PMC-SIO4BX-SYNC

Hardware User's Manual

HIGH SPEED
QUAD CHANNEL
SYNCHRONOUS SERIAL IO CONTROLLER
WITH DEEP TRANSMIT AND RECEIVE FIFOS
AND MULTIPROTOCOL TRANSCEIVERS

RS-485
RS-422 / V.11
RS-423 / V.10
RS-232 / V.28

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 E

PREFACE

Revision History

1. Rev A – Feb 2004 – Original rev from PCI-SIO4B manual.
2. Rev B – Misc Updates
3. Rev C – Aug 2004 - Misc Updates
4. Rev D – Jun 2005 – update for v404 firmware
5. Rev E – Nov 2007 – update for v40C firmware

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.

All rights reserved

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

Copyright © 2007 General Standards Corporation

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 2.1 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

1 General Description

The General Standards PMC-SI04BX-SYNC board provides four, high-speed synchronous serial interface channels for PMC applications. The SIO4BX-SYNC combines a flexible serial/parallel converter, deep FIFO data buffers, and multiprotocol transceivers in four fully independent synchronous serial IO channels. These features, along with four programmable baud rate generators and a high performance PMC/PCI interface engine, give the PMC-SIO4BX-SYNC unsurpassed performance in a synchronous serial interface card.

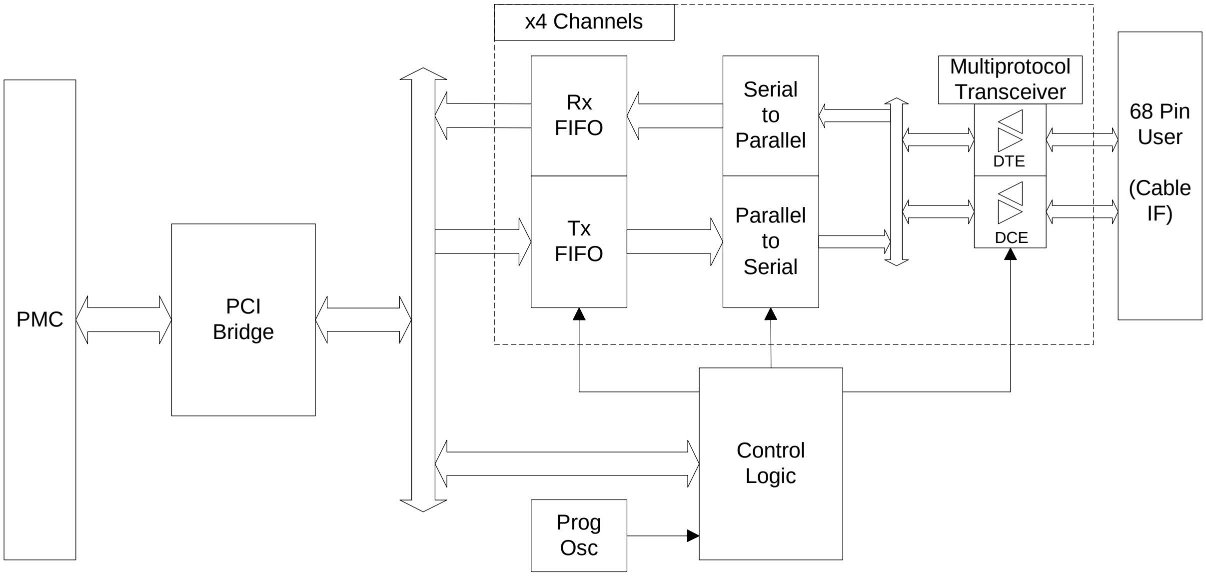


Figure 1-1 Block Diagram of PMC-SIO4BX-SYNC

- Four Multi-Protocol Synchronous Serial Channels with Separate Transmit and Receive Interfaces per Channel
- Synchronous Serial Data Rates up to 10 Mbits/sec
- Configurable for Two Signal (Clock and Data) or Three Signal (Clock, Data, and Envelope) Interface
- Software Selectable RS-422/RS-485, RS-232, or RS-423 interface on a per channel basis
- Fast RS422/RS485 Differential Cable Transceivers Provide Data Rate up to 10Mbps
- RS423 and RS232 Cable Transceivers Provide Data Rate up to 230kbps
- Independent Transmit and Receive FIFOs for Data Buffering for each Serial Channel – Up to 32k Deep Each
- Configurable Polarity on all Serial Interface Signals.
- NRZ and NRZB Data Encoding
- Programmable Transmit Word Bit Count allows transmit word lengths from 1 to 64k bits.
- Programmable Transmit Gap Bit Count allows number of clocks between words from 1 to 64k bits
- Four Programmable Oscillators provides Baud Rate Clock generation from 10Mbps to 400bps
- Bidirectional Signal Interface allows DTE or DCE configuration
- Industry standard SCSI II type 68 pin front edge I/O Connector
- Optional cable adapter splits 68 pin connectors into four separate DB25 connectors (one DB25 per channel).
- Unused signals may be reconfigured as general purpose IO.
- Dual PCI Master DMA Engine to speed transfers and minimize host I/O overhead
- Easily mounted on PCI, cPCI, or PC104P carriers to provide multiprotocol support for other form factors.
- A variety of device drivers are available, including VxWorks, WinNT, Win2k, Linux, and Labview

1.1 Serial Interface

The simple synchronous interface may be configured as a three signal interface - Clock, Data, and Envelope (Data Valid), or an even simpler two signal interface – Clock and Data. The SIO4BX-SYNC allows the serial interface to be further customized with the following user configurable options:

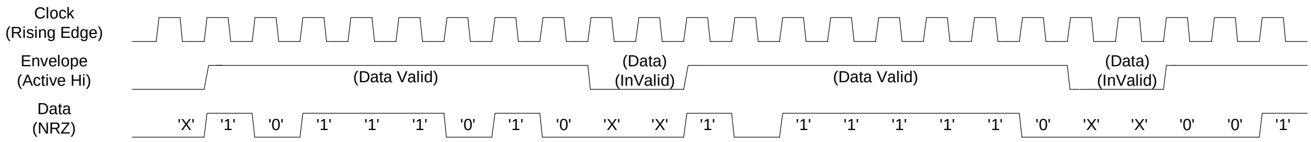
- Clocking Data on either rising or falling edge of the clock.
- Active Hi or Active Lo polarity for the Envelope Signal
- NRZ (Level) or NRZB (Inverted Level) Data Encoding
- Continuous Transmit Clock or Transmit Clock disabled when Data is invalid (Clock present only for valid Data).
- Transmit Word Size may be configured from 1 to 64k bits (consecutive bit count).
- Transmit Gap Size (number of clocks between transmit words) may be configured from 0 to 64k bits
- Data may be transmitted MSB first or LSB first (8-bit or less word size).
- Transmit Clock may be configured from 10MHz down to 400Hz on a per channel basis
- Auxiliary Clock Input from cable may be used as Transmit Clock

The following sections show some typical examples of how the SIO4BX-SYNC can be configured to support different two and three signal interfaces.

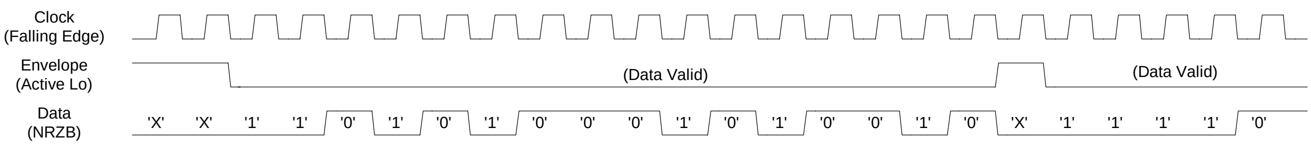
1.1.1 Three Signal Serial Interface

Figure 1-2 shows two examples of typical 3-signal interfaces. The two diagrams show how the card can be configured to handle different interface requirements. For the top diagram, Data and Envelope change on the rising edge of the Clock. The Data and Envelope are both Active Hi. The Clock is continuous – e.g. the Clock continues even when Data is Invalid (TxC Idl). Data is transmitted in 8 bit words (TxCount), with a two Clock ‘gap’ (TxGap) in between each word (Data Valid for 8 bits, Invalid for 2).

