the FIFOs as long as the FIFO is not full. The disadvantage to Demand Mode DMA is that the DMA transfers are dependent on the user data interface. If the user data transfer is incomplete , the Demand mode DMA transfer will also stop. If a timeout occurs, there is no way to determine the exact amount of data transferred before it was aborted.

Non-Demand Mode DMA does not check the FIFO empty/full flags before or during the data transfer – it simply assumes there is enough available FIFO space to complete the transfer. If the transfer size is larger than the available data, the transfer will complete with invalid results. This is the preferred mode for DMA operation. The FIFO Counters may be used to determine how much space is available for DMA so that the FIFO will never over/under run. Demand Mode DMA requires less software control, but runs the risk of losing data due to an incomplete transfer. The GSC library uses this method (Non-Demand DMA and checking the FIFO counters) as the standard transfer method.

CHAPTER 4: PCI INTERFACE

4.0 PCI Interface Registers

The PMC/PCI interface is handled by a PCI9080 I/O Accelerator from PLX Technology. The PCI interface is compliant with the 5V, 66MHz 32-bit PCI Specification 2.2. The PCI9080 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 264MB/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 PCI9080 are not utilized in this design, it is beyond the scope of this document to duplicate the PCI9080 User's Manual. Only those features, which will clarify areas specific to the PMC-HPDI32B-COS are detailed here. Please refer to the PCI9080 User's Manual (See Related Publications) for more detailed information. Note that the BIOS configuration and software driver will handle most of the PCI9080 interface. Unless the user is writing a device driver, the details of this PCI Interface Chapter may be skipped.

4.1 PCI Registers

The PLX 9080 contains many registers, many of which have no effect on the HPDI32B-COS performance. The following section attempts to filter the information from the PCI9080 manual to provide the necessary information for a HPDI32B-COS specific driver.

The HPDI32B-COS uses an on-board serial EEPROM to initialize many of the PCI9080 registers after a PCI Reset. This allows board specific information to be preconfigured correctly.

4.1.1 PCI Configuration Registers

The PCI Configuration Registers allow the PCI controller to identify and control the cards in a system.

PCI device identification is provided by the Vendor ID/Device ID (Addr 0x0000) and Sub-Vendor ID/Sub-Device ID Registers (0x002C). The following definitions are unique to the General Standards HPDI32B-COS boards. All drivers should verify the ID/Sub-ID information before attaching to this card. These values are fixed via the Serial EEPROM load following a PCI Reset, and cannot be changed by software.

Vendor ID	0x10B5	PLX Technology
Device ID	0x9080	PCI9080
Sub-Vendor ID	0x10B5	PLX Technology
Sub-Device ID	0x3198	GSC HPDI32B-COS

The configuration registers also setup the PCI IO and Memory mapping for the HPDI32B-COS. The PCI9080 is setup to use PCIBAR0 and PCIBAR1 to map the internal PLX registers into PCI Memory and IO space respectively. PCIBAR2 will map the Local Space Registers into PCI memory space, and PCIBAR3 is unused. Typically, the OS will configure the PCI configuration space.

For further information of the PCI configuration registers, please consult the PLX Technology PCI9080 Manual.

4.1.2 Local Configuration Registers

The Local Configuration registers give information on the Local side implementation. These include the required memory size. The HPDI32B-COS memory size is initialized to 4k Bytes. All other Local Registers initialize to the default values described in the PCI9080 Manual.

4.1.3 Runtime Registers

The Runtime registers consist of mailbox registers, doorbell registers, and a general-purpose control register. The mailbox and doorbell registers are not used and serve no purpose on the HPDI32B-COS. All other Runtime Registers initialize to the default values described in the PCI9080 Manual.

4.1.4 DMA Registers

The Local DMA registers are used to setup the DMA transfers to and from the on-board FIFOs. DMA is supported only to the four FIFO locations. The HPDI32B-COS supports both Demand (DREQ# controlled) and Non-Demand mode DMA. Both Channel 0 and Channel 1 DMA are supported.

