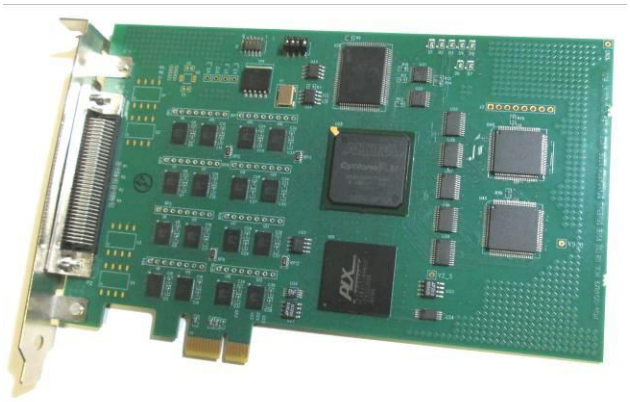
PCIe-SIO4BX2

Hardware User's Manual



Four Channel PCIe High Performance Serial I/O

Featuring RS422/RS485/RS232 Software Configurable Transceivers and 32K Byte FIFO Buffers (256K Byte total)

RS485
RS422/V.11
RS232/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

Rev 2

PREFACE

Revision History

1. Rev NR – Mar 2013 – Original rev from PMC66-SIO4BXR manual.
2. Rev 1 – May 2013 – Update LEDs
3. Rev 2 – Nov 2024 – Update D6 LED

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

ZILOG Z16C30 USC® User's Manual
ZILOG Z16C30 USC® Product Specifications Databook

ZILOG, Inc.
210 East Hacienda Ave.
Campbell, CA 95008-6600
(408) 370-8000
http://www.zilog.com/

PLX PEX8311AA 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-422A)

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

PREFACE ..... **I**

RELATED PUBLICATIONS ..... **II**

TABLE OF CONTENTS ..... **III**

CHAPTER 1: INTRODUCTION **1**

1.0 General Description 1
1.1 Z16C30 Universal Serial Controller 2
1.2 Deep Transmit/Receive FIFOs 2
1.3 Multiprotocol Transceivers 3
1.4 PMC/PCI Interface 3
1.5 General Purpose IO 3
1.6 Connector Interface 3
1.7 New Features 3

CHAPTER 2: LOCAL SPACE REGISTERS **4**

2.0 Register Map 4
2.1 GSC Firmware Registers 4
2.1.1 Firmware Revision: Local Offset 0x0000 5
2.1.2 Board Control: Local Offset 0x0004 6
2.1.3 Board Status: Local Offset 0x0008 7
2.1.4 Timestamp: Local Offset 0x000C 7
2.1.5 Channel TX Almost Flags: Local Offset 0x0010 / 0x0020 / 0x0030 / 0x0040 7
2.1.6 Channel RX Almost Flags: Local Offset 0x0014 / 0x0024 / 0x0034 / 0x0044 8
2.1.7 Channel FIFO: Local Offset 0x0018 / 0x0028 / 0x0038 / 0x0048 8
2.1.8 Channel Control/Status: Local Offset 0x001C / 0x002C / 0x003C / 0x004C 8
2.1.9 Channel Sync Detect Byte: Local Offset 0x0050 / 0x0054 / 0x0058 / 0x005C 9
2.1.10 Interrupt Registers 9
2.1.11 Interrupt Control: Local Offset 0x0060 10
2.1.12 Interrupt Status/Clear: Local Offset 0x0064 10
2.1.13 Interrupt Edge/Level: Local Offset 0x0068 11
2.1.14 Interrupt Hi/Lo: Local Offset 0x006C 11
2.1.15 Channel Pin Source: Local Offset 0x0080 / 0x0084 / 0x0088 / 0x008C 11
2.1.16 Channel Pin Status: Local Offset 0x0090 / 0x0094 / 0x0098 / 0x009C 14
2.1.13 Programmable Clock Registers: Local Offset 0x00A0 / 0x00A4 / 0x00A8 / 0xAC 15
2.1.14 FIFO Count Register: Local Offset 0x00D0 / 0x00D4 / 0x00D8 / 0x00DC 15
2.1.15 FIFO Size Register: Local Offset 0x00E0 / 0x00E4 / 0x00E8 / 0x00EC 15
2.1.16 FW Type ID Register: Local Offset 0x00F8 15
2.1.17 Features Register: Local Offset 0x00FC (0x00197AF4) 16
2.2 Universal Serial Controller Registers 16
2.2.1 USC Reset 16
2.2.2 8-Bit USC Register Access 17
2.2.3 USC Data Transfer 17
2.2.4 USC Register Memory Map 18

CHAPTER 3: PROGRAMMING 19

3.0 INTRODUCTION 19
3.1 RESETS 19
3.2 FIFOS 19
3.2.1 FIFO FLAGS 19
3.2.2 FIFO COUNTERS 20
3.2.3 FIFO SIZE 20
3.3 BOARD VS. CHANNEL REGISTERS 20
3.4 PROGRAMMABLE OSCILLATOR / PROGRAMMABLE CLOCKS 21
3.5 CLOCK SETUP 21
3.6 MULTIPROTOCOL TRANSCEIVER CONTROL 23
3.7 DCE/DTE MODE 23
3.8 LOOPBACK MODES 23
3.9 GENERAL PURPOSE IO 24
3.10 INTERRUPTS 24
3.11 PCI DMA 24

CHAPTER 4: PCI INTERFACE 25

4.0 PCI INTERFACE REGISTERS 25
4.1 PCI REGISTERS 25
4.1.1 PCI CONFIGURATION REGISTERS 25
4.1.2 LOCAL CONFIGURATION REGISTERS 26
4.1.3 RUNTIME REGISTERS 26
4.1.4 DMA REGISTERS 26
4.1.4.1 DMA CHANNEL MODE REGISTER: (PCI 0X80 / 0X94) 26