In the lower example, Data and Envelope change on the falling edge of the Clock. The Data and Envelope signals are both Active Lo. The Clock is still continuous – Clock continues even when Data is Invalid (TxC Idl). Data is transmitted in 16 bit words (TxCount), with a one Clock ‘gap’ (TxGap) in between each word (Data Valid for 16 bits, Invalid for 1).



TxCount = 8, TxGap = 2, TxC Idl = 0



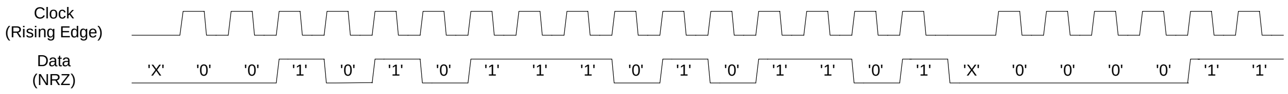
TxCount = 16, TxGap = 1, TxC Idl = 0

Figure 1-2 Three Signal Serial Interface

1.1.2 Two Signal Serial Interface

Figure 1-3 shows how the Clock can be used to qualify the Data to give a two signal serial interface. In this case, Data is considered valid at every Clock. In this example, Data is Active Hi and changes on the rising edge of the Clock. The Clock is not present when Data is invalid.(TxC Idl). Data is transmitted in 8 bit words (TxCount), with a two clock 'gap' (TxGap) in between each word (Data Valid for 8 bits, Invalid for 2).

In the lower example, Data and Envelope change on the falling edge of the Clock. The Data and Envelope signals are both Active Lo. The clock is still continuous – Clock continues even when Data is Invalid (TxC Idl). Data is transmitted in 16 bit words (TxCount), with a one Clock 'gap' (TxGap) in between each word (Data Valid for 16 bits, Invalid for 1).



TxCount = 16, TxGap = 1, TxC Idl = 1

Figure 1-3 Two Signal Serial Interface

1.2 Deep Transmit/Receive FIFOs

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 be one of three sizes: 512 bytes, 8 kbytes, or 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 SIO4BX-SYNC 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.

1.3 Multiprotocol Transceivers

The SIO4BX-SYNC Data is transferred over the user interface using high-speed multiprotocol transceivers. These multiprotocol transceivers are software selectable as RS422/RS485, RS423, or RS232 on a per channel basis. Each channel direction may also be configured as DTE or DCE configuration. This allows for either full duplex or half duplex configurations.

1.4 PMC/PCI Interface

The control interface to the SIO4BX-SYNC 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.

1.5 General Purpose IO

Since some signals may not be used in all applications, the SIO4BX provides the flexibility to configure all signals to be used as general purpose IO. Each channel also contains two dedicated General Purpose IO signals – TxSp/RxSp and TxAuxC/RxAuxC. In addition, all output signals may be forced to a Hi or Lo state. This also allows signals from unused channels to be available as general purpose IO.

1.6 Connector Interface

The SIO4BX provides a user IO interface through a front-side card edge connector. All four serial channels interface through this high-density, 68 pin SCSI II type connector, and are grouped to simplify separating the cable into four distinct serial connectors.

Standard cables are available from General Standards in various lengths to adapt the single 68 pin SCSII connector into four DB25 connectors (one per channel). A standard cable is also available with a single 68 pin SCSII connector on one end and open on the other. This allows the user to add a custom connector (or connect to a terminal block). General Standards will also work with customers to fabricate custom cables. Consult factory for details on custom cables.

CHAPTER 2: LOCAL SPACE REGISTERS

2 GSC Firmware (Local Space) Registers

The PMC-SIO4BX-SYNC is accessed through two sets of registers – PCI Registers and GSC Firmware Registers. The GSC Firmware Registers (referred to as Local Space Registers), which provide the control/status for the SIO4BX-SYNC board, are described below. The PCI registers (internal to the PLX 9080 PCI controller) are discussed in Chapter 3.

Offset Address	Size	Access*	Register Name	Default Value (Hex)
0x0000	D32	Read Only	Firmware Revision	C21104XX
0x0004	D32	Read/Write	Board Control	00000000
0x0008	D32	Read Only	Board Status	000000XX
0x000C	--	--	Reserved	00000000
0x0010	D32	Read/Write	Ch 1 Tx Almost Full/Empty	00070007
0x0014	D32	Read/Write	Ch 1 Rx Almost Full/Empty	00070007
0x0018	D32	Read/Write	Ch l Data FIFO	000000XX
0x001C	D32	Read/Write	Ch 1 Control/Status	0000CC00
0x0020	D32	Read/Write	Ch 2 Tx Almost Full/Empty	00070007
0x0024	D32	Read/Write	Ch 2 Rx Almost Full/Empty	00070007
0x0028	D32	Read/Write	Ch 2 Data FIFO	000000XX
0x002C	D32	Read/Write	Ch 2 Control/Status	0000CC00
0x0030	D32	Read/Write	Ch 3 Tx Almost Full/Empty	00070007
0x0034	D32	Read/Write	Ch 3 Rx Almost Full/Empty	00070007
0x0038	D32	Read/Write	Ch 3 Data FIFO	000000XX
0x003C	D32	Read/Write	Ch 3 Control/Status	0000CC00
0x0040	D32	Read/Write	Ch 4 Tx Almost Full/Empty	00070007
0x0044	D32	Read/Write	Ch 4 Rx Almost Full/Empty	00070007
0x0048	D32	Read/Write	Ch 4 Data FIFO	000000XX
0x004C	D32	Read/Write	Ch 4 Control/Status	0000CC00
0x0060	D32	Read/Write	Interrupt Control	00000000
0x0064	D32	Read/Write	Interrupt Status/Clear	00000000
0x0068	D32	Read Only	Interrupt Edge/Level	FFFFFFFF
0x006C	D32	Read/Write	Interrupt High/Low	FFFFFFFF
0x0070-0x007C	---	--	RESERVED	--------
0x0080	D32	Read/Write	Ch 1Pin Source	00000020
0x0084	D32	Read/Write	Ch 2 Pin Source	00000020
0x0088	D32	Read/Write	Ch 3 Pin Source	00000020
0x008C	D32	Read/Write	Ch 4 Pin Source	00000020
0x0090	D32	Read Only	Ch 1Pin Status	000000XX
0x0094	D32	Read Only	Ch 2 Pin Status	000000XX
0x0098	D32	Read Only	Ch 3 Pin Status	000000XX
0x009C	D32	Read Only	Ch 4 Pin Status	000000XX
0x00A0	D32	Read/Write	Programmable Osc RAM Addr	00000000
0x00A4	D32	Read/Write	Programmable Osc RAM Data	00000000
0x00A8	D32	Read/Write	Programmable Osc Control/Status	00000000
0x00AC	---	--	RESERVED	--------
0x00B0	D32	Read/Write	Ch1 TxCount / TxGap	00000000
0x00B4	D32	Read/Write	Ch2 TxCount / TxGap	00000000
0x00B8	D32	Read/Write	Ch3 TxCount / TxGap	00000000
0x00BC	D32	Read/Write	Ch4 TxCount / TxGap	00000000
0x00C0	D32	Read/Write	Ch1 RxCount	00000000
0x00C4	D32	Read/Write	Ch2 RxCount	00000000
0x00C8	D32	Read/Write	Ch3 RxCount	00000000

0x00CC	D32	Read/Write	Ch4 RxCount	00000000
0x00D0	D32	Read Only	Ch1 FIFO Count	00000000
0x00D4	D32	Read Only	Ch2 FIFO Count	00000000
0x00D8	D32	Read Only	Ch3 FIFO Count	00000000
0x00DC	D32	Read Only	Ch4 FIFO Count	00000000
0x00E0	D32	Read Only	Ch1 FIFO Size	XXXXXXXX
0x00E4	D32	Read Only	Ch2 FIFO Size	XXXXXXXX
0x00E8	D32	Read Only	Ch3 FIFO Size	XXXXXXXX
0x00EC	D32	Read Only	Ch4 FIFO Size	XXXXXXXX
0x00F0-0x00F8	---	--	RESERVED	--------
0x00FC	D32	Read Only	Features Register	000000XX

2.1 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.