4.1.4.1 DMA Channel Mode Register: (PCI 0x80 / 0x94)

The DMA Channel Mode register must be setup to match the hardware implementation

Bit	Description	Value	Notes
D1:0	Local Bus Width	11 = 32 bit 00 = 8 bit	Although the serial FIFOs only contain 8 bits of data, the register access is still a 32bit access. It is possible to "pack" the data by setting the Local Bus Width to 8, but this is only guaranteed to work with Non-Demand Mode DMA
D5:2	Internal Wait States	0000 = Unused	
D6	Ready Input Enable	1 = Enabled	
D7	Bterm# Input Enabled	0 = Unused	
D8	Local Burst Enable	1 = Supported	Bursting allows fast back-to-back accesses to the FIFOs to speed throughput
D9	Chaining Enable (Scatter Gather DMA)	X	DMA source addr, destination addr, and byte count are loaded from memory in PCI Space.
D10	Done Interrupt Enable	X	DMA Done Interrupt
D11	Local Addressing Mode	1 = No Increment	DMA to/from FIFOs only
D12	Demand Mode Enable	X	Demand Mode DMA is supported for FIFO accesses on the HPDI32B-COS. (See Section 3.3)
D13	Write & Invalidate Mode	X	
D14	DMA EOT Enable	0 = Unused	
D15	DMA Stop Data Transfer Enable	0 = BLAST terminates DMA	
D16	DMA Clear Count Mode	0 = Unused	
D17	DMA Channel Interrupt Select	X	
D31:18	Reserved	0	

CHAPTER 5: HARDWARE CONFIGURATION

5.0 Board Layout

The following figure is a drawing of the physical components of the PMC-HPDI32B-COS:

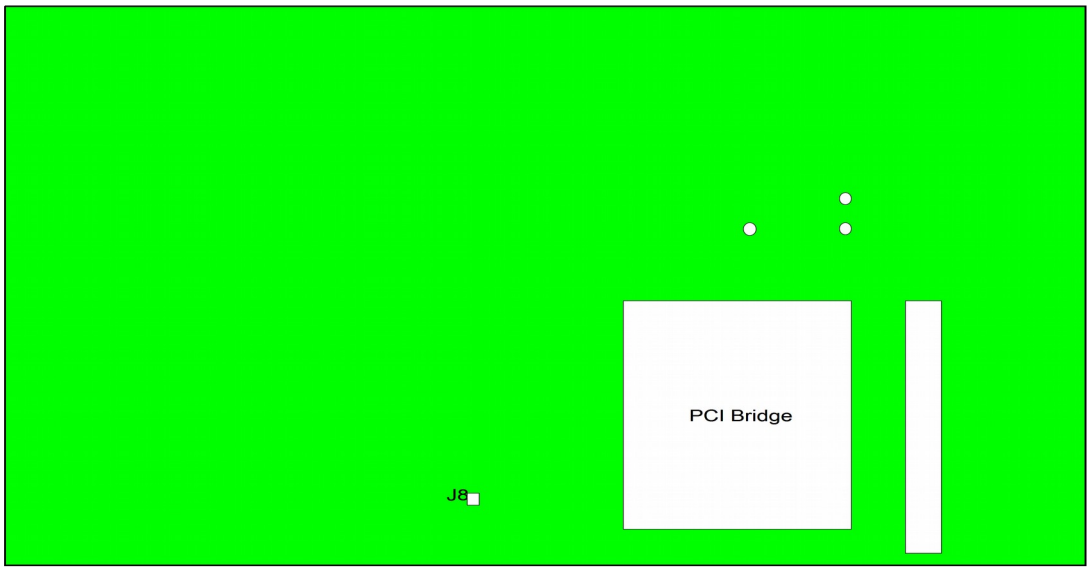


Figure 5-1: Board Layout – Top

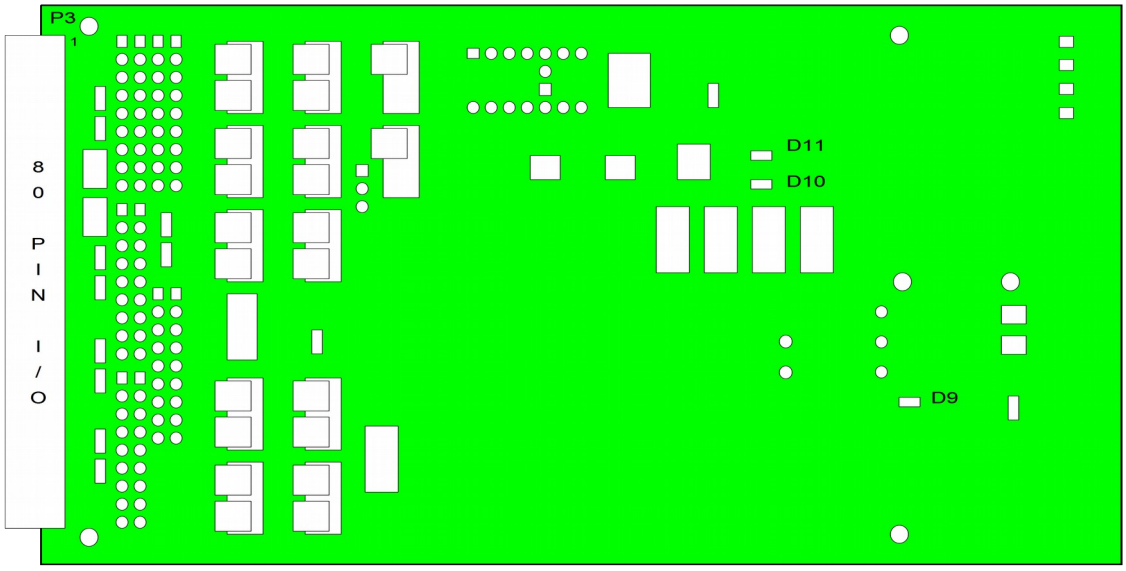


Figure 5-2: Board Layout - Bottom

5.1 Board ID Jumper J6

Jumper J6 allows the user to set the Board ID in the GSC Board Status Register (See Section 2.1.3). This is useful to uniquely identify a board if more than one HPDI32B-COS card is in a system. When the Board ID jumper is installed, it will read '1' in the Board Status Register. The Board Status Register bit will report '0' when the jumper is removed. Only Board ID 0 & 1 are used in the COS boardRefer to Figure 5-1 for Jumper J6 location.

J5 Jumper	Description	Notes
1 - 2	Board ID 0	Board ID 0 in Board Status Register (D16)
3 - 4	Board ID 1	Board ID 1 in Board Status Register (D17)
5 - 6	Unused	
7 - 8	Unused	

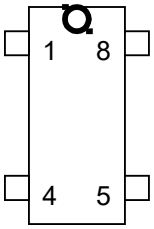
5.2 Termination Resistors

The PMC-HPDI32B-COS transceivers have parallel termination in termination resistors of 120 Ohms for the RS-422/RS485 interface. These 8 pin resistor SIPS are socketed so they may be easily customized if a different value is required. There are ten termination SIPs – RP4, RP5, RP8, RP9, RP13, RP14, RP17, RP18, RP19, and RP20.

Please contact quotes@generalstandards.com if a different termination value is required.

5.3 On-Board Sample Clock

The on-board oscillator, Y1, is used as the receive sample clock. This clock is used to sample the Cable data for the Receiver COS and Logic Analyzer functions. The oscillator is factory installed at 20MHz, but is socketed so it can be by the user. The maximum frequency supported by this clock is 32Mhz.



PIN	NAME
1	NC / OE*
4	GND
5	OUT
8	VCC

Figure 5-2: Oscillator

5.4 Interface Connectors

The PMC-HPDI32B-COS connects through the 80 pin Front panel connector P3. The pinout shows the connection for RS422/RS485.

PCB Connector: Robinson Nugent P50E-080-P1-SR1-TG.
Mating Connectors:

Robinson Nugent P50E-080-S-TG → 50 mil. , twisted pair. (RECOMMENDED)
Robinson Nugent P25E-080-S-TG → 25 mil., multi-drop capability, not twisted pair.