CHAPTER 5: HARDWARE CONFIGURATION 27

5.0 BOARD LAYOUT 27
5.1 BOARD ID JUMPER J2 27
5.2 TERMINATION RESISTORS 28
5.3 LEDS 28
5.4 INTERFACE CONNECTOR 29

CHAPTER 6: ORDERING OPTIONS 30

6.0 ORDERING INFORMATION 30
6.1 INTERFACE CABLE 30
6.2 DEVICE DRIVERS 30
6.3 CUSTOM APPLICATIONS 30

APPENDIX A: PROGRAMMABLE OSCILLATOR PROGRAMMING 31

APPENDIX B: FIRMWARE REVISIONS / FEATURES REGISTER 34

CHAPTER 1: INTRODUCTION

1.0 General Description

The PCIe-SIO4BX2 is a four channel serial interface card which provides high speed, full-duplex, multi-protocol serial capability for PCIe applications. The PCIe-SIO4BX2 combines multi-protocol Dual Universal Serial Controllers, deep external FIFOs, and software selectable multi-protocol transceivers to provide four fully independent synchronous/asynchronous serial channels. These features, along with a high performance one lane PCIe interface engine, give the PCIe-SIO4BX2 unsurpassed performance in a serial interface card.

Features:

- One Lane PCI Express (PCIe) Interface
- Four Independent RS422/RS485/RS232 Serial Channels
- Serial Mode Protocols - Asynchronous, Monosync, Bisync, SDLC, HDLC, Nine-Bit, IEEE 802.3
- Synchronous Serial Data Rates up to 10Mbps
- Asynchronous Serial Data Rates up to 1Mbps
- Independent Transmit and Receive FIFOs for each Serial Channel – 32K byte each
- Multi-protocol Transceivers support RS422/RS485 and RS232
- Parity and CRC detection capability
- Programmable Oscillators provide increased flexibility for Baud Rate Clock generation
- SCSI type 68 pin front edge I/O Connector
- Eight signals per channel, configurable as either DTE or DCE:
  3 Serial Clocks (TxC, RxC, AuxC), 2 Serial Data signals (TxD, RxD), CTS, RTS, DCD
- Unused signals may be reconfigured as General Purpose IO
- Fast RS422/RS485 Differential Cable Transceivers Provide Data Rates up to 10Mbps
- RS232 Cable Transceivers Provide Data Rates up to 250kbps
- Industry Standard Zilog Z16C30 Multi-Protocol Universal Serial Controllers (USC®)
- Standard Cable to four DB25 connectors and Custom Cables available
- Available drivers include VxWorks, Win10, Linux, and Labview
- Industrial Temperature Option Available

Functional Diagram:

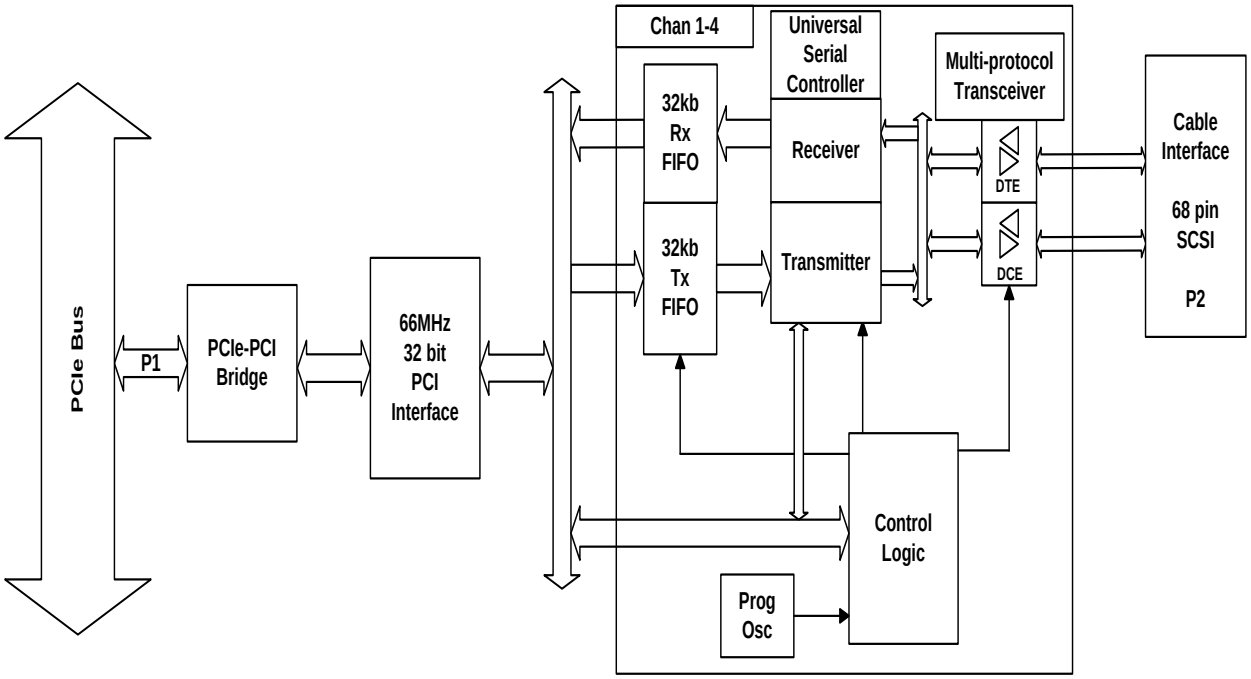


Figure 1-1 Block Diagram of PCIe-SIO4BX2

1.1 Z16C30 Universal Serial Controller

The PCIe-SIO4BX2 is designed around the Z16C30 Universal Serial Controller (USC). The Z16C30 is a dual channel multi-protocol serial controller which may be software configured to satisfy a wide variety of serial communications applications. The USC supports most common asynchronous and synchronous serial protocols. The USC provides many advanced features, including:

- Completely independent transmitter and receiver operation
- Odd/Even/Space/Mark parity
- Two 16-bit or one 32-bit CRC polynomial
- Eight Data Encoding methods – NRZ, NRZB, NRZI-Mark, NRZI-Space, Biphase-Mark, Biphase-Space, Biphase-Level, and Differential Biphase-Level

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 are always 32k bytes deep. 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 SIO4BX2 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 SIO4BX2 data is transferred over the user interface using high-speed multiprotocol transceivers. These multiprotocol transceivers are software selectable as RS422/RS485, 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 SIO4BX2 is through the PMC/PCI interface. An industry standard PCI9056 bridge chip from PLX Technology is used to implement PCI Specification 2.2. The PCI9056 provides the 32bit, 66MHz (264MBit/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 SIO4BX2 provides the flexibility to remap unused signals to be used as general purpose IO. For example, this would allow support for an application requiring DTR/DSR signals to be implemented on an unused DCD or TxAuxC signals. This also allows signals from unused channels to be available as general purpose IO.

1.6 Connector Interface

The SIO4BX2 provides a user IO interface through a front-side card edge connector. All four serial channels interface through this high-density, 68 pin SCSI-3 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 SCSI-3 connector into four DB25 connectors (one per channel). A standard cable is also available with a single 68 pin SCSI-3 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.

1.7 New Features

The PCIe-SIO4BX2 has been enhanced with several new features. These include improved receive data status recording, timestamping of data, flexible FIFO memory allocation, sync/standard channel select, and channel reset.

CHAPTER 2: LOCAL SPACE REGISTERS

2.0 Register Map

The SIO4BX2 is accessed through three sets of registers – PCI Registers, USC Registers, and GSC Firmware Registers. The GSC Firmware Registers and USC Registers are referred to as Local Space Registers and are described below. The PCI registers are discussed in Chapter 3.

The Local Space Registers are divided into two distinct functional register blocks – the GSC Firmware Registers and the USC Registers. The GSC Firmware Registers perform the custom board control functions, while the USC Registers map the Zilog Z16C30 registers into local address space. The register block for each USC channel is accessed at a unique address range. The table below shows the address mapping for the local space registers.

Local Address Range	Base Address Offset	Register Block Description
0x0000 – 0x00FF	0x0000	GSC Firmware Registers
0x0100 – 0x013F	0x0100	Channel 1 USC Registers
0x0140 – 0x01FF		Reserved
0x0200 – 0x023F	0x0200	Channel 2 USC Registers
0x0240 – 0x02FF		Reserved
0x0300 – 0x033F	0x0300	Channel 3 USC Registers
0x0340 – 0x03FF		Reserved
0x0400 – 0x043F	0x0400	Channel 4 USC Registers

The GSC Firmware Registers are detailed in Section 2.1. The USC Registers are briefly touched on in Section 2.2 of this manual, but are described in much greater detail in the Zilog Z16C30 Users Manuals.

2.1 GSC Firmware Registers

The GSC Firmware Registers provide the primary control/status for the SIO4BX2 board. The following table shows the GSC Firmware Registers.

Offset Address	Size	Access*	Register Name	Default Value (Hex)
0x0000	D32	Read/Write	Firmware Revision	E51001XX
0x0004	D32	Read/Write	Board Control	00000000
0x0008	D32	Read Only	Board Status	000000XX
0x000C	D32	Read/Write	Timestamp	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 1 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 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
0x0050	D32	Read/Write	Ch 1 Sync Byte	00000000
0x0054	D32	Read/Write	Ch 2 Sync Byte	00000000
0x0058	D32	Read/Write	Ch 3 Sync Byte	00000000
0x005C	D32	Read/Write	Ch 4 Sync Byte	00000000
0x0060	D32	Read/Write	Interrupt Control	00000000
0x0064	D32	Read/Write	Interrupt Status	00000000
0x0068	D32	Read Only	Interrupt Edge/Level	FFFF7777
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 1	00000000
0x00A8	D32	Read/Write	Programmable Osc Control/Status	00000000
0x00AC	D32	Read/Write	Programmable Osc RAM Data 2	00000000
0x00B0-0x00CC	---	--	RESERVED	--------
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
0x00F0-0x00F4	---	--	RESERVED	--------
0x00F8	D32	Read Only	FW Type Register	01010101
0x00FC	D32	Read Only	Features Register	00197AF4

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. See Appendix B for more detailed information.

D31:16	HW Board Rev	E511 = PCIe-SIO4BX2 Rev A
D15:8	Firmware Type ID	01 = SIO4B Standard
D7:0	Firmware Revision	Firmware Version

2.1.2 Board Control: Local Offset 0x0004

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

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, Reset-In-Progress bit (D31) of the Board Status Register will remain set until the Board reset is complete;

D30 RESERVED (Debug Test)

D29 FIFO Test (Debug Test)
0 = Normal Mode - FIFO Write to Tx FIFO / FIFO Read from Rx FIFO
1 = Test Mode - FIFO Write to Rx FIFO / FIFO Read from Tx FIFO

D28:27 FIFO Allocation (Unused)

D26 RESERVED

D25 LED D1
1 = Turn on Red LED D1

D24 LED D1
1 = Turn on Green LED D1

D23 Timestamp Clear
0 = timestamp counter is enabled
1 = reset timestamp count to zero

D22 Timestamp Source
0 = timestamp counter runs off internal 1us clock

D21:9 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
000 = Ch1 Rx
100 = Ch1 Tx
010 = Ch2 Rx
110 = Ch2 Tx
001 = Ch3 Rx
101 = Ch3 Tx
011 = Ch4 Rx
111 = Ch4 Tx

D3 Demand Mode DMA Channel 0 Single Cycle Disable

D2:0 Demand Mode DMA Channel 0 Request
000 = Ch1 Rx
100 = Ch1 Tx
010 = Ch2 Rx
110 = Ch2 Tx
001 = Ch3 Rx
101 = Ch3 Tx
011 = Ch4 Rx
111 = Ch4 Tx

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 SIO4 boards are present in a system.