D31:16	HW Board Rev	0xC211	PMC-SIO4BX Rev A
D15:8	Firmware Type ID	0x04	Sync Firmware
D7:0	Firmware Revision	XX	Firmware Version

2.2 Board Control: Local Offset 0x0004

The Board Control Register defines the general control functions for the board. The main function in this register defines the Demand mode DMA channel requests.

D31 Board Reset
1 = Reset all Local Registers and FIFOs to their default values
Notes: This bit will automatically clear to 0 following the board reset.
Board Reset will NOT reset programmable oscillator.
Following a Board Reset, ResetInProgress bit (D31) of the Board Status Register will remain set until the Board reset is complete;

D30 RESERVED (Debug Internal FIFO)
D29:9 RESERVED
D8 Rx FIFO Stop on Full
1 = If Rx FIFO becomes full, stop receiving data (disable receiver).
D7 Demand Mode DMA Channel 1 Single Cycle Disable
D6:4 Demand Mode DMA Channel 1 Request

D6	**D5**	**D4**	**Demand Mode DMA 1 Channel**
0	0	0	Channel 1 Rx
1	0	0	Channel 1 Tx
0	1	0	Channel 2 Rx
1	1	0	Channel 2 Tx
0	0	1	Channel 3 Rx
1	0	1	Channel 3 Tx
0	1	1	Channel 4 Rx
1	1	1	Channel 4 Tx

D3 Demand Mode DMA Channel 0 Single Cycle Disable
D2:0 Demand Mode DMA Channel 0 Request

D2	D1	D0	Demand Mode DMA 0 Channel
0	0	0	Channel 1 Rx
1	0	0	Channel 1 Tx
0	1	0	Channel 2 Rx
1	1	0	Channel 2 Tx
0	0	1	Channel 3 Rx
1	0	1	Channel 3 Tx
0	1	1	Channel 4 Rx
1	1	1	Channel 4 Tx

2.3 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 SIO4 boards are present in a system. Most other information contained in this register is for debug and configuration information only and will have no use to the typical user.

D31 Reset In Progress – following a Board Reset, this bit will remain set while the FIFO size is being detected (less than 2ms). No accesses (other than monitoring this bit) should be attempted until the Board reset has completed.
D30:D16 RESERVED
D15:D8 External FIFO Configuration

D15	External Ch4 Rx FIFO Present
D14	External Ch4 Tx FIFO Present
D13	External Ch3 Rx FIFO Present
D12	External Ch3 Tx FIFO Present
D11	External Ch2 Rx FIFO Present
D10	External Ch2 Tx FIFO Present
D9	External Ch1 Rx FIFO Present
D8	External Ch1 Tx FIFO Present

D7:D2 RESERVED
D1:D0. Board Jumpers are physical jumpers which can be used to distinguish between boards if multiple SIO4 boards are present in a system.
D1 Board Jumper 1
0 = Jumper J5:3-4 installed
D0 Board Jumper 0
0 = Jumper J5:1-2 installed

2.4 Channel TX Almost Flags: Local Offset 0x0010 / 0x0020 / 0x0030 / 0x0040

The Tx Almost Flag Registers are used to set the Almost Full and Almost Empty Flags for the transmit 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.

D31:16	TX Almost Full Flag 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.)
D15:0	TX Almost Empty Flag Value Number of words from FIFO Empty when the Almost Empty Flag will be asserted.

2.5 Channel Rx Almost Flags: Local Offset 0x0014 / 0x0024 / 0x0034 / 0x0044

The Rx Almost Flag Registers are used to set the Almost Full and Almost Empty Flags for the transmit 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.

D31:16	RX Almost Full Flag 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.)
D15:0	RX Almost Empty Flag Value Number of words from FIFO Empty when the Almost Empty Flag will be asserted

2.6 Channel FIFO: Local Offset 0x0018 / 0x0028 / 0x0038 / 0x0048

The Channel FIFO Register passes serial data to/from the serial controller. The same register is used to access both the Transmit FIFO (writes) and Receive FIFO (reads).

D31:8	RESERVED
D7:0	Channel FIFO Data

2.7 Channel Control/Status: Local Offset 0x001C / 0x002C / 0x003C / 0x004C

The Channel Control/Status Register provides the reset functions and data transceiver enable controls, and the FIFO Flag status for each channel.

D31 RESERVED

D30:24 Channel Control Bits

D30	Receive Gap Enable 0 = Bit D24 (Receive Enable) controls receiver enable 1 = Receiver won't be started unless Bit D24=1 and RxE is negated. This basically waits for a gap in the data stream before data reception begins.
D29	Receive Bit Count Reset 1 = Reset Receive Bit Counter
D28	Transmit MSB/LSB 0 = Transmit MSB first (default) 1 = Transmit LSB first

D27 Receive MSB/LSB

0 = Receive MSB first (default)
1 = Receive LSB first

D26 Stop Transmit On FIFO Empty

0 = Transmitter remains enabled under software control (D17)
1 = Transmitter will be disabled (D25 = ‘0’) if Tx FIFO becomes empty

D25 Transmit Enable

1 = Transmitter enabled. Note that cable transceiver direction should be set and transceivers enabled before the Transmit Enable is set.

D24 Receive Enable

1 = Receiver enabled. Note that cable transceiver direction should be set and transceivers enabled before the Receive Enable is set.

D23:17 RESERVED

D16:8 Channel Status Bits

D16 Rx FIFO Overflow (Latched)

1= Rx Data was lost due to Rx Overflow.
Note: This bit is latched. Write D16=1 to clear.

D15	Rx FIFO Full Flag Lo	(0 = Rx FIFO Full)
D14	Rx FIFO Almost Full Flag Lo	(0 = Rx FIFO Almost Full)
D13	Rx FIFO Almost Empty Flag Lo	(0 = Rx FIFO Almost Empty)
D12	Rx FIFO Empty Flag Lo	(0 = Rx FIFO Empty)
D11	Tx FIFO Full Flag Lo	(0 = Tx FIFO Full)
D10	Tx FIFO Almost Full Flag Lo	(0 = Tx FIFO Almost Full)
D9	Tx FIFO Almost Empty Flag Lo	(0 = Tx FIFO Almost Empty)
D8	Tx FIFO Empty Flag Lo	(0 = Tx FIFO Empty)

D7:2 RESERVED

D1:0 Channel FIFO Reset

D1 Reset Channel Rx FIFO (Pulsed)

Note: This value will automatically clear to ‘0’.

D0 Reset Channel Tx FIFO (Pulsed)

Note: This value will automatically clear to ‘0’.