Pin No.	Cable Signal Name
1	Unused
2	Unused
3	CMD0 +
4	CMD0 -
5	CMD1 +
6	CMD1 -
7	CMD2 +
8	CMD2 -
9	CMD3 +
10	CMD3 -
11	CMD4 +
12	CMD4 -
13	TX ENABLE +
14	TX ENABLE -
15	RX ENABLE +
16	RX ENABLE -
17	D0 +
18	D0 -
19	D1 +
20	D1 -
21	D2 +
22	D2 -
23	D3 +
24	D3 -
25	D4 +
26	D4 -
27	D5 +
28	D5 -
29	D6 +
30	D6 -
31	D7 +
32	D7 -
33	D8 +
34	D8 -
35	D9 +
36	D9 -
37	D10 +
38	D10 -
39	D11 +
40	D11 -

Pin No.	Cable Signal Name
41	D12 +
42	D12 -
43	D13 +
44	D13 -
45	D14 +
46	D14 -
47	D15 +
48	D15 -
49	D16 +
50	D16 -
51	D17 +
52	D17 -
53	D18 +
54	D18 -
55	D19 +
56	D19 -
57	D20 +
58	D20 -
59	D21 +
60	D21 -
61	D22 +
62	D22 -
63	D23 +
64	D23 -
65	D24 +
66	D24 -
67	D25 +
68	D25 -
69	D26 +
70	D26 -
71	D27 +
72	D27 -
73	D28 +
74	D28 -
75	D29 +
76	D29 -
77	D30 +
78	D30 -
79	D31 +
80	D31 -

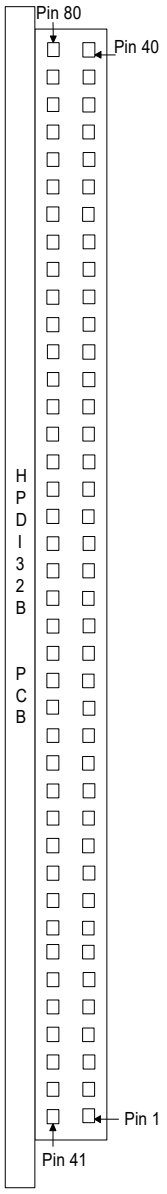


Table 5-1: RS485/RS422 P3 Pin-Out

CHAPTER 6: ORDERING OPTIONS

6.0 Ordering Information

PMC-HPDI32B-COS

Notes:

The on-board sample clock is 20MHz (default). Consult GSC for different clock options.
The FIFO Size is 8Kx32 (default). Consult GSC for different FIFO depth options.

Please consult our sales department with your application requirements to determine the correct ordering options. (quotes@generalstandards.com).

APPENDIX A: FIRMWARE REVISIONS

Since SIO4 boards can exist across multiple form factors and with various hardware features, the firmware/features registers attempt to help identify the exact version of a SIO4 board. This appendix provides a more detailed breakdown of what the firmware and features registers, and detail differences between the firmware revisions.

Firmware Register - Local Offset 0x00 (0xE2240113)

D31:16 HW Board Rev 0xE224 PMC-HPDI32B-COS Rev C

- D31 **1** = Features Register Present
- D30 **1** = Complies with this standard
- D29 **1** = 66MHz PCI bus interface
  0 = 33MHz PCI bus interface
- D28 1 = 64 bit PCI bus interface
  0 = 32 bit bus interface
- D27:D24 Form Factor
  0 = Reserved
  1 = PCI
  2 = PMC
  3 = cPCI
  4 = PC104P
- D23:D20 HW Board (sub-field of form factor)
  0 = PMC-SIO4AR
  1 = PMC-SIO4BX
  2 = PMC66-HPDI32B-COS
- D19:D16 HW Board Rev (lowest rev for firmware version)
  0=NR
  1=A,
  2=B
  3=C
  4=D

D15:8	Firmware Type ID	**0x01**	Std Firmware default
		0x04	Sync Firmware default
D7:0	Firmware Revision	XX	Firmware Version

0x13 - Add L3RIO support. Force Xcvr mode reset when channel reset.
0x14 - Update Ext FIFO counters to support <32K FIFOs. Add more sync code.
0x15 - Sync code release. Update FIFO Config option.