D31:9	RESERVED
D8	0 = Standard 1 = Sync
D7:D6	RESERVED
D5:D4	FIFO Size 10 = 256K
D3:D0	Board Jumper (J2)
D3	Board ID4 0=J2:7-J2:8 jumper installed
D2	Board ID3 0=J2:5-J2:6 jumper installed
D1	Board ID2 0=J2:3-J2:4 jumper installed
D0	Board ID1 0=J2:1-J2:2 jumper installed

2.1.4 Timestamp: Local Offset 0x000C

The timestamp will add a 24 bit timestamp value for each data value in the data stream.

D31:24	RESERVED
D23:0	Current timestamp value

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

Defines the Almost Full and Almost Empty Flags for the Tx FIFO. The Almost Full/Empty Flags are status bits in the Channel Control/Status Register, and are edge-triggered interrupt sources to the Interrupt Registers.

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

2.1.6 Channel RX Almost Flags: Local Offset 0x0014 / 0x0024 / 0x0034 / 0x0044

Defines the Almost Full and Almost Empty Flags for the Tx FIFO. The Almost Full/Empty Flags are status bits in the Channel Control/Status Register, and are edge-triggered interrupt sources to the Interrupt Registers.

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

2.1.7 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

2.1.8 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.

D31:24 RESERVED
D23:20 LED Control
Each Channel controls 1 red/green LED on the back of the PCB. See Section 5.3 for more detailed information about the LEDs.
D19 RESERVED

D18:8 Channel Status Bits

D18 Rx FIFO Underflow
D17 Tx FIFO Overflow (Latched)
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)

D7:0 Channel Control Bits

D7 1 = Reset USC ((Pulsed - will automatically clear to '0')

Notes:

- Following a USC Reset, the next access to the USC must be a write of 0x00 to Local Offset 0x100 (Ch1/2) or Local Offset 0x300 (Ch3/4).
- Since two channels share each USC (Ch1 & Ch2, Ch3 & Ch4), resetting a USC will affect both channels.

D6 1 = Reset Channel (Pulsed - will automatically clear to '0')

D5:D4 RESERVED (FIFO Rx/Tx Allocation)

D3 Receive Status Word Enable
1 = Receive status word (RSR) is saved in data stream with every received data word.

D2 Timestamp Enable
1 = 24-bit timestamp word is saved in data stream with every received data word.

D1 1 = Reset Channel Rx FIFO (Pulsed - will automatically clear to '0')

D0 1 = Reset Channel Tx FIFO (Pulsed - will automatically clear to '0').

2.1.9 Channel Sync Detect Byte: Local Offset 0x0050 / 0x0054 / 0x0058 / 0x005C

The Sync Detect Byte allows an interrupt to be generated when the received data matches the Sync Detect Byte.

D31:8 RESERVED

D7:0 Channel Sync Detect Byte
If the data being loaded into the Receive FIFO matches this data byte, an interrupt request (Channel Sync Detect IRQ) will be generated. The interrupt source must be enabled in the Interrupt Control Register in order for an interrupt to be generated.

2.1.10 Interrupt Registers

There are 32 on-board interrupt sources (in addition to USC interrupts and 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	Alternate Level
IRQ0	Channel 1 Sync Detected	Rising Edge	NONE
IRQ1	Channel 1 Tx FIFO Almost Empty	Rising Edge	Falling Edge
IRQ2	Channel 1 Rx FIFO Almost Full	Rising Edge	Falling Edge
IRQ3	Channel 1 USC Interrupt	Level Hi	NONE
IRQ4	Channel 2 Sync Detected	Rising Edge	NONE
IRQ5	Channel 2 Tx FIFO Almost Empty	Rising Edge	Falling Edge
IRQ6	Channel 2 Rx FIFO Almost Full	Rising Edge	Falling Edge
IRQ7	Channel 2 USC Interrupt	Level Hi	NONE
IRQ8	Channel 3 Sync Detected	Rising Edge	NONE
IRQ9	Channel 3 Tx FIFO Almost Empty	Rising Edge	Falling Edge
IRQ10	Channel 3 Rx FIFO Almost Full	Rising Edge	Falling Edge
IRQ11	Channel 3 USC Interrupt	Level Hi	NONE
IRQ12	Channel 4 Sync Detected	Rising Edge	NONE
IRQ13	Channel 4 Tx FIFO Almost Empty	Rising Edge	Falling Edge
IRQ14	Channel 4 Rx FIFO Almost Full	Rising Edge	Falling Edge

IRQ15	Channel 4 USC Interrupt	Level Hi	NONE
IRQ16	Channel 1 Tx FIFO Empty	Rising Edge	Falling Edge
IRQ17	Channel 1 Tx FIFO Full	Rising Edge	Falling Edge
IRQ18	Channel 1 Rx FIFO Empty	Rising Edge	Falling Edge
IRQ19	Channel 1 Rx FIFO Full	Rising Edge	Falling Edge
IRQ20	Channel 2 Tx FIFO Empty	Rising Edge	Falling Edge
IRQ21	Channel 2 Tx FIFO Full	Rising Edge	Falling Edge
IRQ22	Channel 2 Rx FIFO Empty	Rising Edge	Falling Edge
IRQ23	Channel 2 Rx FIFO Full	Rising Edge	Falling Edge
IRQ24	Channel 3 Tx FIFO Empty	Rising Edge	Falling Edge
IRQ25	Channel 3 Tx FIFO Full	Rising Edge	Falling Edge
IRQ26	Channel 3 Rx FIFO Empty	Rising Edge	Falling Edge
IRQ27	Channel 3 Rx FIFO Full	Rising Edge	Falling Edge
IRQ28	Channel 4 Tx FIFO Empty	Rising Edge	Falling Edge
IRQ29	Channel 4 Tx FIFO Full	Rising Edge	Falling Edge
IRQ30	Channel 4 Rx FIFO Empty	Rising Edge	Falling Edge
IRQ31	Channel 4 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, …D31 = IRQ31.)