2.8 Interrupt Registers

There are 32 on-board interrupt sources (in addition to PLX interrupts), each of 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 Interrupt sources are:

IRQ #	Source	Default Level	Alternate Level
IRQ0	Ch1 RxE	Rising Edge	Falling Edge
IRQ1	Ch1 Tx FIFO Almost Empty	Rising Edge	Falling Edge
IRQ2	Ch1 Rx FIFO Almost Full	Rising Edge	Falling Edge
IRQ3	Ch1 RxSp	Rising Edge	Falling Edge
IRQ4	Ch2 RxE	Rising Edge	Falling Edge
IRQ5	Ch2 Tx FIFO Almost Empty	Rising Edge	Falling Edge
IRQ6	Ch2 Rx FIFO Almost Full	Rising Edge	Falling Edge
IRQ7	Ch2 RxSp	Rising Edge	Falling Edge
IRQ8	Ch3 RxE	Rising Edge	Falling Edge
IRQ9	Ch3 Tx FIFO Almost Empty	Rising Edge	Falling Edge
IRQ10	Ch3 Rx FIFO Almost Full	Rising Edge	Falling Edge
IRQ11	Ch3 RxSp	Rising Edge	Falling Edge
IRQ12	Ch4 RxE	Rising Edge	Falling Edge
IRQ13	Ch4 Tx FIFO Almost Empty	Rising Edge	Falling Edge
IRQ14	Ch4 Rx FIFO Almost Full	Rising Edge	Falling Edge
IRQ15	Ch4 RxSp	Rising Edge	Falling Edge
IRQ16	Ch1 Tx FIFO Empty	Rising Edge	Falling Edge
IRQ17	Ch1 Tx FIFO Full	Rising Edge	Falling Edge
IRQ18	Ch1 Rx FIFO Empty	Rising Edge	Falling Edge
IRQ19	Ch1 Rx FIFO Full	Rising Edge	Falling Edge
IRQ20	Ch2 Tx FIFO Empty	Rising Edge	Falling Edge
IRQ21	Ch2 Tx FIFO Full	Rising Edge	Falling Edge
IRQ22	Ch2 Rx FIFO Empty	Rising Edge	Falling Edge
IRQ23	Ch2 Rx FIFO Full	Rising Edge	Falling Edge
IRQ24	Ch3 Tx FIFO Empty	Rising Edge	Falling Edge
IRQ25	Ch3 Tx FIFO Full	Rising Edge	Falling Edge
IRQ26	Ch3 Rx FIFO Empty	Rising Edge	Falling Edge
IRQ27	Ch3 Rx FIFO Full	Rising Edge	Falling Edge
IRQ28	Ch4 Tx FIFO Empty	Rising Edge	Falling Edge
IRQ29	Ch4 Tx FIFO Full	Rising Edge	Falling Edge
IRQ30	Ch4 Rx FIFO Empty	Rising Edge	Falling Edge
IRQ31	Ch4 Rx FIFO Full	Rising Edge	Falling Edge

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 FIFO interrupts are edge triggered active high. This means that an interrupt will be asserted (assuming it is enabled) when a FIFO Flag transitions from FALSE to TRUE (rising edge triggered) or TRUE to FALSE (falling edge). For example: If Tx FIFO Empty Interrupt is set for Rising Edge Triggered, the interrupt will occur when the FIFO transitions from NOT EMPTY to EMPTY. Likewise, if Tx FIFO Empty Interrupt is set as Falling Edge Triggered, the interrupt will occur when the FIFO transitions from EMPTY to NOT EMPTY.

All Interrupt Sources share a single interrupt request back to Local Interrupt Input of the PCI9080 PLX chip. This Local Interrupt input must be enabled in the PLX Interrupt Control/Status Register to be recognized as a PCI interrupt source. See Section **4.4 Interrupts** for further interrupt programming information.

2.8.1 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.

2.8.2 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. Clearing an interrupt which is not enabled or not asserted will have no effect.

2.8.3 Interrupt Edge/Level: Local Offset 0x0068

The Interrupt Edge Register is an information only (read only) register. This register can be used by a generic driver to determine if the interrupt source is edge or level triggered. All interrupt sources on the SIO4BX-SYNC are edge triggered.

2.8.4 Interrupt Hi/Lo: Local Offset 0x006C

The Interrupt Edge Register is an information only register which denotes all interrupt sources as edge triggered. The Interrupt Hi/Lo Register define each interrupt source as rising edge or falling edge. For example, a rising edge of the TX Empty source will generate an interrupt when the TX FIFO becomes empty. Defining the source as falling edge will trigger an interrupt when the TX FIFO becomes “NOT Empty”.

2.9 Channel Pin Source: Local Offset 0x0080 / 0x0084 / 0x0088 / 0x008C

The Channel Pin Source Register configures the function of the cable interface signals as well as controls the transceiver protocols.

31	30	29	28	27	26	25	24
Cable Xcvr Enable	X	Ext Loopback Enable	DCE/DTE Mode	Transceiver Protocol Mode			

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Int LB	XXX			TxD Idl	X	RxD Src		RxE Src		RxC Src	TxSp Src		TxAux Src		TxD Src			TxE Src		TxC Idl	TxC Src		

Pin Source Register

D31 Cable Transceiver Enable
Setting this bit turns on the cable transceivers. If this bit is cleared, the transceivers are tristated.

D30 RESERVED

D29 External Loopback Mode
When Cable Transceiver is enabled (Bit D31), this bit will automatically loopback the TxC/RxC, TxD/RxD, and TxE/RxE signals at the cable (transceivers enabled).
Notes:

- The DCE/DTE mode will select the set of signals (DCE or DTE) to be looped back

D28 DCE/DTE Mode
This bit sets up the transceiver direction. Setting the mode to '1' will enable DCE mode, while '0' will set DTE mode (default). See Section 5.3 for a detail of the signal direction as defined for each mode.

D27:24 Transceiver Protocol Mode

D27	D26	D25	D24	Transceiver Mode
0	0	0	0	RS-422 / RS-485
0	0	0	1	RS-423
0	0	1	0	RS-232
0	0	1	1	RESERVED
0	1	X	X	RESERVED
1	X	X	X	RESERVED

D23	Int LB	0	Normal Mode
		1	Internal Loopback – TxC, TxD, TxE looped back internally

D22:21 RESERVED

D20	TxC Idl Lev	0	TxC driven low ('0') while Idle (Envelope Negated)
		1	TxC driven high ('1') while Idle (Envelope Negated)

D19	TxD Idl Lev	0	TxD driven low ('0') while Idle (Envelope Negated)
		1	TxD driven high ('1') while Idle (Envelope Negated)

D18:17	RESERVED		
D16	RxD Src	0	RxD Active Hi (NRZ)
		1	RxD Active Lo (NRZB)
D15:14	RxE Src	00	RxE Active Hi
		01	RxE Active Lo
		1X	RxE Disabled
D13	RxC Src	0	Sample Data on Falling Edge of Clock (Data Change on Rising)
		1	Sample Data on Rising Edge of Clock (Data Change on Falling)
D12:11	TxSp Src	0X	Tri-State
		10	'0'
		11	'1'
D10:9	TxAuxC Src	00	Tri-State
		01	Prg_Clk / 2
		10	'0'
		11	'1'
D8:6	TxD Src	000	TxD Active Hi (NRZ)
		001	TxD Active Lo (NRZB)
		01X	RESERVED
		10X	RESERVED
		110	'0'
		111	'1'
D5:4	TxE Src	00	TxE Active Hi
		01	TxE Active Lo
		10	'0'
		11	'1'
D3	TxC Idl	0	TxC driven while Idle (Envelope Negated)
		1	No TxClk while Idle (Envelope Negated)
D2:0	TxC Src	000	Clock Data on Rising Edge of Internal Programmable Clock / 2 (Data/Envelope change on rising edge)
		001	Clock Data on Falling Edge of Internal Programmable Clock / 2 (Data/Envelope change on falling edge)
		010	Clock Data on Rising Edge of External Clock (Data/Envelope change on rising edge)
		011	Clock Data on Falling Edge of External Clock (Data/Envelope change on falling edge)
		10X	RESERVED
		110	'0'
		111	'1'

2.10 Channel Pin Status: Local Offset 0x0090 / 0x0094 / 0x0098 / 0x009C

In addition to standard inputs, unused inputs may be utilized as general purpose input signals. The Channel Pin Status Register allows the input state of all the IO pins to be monitored. Output signals as well as inputs are included to aid in debug operation. As the input signals are inputs from the cable, the transceivers must be enabled before the

D31:D10	RESERVED
D9	TxSp Output
D8	RxSp Input
D7	TxAuxC Output
D6	TxE Output
D5	TxD Output
D4	TxC Output
D3	RxAuxC Input
D2	RxE Input
D1	RxD Input
D0	RxC Input

2.11 Programmable Clock Registers: Local Offset 0x00A0 / 0x00A4 / 0x00A8

The Programmable Clock Registers allow the user to program the on-board programmable oscillator and configure the channel clock post-dividers. As GSC should provide software routines to program the clock, the user should have no need to access these registers. See Section 4.6 for more information.

2.12 Tx Count Register: Local Offset 0x00B0 / 0x00B4 / 0x00B8 / 0xBC

D31:16	Gap Bit Count When transmitting, these bits indicate the number of idle clocks between transmitted words. To output a continuous stream of bits, this value should be set to zero.
D15:0	Transmit Bit Count These bits indicate the number of consecutive bits to transmit for each transmit word.

2.13 Rx Count Register: Local Offset 0x00C0 / 0x00C4 / 0x00C8 / 0xCC

D31:16	RESERVED
D15:0	Receive Bit Count When receiving, these bits indicate the number of consecutive bits received for the last received word.

2.14 FIFO Count Register: Local Offset 0x00D0 / 0x00D4 / 0x00D8 / 0x00DC

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

D31:D16	Number of words in Rx FIFO
D15:D0	Number of words in Tx FIFO

2.15 FIFO Size Register: Local Offset 0x00E0 / 0x00E4 / 0x00E8 / 0x00EC

The FIFO Size Registers display the sizes of the installed data FIFOs. This value is calculated at power-up 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:D16	Size of installed Rx FIFO
D15:D0	Size of installed Tx FIFO

2.16 Features Register: Local Offset 0x00FC

The Features Register allows software to account for added features in the firmware versions. Bits will be assigned as new features are added.

D31:16	RESERVED
D15:8	Features Rev Level
D7:4	0x7 - FIFO Counters/Size and Board Reset Feature implemented
D3:0	0x3 - Programmable Clock Configuration = CY22393

CHAPTER 3: PCI INTERFACE

3 PCI Interface Registers

A PCI9080 I/O Accelerator from PLX Technology handles the PCI Interface. The PCI interface is compliant with the 5V, 33MHz 32-bit PCI Specification 2.1. 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 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 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 PCI-X 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.

The PLX 9080 contains many registers, many of which have no effect on the SIO4BX performance. The following section attempts to filter the information from the PCI9080 manual to provide the necessary information for a SIO4BX specific driver.

The SIO4BX 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.

3.1 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 SIO4BX 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	0x2401	GSC SIO4

The configuration registers also setup the PCI IO and Memory mapping for the SIO4BX. 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.

3.2 Local Configuration Registers

The Local Configuration registers give information on the Local side implementation. These include the required memory size. The SIO4 memory size is initialized to 4k Bytes. All other Local Registers initialize to the default values described in the PCI9080 Manual.

3.3 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 SIO4BX. All other Runtime Registers initialize to the default values described in the PCI9080 Manual.

3.4 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 SIO4BX supports both Demand (DREQ# controlled) and Non-Demand mode DMA. Both Channel 0 and Channel 1 DMA are supported.

3.4.1 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 SIO4BX. (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 4: PROGRAMMING

4.1 Serial Interface

4.1.1 Serial Interface Definition

The Pin Source Register contains information which defines the physical serial interface. This register contains fields to setup the polarity of the TxC, RxC, TxD, RxD, TxE, and RxE signals. As these signals are all individually configurable, it is possible to setup the Receive channel differently than the Transmit channel. In addition, the TxD Idle field defines the state of the TxD signal while idling (not sending data).

The MSB/LSB for both transmit and receive is setup in the Channel Control/Status register. Note that this acts only upon the current byte. MSB will send/receive bit D7 first, LSB will send/receive D0 first. For word lengths of other than 8 bits, the word should be right/left justified accordingly. Since the MSB/LSB only acts upon 8 bits, if the bit length is greater than 8 bits, the bits will only be reversed on a byte by byte basis. The user may need to rearrange bytes for bit lengths greater than 8 bits.

4.1.2 Two Signal Interface

A two signal interface is a special setup case of the three signal serial interface. In a three signal interface, an envelope signal defines when data is valid. For a two wire case, every clock indicates valid data.   When data is invalid, the clock simply stops.

For transmit, the TxC Idle field defines the two signal interface. By setting the TxC Idle enable, the clock will stop during idle periods. The TxCount and TxGap still apply. The TxE signal may be left enabled and simply unconnected, or may be reconfigured as a general purpose output.

For receive, a two wire interface is defined when RxE is set as a general purpose input. When RxE is set as an input, the internal logic simply assumes the internal RxE is always valid. Thus, all data is considered valid based on the RxC clock.

4.1.3 Tx Bit Count / Tx Gap

The TxCount/TxGap register defines the number of consecutive bits to transmit in a word, as well as the number of idle clock cycles between words. This configurability allows this board to interface with a custom user interface. All data sent to/from the board is in 8 bit increments. Therefore, if TxCount is not a multiple of 8, all extra bits will be padded with zeros. .For example, a TxCount of 14 would use 14 bits out of two consecutive bytes (and the two extra bits would be ignored).

Note that there is no hardware interlock to ensure that TxCount bits are present in the Tx FIFO before a transmit can begin. If TxCount is greater than 8 bits (and transmit is enabled), the first 8 bits will be transmitted as soon as it is loaded into the TxFIFO. If the TxFIFO is empty when the first 8 bits complete, a gap will be inserted. Therefore, if the TxCount is greater than 8 bits, some data should be preloaded into the TxFIFO before the transmit is enabled. This will ensure a continuous data stream of the correct length.

For a continuous data stream, the TxCount should be set to 8 and TxGap to 0.

Certain TxCount/TxGap combinations may not work correctly in a very few instances. In general, a data word cannot be transmitted or received faster than 500ns per byte. If the TxCount, TxGap, and serial data rate result in a throughput rate of greater than 1 byte in 500ns, correct operation cannot be guaranteed.   If an application requires such an interface, please contact GSC tech support to determine if the board will work for your application.

4.1.4 Rx Bit Count

The Rx Bit Count is primarily a debug feature to check that the expected number of bits in a frame were received. The Rx Bit Count will simply count received bits in the current frame. It will reset at the beginning of each frame based on RxE, or may be reset via the Rx Bit Count Reset bit of the Channel Control Register. For a two signal interface, this register will count all bits reveived.

4.1.5 DCE/DTE Mode

Since the main signal interface signals (Clock, Data, Envelope) are bidirectional signals, DTE/DCE mode is used to control the direction of the cable interface for each signal set. The DTE/DCE mode bit in the Pin Source Register is used to set the interface direction. Setting the channel mode to DTE or DCE will swap the location of the interface signals on the connector (See Section 5.3 **Interface Connector** for cable pinout). By allowing the signal direction (DTE or DCE) to be set via software control, the user can send and receive on a single set of bidirectional lines. If bidirectional signals are not required, the mode should be set to DTE (default) and connected appropriately.