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 the PCI9056 PLX chip. Likewise, all USC interrupt sources share a single interrupt request back to the interrupt controller and must be further qualified in the USC.

2.1.11 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.

2.1.12 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 (USC 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.

2.1.13 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. Only the USC interrupts are level triggered. All other interrupt sources on the SIO4BX2 are edge triggered.

2.1.14 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 defines 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".

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

The Channel Pin Source Register configures the Output source for the Clocks, Data, RTS, and DCD outputs.

31	30	29	28	27	26	25	24
Transceiver Enable	Termination Disable	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	TxAuxC Source X	TxD Source XD			Unused		DCD Source		RTS Source		USC_DCD Direction		USC_CTS Direction		TxC Source			USC_RXC Source			USC_TxC Source		

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 Termination Disable
For RS422/RS485, the receive signals (RxC, RxD, RxAuxC, CTS, and DCD) have built in termination at the transceivers. These internal terminations may be disabled to allow external terminations (or no terminations) to be used. Setting this bit will disable the internal transceiver termination resistors.

D29 External Loopback Mode
When DCE/DTE Mode is enabled (Bit D31=1), this bit will automatically loopback the TxC/RxC, TxD/RxD, and RTS/CTS signals at the cable (transceivers enabled). This allows the transceivers to be tested in a standalone mode.
Notes:
- The DCE/DTE mode will select the set of signals (DCE or DTE) to be looped back
- Since the transceivers will be enabled in this mode, all external cables should be disconnected to prevent interference from external sources.

D28 DCE/DTE Mode
When DCE/DTE Mode is enabled (Bit D31=1), this bit set the mode to DCE (1) or DTE (0). DCE/DTE mode changes the direction of the signals at the IO Connector.

D27:24 Transceiver Protocol Mode

D27	D26	D25	D24	Transceiver Mode
0	0	0	0	RS422 / RS485
0	0	0	1	RESERVED
0	0	1	0	RS232
0	0	1	1	RESERVED
0	1	X	X	RESERVED
1	X	X	X	RESERVED

D23 Internal Loopback Mode
When DCE/DTE Mode is enabled (Bit D31=1), this bit will automatically loopback the TxC/RxC, TxD/RxD, and RTS/CTS signals internal to the board.

D22 Reserved

D21:19 Cable TxD Output Control
Allows TxD output to be used as a general purpose output.

D21	D20	D19	TxD Source
0	0	X	USC_TxD
0	0	0	Output '0'
0	1	1	Output '1'
1	0	0	Differential Biphase Mark
1	0	1	Differential Biphase Space
1	1	0	Level
1	1	1	Differential Biphase Level

D18:17 Cable TxAuxC Output Control
Defines the Clock Source for the TxAuxC signal to the IO connector.

D18	D17	TxAuxC Source
0	0	Tristate
0	1	On-board Programmable Clock
1	0	Output '0'
1	1	Output '1'

D16:15 Cable DCD Output Source

D16	D15	Output Source	Notes
0	0	USC_DCD Output	USC_DCD field (D12:D11) must equal '11'
0	1	RTS Output	Rx FIFO Almost Full
1	0	'0'	Drive low
1	1	'1'	Drive Hi

D14:13 Cable RTS Output Source

D14	D13	Output Source	Notes
0	0	USC_CTS Output	USC_CTS field (D10:D9) must equal '11'
0	1	RTS Output	Rx FIFO Almost Full
1	0	'0'	Drive low
1	1	'1'	Drive Hi

D12:11 USC_DCD Direction Setup

- If DCD is used as GPIO, set this field to '00' and set Pin Source Register D16:D15 for output / Pin Status Register D3 for input.
- If set, the DCD direction must agree with the USC DCD setup (USC IOCR D13:12) to ensure proper operation.
- If field set to '11' (Output), DCD Source field (D16:15) must be set to '00'.

D12	D11	DCD Buffer Direction	USC IOCR D13:D12 Setup
0	0	Buffer Disabled	XX (Don't Care)
0	1	Input from IO Connector - DCD	0X (Input)
1	0	Reserved	XX (Don't Care)
1	1	Output to IO Connector	1X (Output)

D10:9 USC_CTS Direction Setup

- If CTS is used as GPIO, set this field to '00' and set Pin Source Register D14:D13 for output / Pin Status Register D2 for input.
- If set, the CTS direction must agree with the USC CTS setup (USC IOCR D15:14) to ensure proper operation.
- If field set to '11' (Output), RTS Source field (D14:13) must be set to '00'.

D10	D9	CTS Buffer Direction	USC IOCR D15:D14 Setup
0	0	Tristate	XX (Don't Care)
0	1	Input from IO Connector – CTS	0X (Input)
1	0	Reserved	XX (Don't Care)
1	1	Output to IO Connector	1X (Output)

D8:6 Cable TxC Source

D8	D7	D6	TxC Source
0	0	0	Prog Clock
0	0	1	Inverted Prog Clock
0	1	0	'0' (Drive Line Lo)
0	1	1	'1' (Drive Line Hi)
1	0	0	USC_TxC
1	0	1	USC_RxC
1	1	0	Cable RxC Input
1	1	1	Cable RxAuxC Input

D5:3 USC_RxC Source

The clock source must agree with the USC Clock setup (USC I/O Control Reg D5:3) to ensure the signal is not being driven by both the USC and the FPGA.