Note that the pin direction for the TxC/RxC, TxD/RxD, and TxE/RxE are always set by the DTE/DCE mode. If these signals are used as GPIO (such as TxE and RxE in 2 wire mode), the IO direction is still set by the DTE/DCE Mode – they are not individually configurable.

4.1.6 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.

In addition, 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, TXD), 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 the signals are driven at the transceivers, any external connections could interfere with loopback operation. Therefore, the cable 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.

4.2 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.

4.2.1 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.

4.2.2 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.

4.2.3 FIFO Size

In some applications, 512byte 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 512 bytes at a time (and an effective throughput rate less than 500kpbs). For these applications, a PMC-SIO4BX-SYNC-4KLC 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.

4.2.4 Internal vs. External FIFOs

In this manual, references may be made to internal and external FIFOs. There is really no difference between the two as far as software is concerned. Internal FIFOs are simply small (512 byte) FIFOs which can be implemented internal to the on-board FPGA to provide a lower cost board. From the user standpoint, there is no difference between internal and external FIFOs. However, different firmware may be required. Therefore, certain status bits may indicate internal or external FIFOs as a debug aid (if tech support issues arise).

4.3 Board vs. Channel Registers

Since four serial channels are implemented on a single board, some registers apply to the entire board, while others are unique to each channel. It is intended that each channel can act independently, but the user must keep in mind that certain accesses will affect the entire board. Typically, the driver will adequately handle keeping board and channel interfaces separate. However, the user must also be mindful that direct access to certain registers will affect the entire board, not just a specific channel.

The Board Control and Board Status registers provide board level controls. Fundamentally, a board reset will do just that, reset all the GSC registers and FIFOs to their default state. Interrupt control is also shared among all registers, although local bits are segregated by channel. The device driver should take care of appropriately handling the inter-mixed channel interrupts and pass them on to the application appropriately.

4.4 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.

Since TxAuxC and RxAuxC share a single pin, the TxAuxC Src field in the Pin Source Register controls whether AuxC will function as an input or output. If the field is set to ‘Tri-State’, the pin is set as RxAuxC Input. Otherwise, The TxAuxC output will be driven (and the RxAuxC input will be equal to the TxAuxC output). Likewise, the TxSp Src field in the Pin Source Register controls the Spare pin direction. If the field is set to ‘Tri-State’, RxSp will be an Input. Otherwise, RxSp will follow the TxSp output.

4.5 Interrupts

The PMC-SIO4BX-SYNC has a number of interrupt sources which are passed to the host CPU via the PCI IRQA. Since there is only one physical interrupt source for the board, 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 on-board interrupt. . The single Local Interrupt is made up of the interrupt sources described in Section 2.8. The user should be aware that interrupts must be enabled at each level for an interrupt to occur. For example, if a FIFO interrupt is used, it must be setup and enabled in the GSC Firmware Interrupt Control Register, as well as enabled in the PCI9080. In addition, the interrupt must be acknowledged and/or cleared at each level following the interrupt. The driver will typically take care of setting up and handling the PCI9080 interrupts as well as most local interrupts. The specific driver manual should have more information on how to handle these interrupts.

4.6 Programmable Oscillator / Programmable Clocks

The On-Board Programmable Oscillator provides each channel with a unique programmable clock source using a Cypress Semiconductor CY22393 Programmable Clock generator. In order to program the oscillator, it is necessary to calculate and program values for different clock frequencies. General Standards has developed routines to calculate the necessary values for a given setup and program the clock generator. These clock setup routines have been incorporated into most of the drivers.

The default clock configuration at power-up for the programmable clock on all channels is 20MHz. See Appendix A for more detailed information concerning programming the on-board clock frequencies, as well as common frequency setups. The specific driver manual should have information on clock setup. If not, please contact GSC tech support for assistance.

4.7 PCI DMA

The PCI DMA functionality allows data to be transferred between host memory and the PCI-SIO4B 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.

There are two PCI DMA modes – Non-Demand Mode DMA and Demand Mode DMA. In Non-Demand mode, all data is transferred without data checks, so the driver must ensure no data will be lost. This adds a little software overhead, but no data should be lost. For Demand mode, a hardware handshake is implemented such that the DMA is throttled as the Tx FIFO becomes full or the Rx FIFO is empty. However, Demand Mode has some potential drawbacks which could result in lost data. Given these tradeoffs, Non-Demand Mode DMA is the preferred method for DMA transfers.

Non-Demand Mode DMA is fast, but it requires the driver to ensure there is enough data (receive) or space in the FIFO (transmit) to ensure data will not be lost. Fortunately, the FIFO counters provide a real-time count of the number of data bytes currently in the FIFO, so the driver should be able to break large transfers into smaller transfers to ensure data is not lost. Although this adds a small bit of overhead, it can ensure that no data is lost.

With Demand Mode DMA, a large DMA transfer (greater than the size of the FIFO) can be setup, which requires less software overhead. However, there are two potential drawbacks to this method. Firstly, if the transfer does not complete, there is no way to determine how much data was actually transferred. This is primarily a problem for receiving – the expected amount of data was never received. The workaround for this is to DMA the read data from the RxFIFO in small blocks as it is received (the same as Non-Demand mode). The second problem is that the PCI interface chip always wants to transfer 32bits at a time. Since the serial data is only 8bits, the Demand mode DMA always wants to transfer 4 bytes (even if only 1 byte is available). This can obviously lead to extraneous data (transmit) or data loss (receive). There are a couple ways to deal with this problem. The driver can convert all the FIFO accesses to 32bits, converting every byte to 32bits. This can obviously be very inefficient. Another solution involves a hardware/software workaround where almost a complete block is transferred via DMA, but then driver reads/writes any leftover bytes to cleanup the transfer.

CHAPTER 5: HARDWARE CONFIGURATION

5 Board Layout

The following figure is a drawing of the physical components of the PMC-SIO4BX-SYNC:

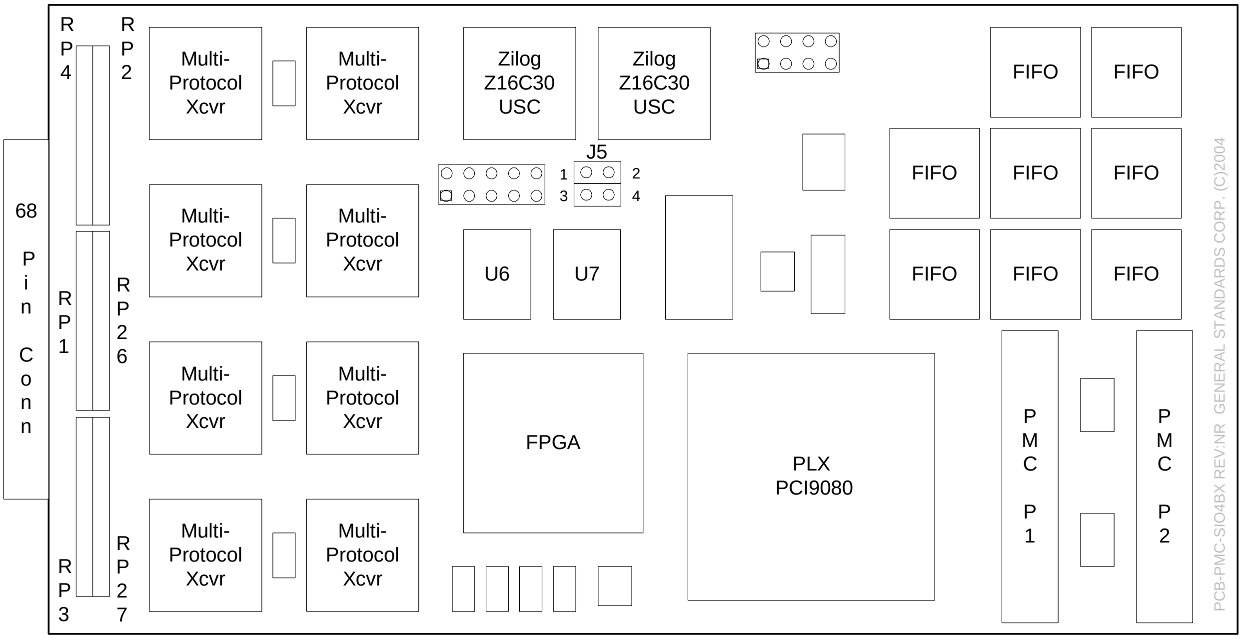


Figure 5-1: Board Layout

5.1 Board ID Jumper J5

Jumper J5 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 SIO4BX card is in a system. When the Board ID jumper is installed, it will read ‘0’ in the Board Status Register. The Board Status Register bit will report ‘1’ when the jumper is removed. Refer to Figure 5.1-1 for Jumper J5 location.

J5 Jumper	Description	Notes
1 - 2	Board ID 0	Defines Board ID 0 In Board Status Register
3 - 4	Board ID 1	Defines Board ID 1 In Board Status Register

5.2 RS485/RS422 Termination Resistors

The RS485/RS422 cable interface requires termination of the differential signals to match the cable impedance. There are six on-board termination resistors to provide a parallel termination of 150 Ohms for the TxE/RxE, TxC/RxC, and TxD/RxD signals. Note that the TxSp/RxSp and TxAuxC/RxAuxC GPIO signals are not terminated. If a different value of termination resistor is required, the termination resistors are socketed so they can be changed or removed as necessary. There are six termination resistors – RP1 to RP4, RP26, and RP27. The termination resistors are standard 8-pin isolated resistor SIPs (four resistors per SIP). Refer to Figure 5.1-1 for resistor pack locations.

5.3 Interface Connector

The user interface connector for the PMC-SIO4BX-SYNC is a SCSI II type 68-pin connector (female) mounted to the front edge of the board (P2). The part number for this 68 pin SCSI II connector is AMP 787170-7. The mating cable connector is AMP 749621-7 (or AMP 749111-6) or equivalent. The tables below show the pinout for the RS485/RS422. Single Ended signals for RS-423 and RS-232 use the negative (-) side of the differential pair.

Pin #	DTE Signal	DCE Signal	Pin #	DTE Signal	DCE Signal
1	Ch1 AuxC +		35	Ch3 AuxC +	
2	Ch1 AuxC -		36	Ch3 AuxC -	
3	Ch1 Spare +		37	Ch3 Spare +	
4	Ch1 Spare -		38	Ch3 Spare -	
5	Ch1 RxE +	Ch1 TxE +	39	Ch3 RxE +	Ch3 TxE +
6	Ch1 RxE -	Ch1 TxE -	40	Ch3 RxE -	Ch3 TxE -
7	Ch1 RxD +	Ch1 TxD +	41	Ch3 RxD +	Ch3 TxD +
8	Ch1 RxD -	Ch1 TxD -	42	Ch3 RxD -	Ch3 TxD -
9	Ch1 RxC +	Ch1 TxC +	43	Ch3 RxC +	Ch3 TxC +
10	Ch1 RxC -	Ch1 TxC -	44	Ch3 RxC -	Ch3 TxC -
11	Ch1 TxE +	Ch1 RxE +	45	Ch3 TxE +	Ch3 RxE +
12	Ch1 TxE -	Ch1 RxE -	46	Ch3 TxE -	Ch3 RxE -
13	Ch1 TxD +	Ch1 RxD +	47	Ch3 TxD +	Ch3 RxD +
14	Ch1 TxD -	Ch1 RxD -	48	Ch3 TxD -	Ch3 RxD -
15	Ch1 TxC +	Ch1 RxC +	49	Ch3 TxC +	Ch3 RxC +
16	Ch1 TxC -	Ch1 RxC -	50	Ch3 TxC -	Ch3 RxC -
17	GND	GND	51	GND	GND
18	GND	GND	52	GND	GND
19	Ch2 RxE +	Ch2 TxE +	53	Ch4 RxE +	Ch4 TxE +
20	Ch2 RxE -	Ch2 TxE -	54	Ch4 RxE -	Ch4 TxE -
21	Ch2 RxD +	Ch2 TxD +	55	Ch4 RxD +	Ch4 TxD +
22	Ch2 RxD -	Ch2 TxD -	56	Ch4 RxD -	Ch4 TxD -
23	Ch2 RxC +	Ch2 TxC +	57	Ch4 RxC +	Ch4 TxC +
24	Ch2 RxC -	Ch2 TxC -	58	Ch4 RxC -	Ch4 TxC -
25	Ch2 TxE +	Ch2 RxE +	59	Ch4 TxE +	Ch4 RxE +
26	Ch2 TxE -	Ch2 RxE -	60	Ch4 TxE -	Ch4 RxE -
27	Ch2 TxD +	Ch2 RxD +	61	Ch4 TxD +	Ch4 RxD +
28	Ch2 TxD -	Ch2 RxD -	62	Ch4 TxD -	Ch4 RxD -
29	Ch2 TxC +	Ch2 RxC +	63	Ch4 TxC +	Ch4 RxC +
30	Ch2 TxC -	Ch2 RxC -	64	Ch4 TxC -	Ch4 RxC -
31	Ch2 Spare +		65	Ch4 Spare +	
32	Ch2 Spare -		66	Ch4 Spare -	
33	Ch2 AuxC +		67	Ch4 AuxC +	
34	Ch2 AuxC -		68	Ch4 AuxC -	

Table 5-1: 68 Pin Connector Pin-Out

See Section 4.1.5 for a description of DTE and DCE mode.

CHAPTER 6: ORDERING OPTIONS

6 Ordering Information

Since the PMC-SIO4BX-SYNC is designed to fit a variety of high-speed serial interface needs, there are several options that must be specified when ordering. Please consult our sales department with your application requirements to determine the correct ordering option. (sales@generalstandards.com).

6.1 Board Ordering Option – FIFO Size

The PMC-SIO4BX-SYNC can be installed with FIFOs ranging in depth from 512 bytes to 32k bytes. Larger FIFO depth is important for faster interfaces to reduce the risk of data loss due to software overhead. The following FIFO depth options are availabla:

Part Number	FIFO Depth
PMC-SIO4BX-SYNC-4KLC	512 bytes
PMC-SIO4BX-SYNC-64K	8k bytes
PMC-SIO4BX-SYNC 256K	32k bytes

Note that the FIFO size option in the board part number refers to the total FIFO size for all 8 channels, not the FIFO size of a single FIFO. For example, PMC-SIO4BX-SYNC-256K would contain eight 32k deep FIFOs. Please consult our sales department for pricing and availability.

6.2 Interface Cable

General Standards Corporation can provide off-the-shelf or custom interface cables for the PMC-SIO4BX-SYNC board. The standard cable is a non-shielded, twisted pair 68-conductor ribbon cable for increased noise immunity. Several standard cable lengths are offered, or the cable length can be custom ordered. 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. A standard cable is also available which will breakout the serial channels into four DB25 connectors. Shielded cable options are also available. Please consult our sales department for more information on cabling options and pricing.

6.3 Device Drivers

General Standards has developed many device drivers for the PMC-SIO4BX boards, including VxWorks, Windows, Linux, and LabView. As new drivers are always being added, please consult our website (www.generalstandards.com) or consult our sales department for a complete list of available drivers and pricing.