D5	D4	D3	USC_RxC Source	USC IOCR D2:D0 Setup
0	0	0	Prog Clock	000 (Input)
0	0	1	Inverted Prog Clock	000 (Input)
0	1	0	'0'	000 (Input)
0	1	1	'1'	000 (Input)
1	0	0	Cable RxC Input	000 (Input)
1	0	1	Cable RxAuxC Input	000 (Input)
1	1	0	RESERVED	--------
1	1	1	Driven from USC	IOCR D2:D0 != 000 (Output)

D2:0 USC_TxC Source

Since this signal is bidirectional (it may be used as either an input or output to the USC), the clock source must agree with the USC Clock setup (USC IO Control Reg D2:0) to ensure the signal is not being driven by both the USC and the FPGA.

D2	D1	D0	USC_TxC Source	USC IOCR D5:D3 Setup
0	0	0	Prog Clock	000 (Input)
0	0	1	Inverted Prog Clock	000 (Input)
0	1	0	'0'	000 (Input)
0	1	1	'1'	000 (Input)
1	0	0	Cable RxC Input	000 (Input)
1	0	1	Cable RxAuxC Input	000 (Input)
1	1	0	RESERVED	--------
1	1	1	Driven from USC	IOCR D5:D3 != 000 (Output)

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

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.

D31:D10 RESERVED
D9 TxAuxC Output
D8 RxAuxC Input
D7 DCD Output
D6 RTS Output
D5 TxD Output
D4 TxC Output
D3 DCD Input
D2 CTS Input
D1 RxD Input
D0 RxC Input

2.1.13 Programmable Clock Registers: Local Offset 0x00A0 / 0x00A4 / 0x00A8 / 0xAC

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 3.6 for more information.

2.1.14 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.

D31:16	Number of words in Rx FIFO
D15:D0	Number of words in Tx FIFO

2.1.15 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.

D31:16	Size of installed Rx FIFO
D15:D0	Size of installed Tx FIFO

2.1.16 FW Type ID Register: Local Offset 0x00F8

This register allows boards to change functionality on each channel. Currently, a channel can only be defined as Standard or Sync. For SIO4BX-Sync information, please refer to the PCIe-SIO4BX2-SYNC manual.

D31:D24	Channel 4 FW Type –> 01 = Standard / 04 = Sync
D23:D16	Channel 3 FW Type –> 01 = Standard / 04 = Sync
D15:D8	Channel 2 FW Type –> 01 = Standard / 04 = Sync
D7:D0	Channel 1 FW Type –> 01 = Standard / 04 = Sync

2.1.17 Features Register: Local Offset 0x00FC (0x00197AF4)

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

D31:21	RESERVED
D20	**1** = No Rx Status byte (std only)
D19:D18	**10** = Internal Timestamp (std only)
D17:D16	**01** = FPGA Reprogram field
D15:D14	**01** = Configurable FIFO space
D13	**1** = FIFO Test Bit
D12	**1** = FW Type Reg
D11:8	Features Rev Level **0xA** = BX level
D7	**1** = Demand Mode DMA Single Cycle Disable feature implemented
D6	**1 =** Board Reset
D5	**1 =** FIFO Counters/Size
D4	**1**
D3:0	Programmable Clock Configuration **0x4** = Two CY22393 - 6 Oscillators

2.2 Universal Serial Controller Registers

The internal registers of the Zilog Z16C30 Universal Serial Controller (USC) are memory mapped into Local Address space. It is beyond the scope of this manual to provide comprehensive USC programming information. For detailed programming information, please refer to the Zilog High Speed Communication Controller Product Specifications Databook for the Z16C30 and the Zilog Z16C30USC User's Manual. These manuals may be obtained directly from Zilog (www.zilog.com), or copies of these manuals may be downloaded from the General Standards website (www.generalstandards.com).

Some specific setup information may be needed for a driver to interface to the USC. Typically, the driver will handle the hardware specific characteristics and the end user will only need to be concerned with the driver interface - the following hardware setup information may be safely ignored. If you aren't sure if you need this information, you probably don't.

2.2.1 USC Reset

The four serial channels are implemented in two Z16C30 Universal Serial Controllers – Channels 1 and 2 share one USC, and Channels 3 and 4 share the other. This implementation is important to realize since resetting a Z16C30 chip will have an effect on two serial channels. Since the USC chips are typically reset upon initialization, this means a "Reset USC" for Channel 1 will also "Reset USC" for Channel 2. In addition to making the second reset redundant and unnecessary, a Reset USC on one channel may inadvertently adversely affect normal operation on the second channel. Therefore, care must be exercised when resetting a USC (USC Reset bit in the Board Control Register), especially in multithreaded environments.

Since the USC Reset physically resets the USC, the first access to the USC following the reset must reinitialize the BCR in the USC. To complete the Reset process, the user should write data 0x00 to USC base address offset 0x100 or 0x300 to correctly initialize the BCR. Following this initial byte write, the USC may be accessed normally.

Due to the ability for a USC Reset to affect two channels, it is recommended that a single USC Channel be Reset via the RTReset bit of the USC Channel Command/Address Register (CACR).

2.2.2 8-Bit USC Register Access

As the USC has a configurable bus interface, the USC must be set to match the 8-bit non-multiplex interface implementation of the SIO4BX2. This setup information must be programmed into the USC Bus Configuration Register (BCR) upon initial power up and following every hardware reset of the USC. The BCR is accessible only following a USC hardware reset – the first write to the USC following a USC Reset programs the BCR. Even though the Zilog manual states the BCR has no specific address, the driver must use the channel USC base address – 0x100 for Ch 1 & Ch 2, 0x300 for Ch 3 & Ch 4 – as the BCR address. Failure to do so may result in improper setup. Since the user interface to the USC is an 8 bit interface, the software only needs to set the lower byte to 0x00 (hardware implementation will program the upper byte of the BCR).

2.2.3 USC Data Transfer

Although the Z16C30 USC contains 32 byte internal FIFOs for data transfer, these are typically not used on the SIO4BX2. Since the SIO4BX2 has much deeper external FIFOs (or internal FPGA FIFOs), the internal USC FIFOs are setup to immediately transfer data to/from the external FIFOs. Immediate transfer of received data to the external FIFOs eliminates the possibility of data becoming "stuck" in the USC internal receive FIFOs, while bypassing the USC internal transmit FIFOs ensures better control of the transmit data.

In order to automatically transfer data to and from the external FIFOs, the USC should use DMA to request a data transfer whenever one byte is available in the USC internal FIFOs. This "DMA" should not be confused with the DMA of data from the SIO4BX2 external FIFOs to the PCI interface. To accomplish the USC-to-External FIFO DMA transfer, the TxReq/RxReq pins should be set as DMA Requests in the IOCR, and the TxAck/RxAck pins should be set as DMA Acknowledge inputs in the HCR. In addition, the Tx Request Level should be set to 0x1F (31) using TCSR/TICR and the Rx Request Level should be set to 0 using RCSR/RICR. See Z16C30 manual for further details on programming the DMA request levels.

2.2.4 USC Register Memory Map

To access the USC in 8-bit mode, the driver is required to access the upper and lower bytes of each register independently. The odd address byte will access the upper byte of each register (D15-D8), and the even address byte will access the lower byte (D7-D0). Each USC register must be accessed independently as a byte access– the software cannot perform word or long word accesses to the USC registers.

The USC register map is provided below. The Channel Offset Address depicted is from the Channel Base Address – (Ch 1 Base Address = 0x100, Ch 2 Base Address = 0x200, Ch 3 Base Address = 0x300, Ch 4 Base Address = 0x400). For further programming details, please refer to the Zilog Z16C30 data books.

Channel Offset Address	Access*	Register Name
0x01 / 0x00	CCAR Hi / Lo	Channel Command / Address Register
0x03 / 0x02	CMR Hi / Lo	Channel Mode Register
0x05 / 0x04	CCSR Hi / Lo	Channel Command / Status Register
0x07 / 0x06	CCR Hi / Lo	Channel Control Register
0x11 / 0x10	CMCR Hi / Lo	Clock Mode Control Register
0x13 / 0x12	HCR Hi / Lo	Hardware Configuration Register
0x17 / 0x16	IOCR Hi/Lo	I/O Control Register
0x19 / 0x18	ICR Hi / Lo	Interrupt Control Register
0x1B / 0x1A	DCCR Hi / Lo	Daisy Chain Control Register
0x1D / 0x1C	MISR Hi / Lo	Miscellaneous Interrupt Status Register
0x1F / 0x1E	SICR Hi / Lo	Status Interrupt Control Register
0x20	RDR	Receive Data Register
0x23 / 0x22	RMR	Receive Mode Register
0x25 / 0x24	RCSR Hi / Lo	Receive Command / Status Register
0x27 / 0x26	RICR Hi / Lo	Receive Interrupt Control Register
0x29 / 0x28	RSR Hi / Lo	Receive Sync Register
0x2B / 0x2A	RCLR Hi / Lo	Receive Count Limit Register
0x2D / 0x2C	RCCR Hi / Lo	Receive Character Count Register
0x2F / 0x2E	TC0R	Time Constant 0 Register
0x30	TDR	Transmit Data Register
0x33 / 0x32	RMR	Transmit Mode Register
0x35 / 0x34	TCSR Hi / Lo	Transmit Command / Status Register
0x37 / 0x36	TICR Hi / Lo	Transmit Interrupt Control Register
0x39 / 0x38	TSR Hi / Lo	Transmit Sync Register
0x3B / 0x3A	TCLR Hi / Lo	Transmit Count Limit Register
0x3D / 0x3C	TCCR Hi / Lo	Transmit Character Count Register
0x3F / 0x3E	TC1R	Time Constant 1 Register

CHAPTER 3: PROGRAMMING

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 four unique reset sources: a USC Reset, a Channel Reset, a Transmit FIFO Reset, and a Receive FIFO Reset. All 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), as well as the Channel Reset.

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 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

Unlike previous SIO4 boards which had ordering options for different FIFO sizes, the PCIe-SIO4BX2 always uses 32k byte deep FIFOs.

3.3 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.

3.4 Programmable Oscillator / Programmable Clocks

Four On-Board Programmable Oscillators provide each channel with a unique programmable clock source. 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 routines are written in C, but may be ported for user specific applications. Contact GSC for help in porting these routines.

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.

3.5 Clock Setup

Figure 3-1 shows the relationship of the various clock sources on the SIO4BX2 board. These clock sources can be most simply viewed in three sections: On-Board Programmable Clocks, IO Connector Clocks, and USC Clocks.

The Programmable Clocks consist of a one on-board programmable PLL (with post divider) per channel. This allows each channel to have a unique programmable clock (ProgClk).

The IO Connector Clocks consist of a Receive Clock (RxC), a Transmit Clock (TxC), and a bidirectional Auxiliary Clock (AuxC) for each channel. RxC is always an input and may be used as a clock source for either TxC or the USC Clocks. The Auxiliary clock may be set as an input (RxAuxC) or output (TxAuxC).

TxC is always an output. It may be generated from ProgClk, inverted ProgClk, RxC, RxAuxC, either of the USC clocks (USC_TxC or USC_RxC), or forced hi or low (for software control). The TxC Source is controlled by bits D8-D6 of the Pin Source Register.