6.4 Custom Applications

Although the PMC-SIO4BX-SYNC board provides extensive flexibility to accommodate most user interfaces, some applications may require modifications to conform to a specialized user interface. General Standards Corporation has worked with many customers to provide customized versions based on the SIO4BX boards. Please consult our sales department with your specifications to inquire about a custom application.

APPENDIX A: PROGRAMMABLE OSCILLATOR PROGRAMMING

The four on-baord clock frequencies (one per channel) are supplied via a Cypress Semiconductor CY22393 Programmable Clock Generator.   This chip must be reprogrammed in order to change the clock frequencies. This document supplies the information necessary to reprogram the on-board clock frequencies.

The serial drivers supplied by GSC should include routines to calculate and program the on-board oscillator for a given set of frequencies. Therefore, it should not be necessary for the user need the following reprogramming information. It is provided for documentation purposes. Please contact GSC for help in setting up the on-board oscillator.

The CY22393 contains several internal address which contain the programming information. GSC has mirrored this data internal to the FPGA to allow the user to simply setup the data in the FPGA RAM and then command the on-board logic to program the clock chip. This isolates the user from the hardware serial interface to the chip. For detailed CY22393 programming details, please refer to the Cypress Semiconductor CY22393 data sheet.

The GSC CLOCK RAM (internal to FPGA) is accessed through 2 registers at local offsets 0x00A0 (Address Reg) and 0x00A4 (Data Reg). The user simply sets the RAM Address register to the appropriate offset, then reads or writes the the RAM data. The Programmable Osc Control/Status register allows the user to program the CY22393 or setup the clock post-dividers.

The GSC Local Programmable Clock Registers are defined as follows:

0x00A0 – RAM Address Register

Defines the internal CLOCK RAM address to read/write

0x00A4 – RAM Data Register

Provides access to the CLOCK RAM pointed to by the RAM Addr Register.

0x00A8 – Programmable Osc Control/Status Register

Provides control to write the contents of the CLOCK RAM to the CY22393 and setup additional post-dividers for the input clocks.

Control Word (Write Only)

D0	Program Oscillator 1 = Program contents of CLOCK RAM to CY22393. Automatically resets to 0.
D1	Measure Channel 1 Clock
D2	Measure Channel 2 Clock
D3	Measure Channel 3 Clock
D4	Measure Channel 4 Clock
D5	Reserved (Unused)
D6	Status Word Readback Control 0 => Status Word D31-D8 == Measured Channel Value 1 => Status Word D31-D8 == Control Word D23-D0
D7	Post-divider set 0 = Ignore D23-D8 during Command Word Write 1 = Set Channel Post-Dividers from D23-D8 during Command Word Write
D11-D8	Channel 1 Post-Divider
D15-D12	Channel 2 Post-Divider
D19-D16	Channel 3 Post-Divider
D23-D20	Channel 4 Post-Divider
D31-D24	Reserved (Unused)

Status Word (Read Only)

D0	Program Oscillator Done 0 = Oscillator Programming in progress.
D1	Program Oscillator Error 1 = Oscillator Programming Error has occurred.
D2	Clock Measurement complete. 0 = Clock Measurement in progress.
D7-D3	Reserved (Unused)
D31-D8	If Command Word D6 = 0, Measured Channel Clock Value If Command Word D6 = 1, Control Word D23-D0

Channel Clock Post-Dividers:

The Control Word defines 4 fields for Channel Clock Post-dividers, one field for each channel (D8-D23). These post-dividers divide down the clocks from the programmable oscillator (CY22393 outputs) to provide for slower baud rates. These divided-down clocks are used as the final programmable clock values. Each 4 bit field will allow a post divider of 2^n. For example, if the post-divider value=0, the input clock is not post-divided. A value of 2 will provide a post-divide of 4 (2^2). This will allow for a post-divide value of up to 32768 (2^15) for each input clock.

Bit D7 of the Control Word qualifies writes to the post-divide registers. This allows other bits in the command register to be set while the post-divide values are maintained.

A value of ‘0’ in the post divider field will bypass the post-divider (final programmamle clock = CY22393 output).

Channel Clock Measurement:

The Control Word defines 4 bits which will select one of the 4 channel clocks (input clock + post-divide) for a measurement. This will allow the user feedback as to whether the programmable oscillator was programmed correctly. To measure a clock, select the clock to measure in the Control word, and also clear Bit D6 to allow for readback of the result. Read back the Status Word until D2 is set. Status Word D31-D8 should contain a value representing 1/10 the measured clock frequency (Value * 10 = Measured Frequency in MHz). Keep in mind that this value will not be exactly the programmed frequency due to the 100ppm (0.01%) accuracy of the on-board reference.

The Internal FPGA RAM is defined as follows: RAM Address 0x08–0x57 correspond directly to the CY22393 registers. Do not change ‘Reserved’ values from their defaults.

Address	Description	Default Value
0x00 – 0x05	Reserved (Unused)	0x00
0x06	Reserved	0xD2
0x07	Reserved	0x08
0x08	ClkA Divisor (Setup0)	0x01
0x09	ClkA Divisor (Setup1)	0x01
0x0A	ClkB Divisor (Setup0)	0x01
0x0B	ClkB Divisor (Setup1)	0x01
0x0C	ClkC Divisor	0x01
0x0D	ClkD Divisor	0x01
0x0E	Source Select	0x00
0x0F	Bank Select	0x50
0x10	Drive Setting	0x55
0x11	PLL2 Q	0x00
0x12	PLL2 P Lo	0x00
0x13	PLL2 Enable/PLL2 P Hi	0x00
0x14	PLL3 Q	0x00
0x15	PLL3 P Lo	0x00
0x16	PLL3 Enable/PLL3 P Hi	0x00
0x17	OSC Setting	0x00
0x18	Reserved	0x00
0x19	Reserved	0x00
0x1A	Reserved	0xE9
0x1B	Reserved	0x08
0x1C-0x3F	Reserved (Unused)	0x00
0x40	PLL1 Q (Setup0)	0x00
0x41	PLL1 P Lo 0 (Setup0)	0x00
0x42	PLL1 Enable/PLL1 P Hi (Setup0)	0x00
0x43	PLL1 Q (Setup1)	0x00
0x44	PLL1 P Lo 0 (Setup1)	0x00
0x45	PLL1 Enable/PLL1 P Hi (Setup1)	0x00
0x46	PLL1 Q (Setup2)	0x00
0x47	PLL1 P Lo 0 (Setup2)	0x00
0x48	PLL1 Enable/PLL1 P Hi (Setup2)	0x00
0x49	PLL1 Q (Setup3)	0x00
0x4A	PLL1 P Lo 0 (Setup3)	0x00
0x4B	PLL1 Enable/PLL1 P Hi (Setup3)	0x00
0x4C	PLL1 Q (Setup4)	0x00
0x4D	PLL1 P Lo 0 (Setup4)	0x00
0x4E	PLL1 Enable/PLL1 P Hi (Setup4)	0x00
0x4F	PLL1 Q (Setup5)	0x00
0x50	PLL1 P Lo 0 (Setup5)	0x00
0x51	PLL1 Enable/PLL1 P Hi (Setup5)	0x00
0x52	PLL1 Q (Setup6)	0x00
0x53	PLL1 P Lo 0 (Setup6)	0x00
0x54	PLL1 Enable/PLL1 P Hi (Setup6)	0x00
0x55	PLL1 Q (Setup7)	0x00
0x56	PLL1 P Lo 0 (Setup7)	0x00
0x57	PLL1 Enable/PLL1 P Hi (Setup7)	0x00
0x58-0xFF	Reserved (Unused)	0x00