The USC Clocks (USC RxC and USC TxC) are bidirectional signals. Even though the names of these clocks seem to imply a receive clock and a transmit clock, both clocks are fully programmable and identical in function – either clock may be used for transmit or receive. The USC clocks may be sourced from either the USC or the FPGA (via the Pin Source register). The user must be careful to ensure that both the USC and Pin Source Register are setup to agree. If a USC clock is set as an output in the USC, it should be programmed as an input in the Pin Source register. Likewise, if a USC clock source is driven from the Pin Source register, the user should program the pin as an input to the USC.

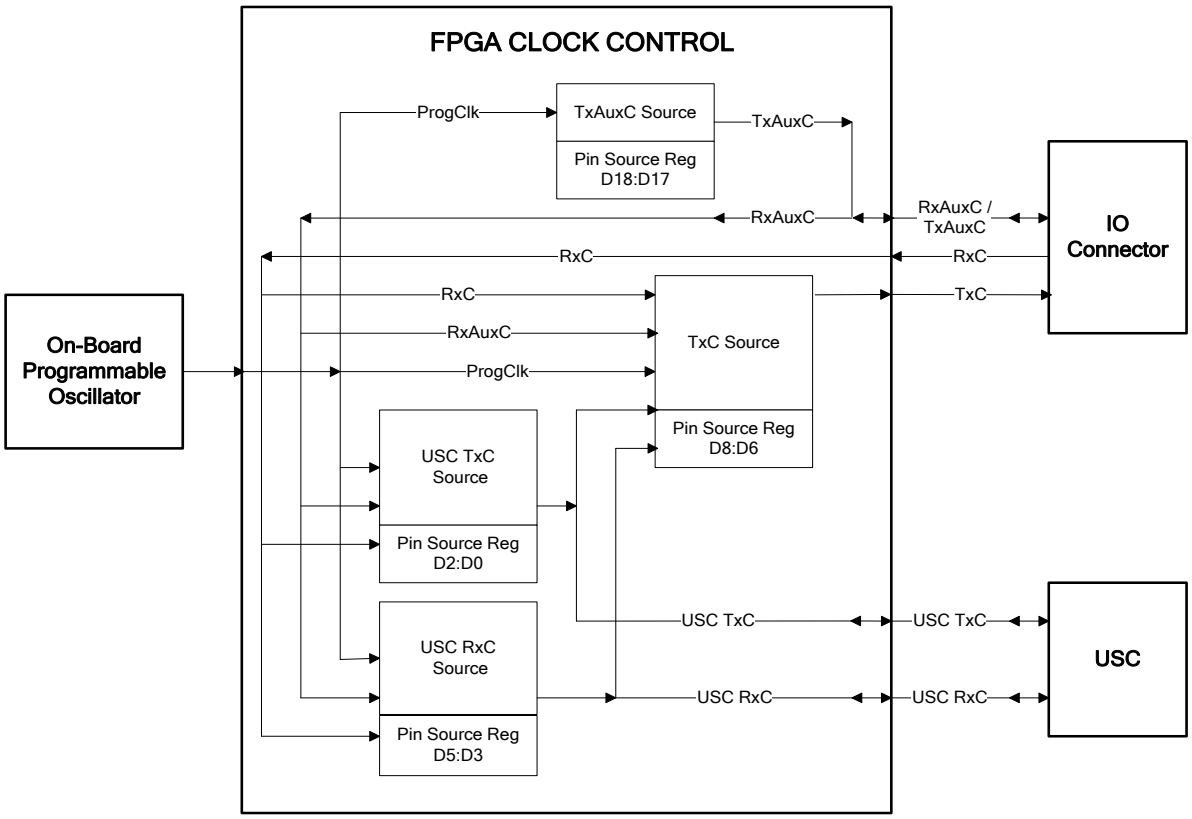


Figure 3-1 – Clock Configuration

The programmable clocks on the SIO4BX2 provides flexibility to handle almost any clock configuration scenario. However, this flexibility can also complicate the clock setup, especially for simple setups. The following guidelines are typical asynchronous and synchronous setups which should work for most setups.

In asynchronous mode, the clock does not need to be transmitted with the data. Therefore, the USC Clock pins will be used for the input baud rate clock. Since the RxC and TxC pins have identical functions, the RxC and TxC pins may be used interchangeably. The async baud rate clock will be 16x / 32x / or 64x the actual baud rate due to the async oversampling. This oversample rate is set in the USC Channel Mode Register when async mode is selected. The simplest method will be to program the channel programmable clock to be 16/32/64 times the desired baud rate and use this clock as the source for the TxC/RxC pin. Section 2.1.11 describes how to program the Pin Source Register to set TxC / RxC = Programmable Clock. The USC should be programmed such that TxC / RxC is an input (in the USC I/O Control Register), and the USC baud rate generator will be bypassed completely. If both Rx and Tx are operating at the same baud rate, the same USC clock pin can be used for both the transmit and receive clocks.

For synchronous modes, the clock is transmitted and received on the cable along with the data. This can present a problem since the USC only has two clock pins. Since one clock is necessary for receive clock and the other is necessary for the transmit clock, there is no clock pin available for an input to the USC baud rate generators. The on-board programmable clocks provide a solution for this situation. By using the programmable oscillator and the programmable clock post-divider, the on-board programmable clock can usually be set directly to the desired transmit baud rate. The USC TxC pin and the Cable TxC are both set equal to the Programmable Clock in the Pin Source Register. The USC RxC pin is used for the receive clock from the cable interface, so it will be set to the

cable RxC in the Pin Source Register. Since the FPGA will source both USC clocks, they must be programmed as inputs in the USC I/O Control Register.

The preceding suggestions should work for most applications. The default Pin Source Register value should set the clocks to work with both scenarios – USC TxC pin = Programmable Clock, USC RxC Pin = Cable RxC, Cable TxC = Programmable Clock. (For async, use USC TxC is input clock).

3.6 Multiprotocol Transceiver Control

The SIO8BX2 has multiprotocol transceivers which allow RS422/RS485 and RS232 modes. The mode is set by the Protocol Mode field in the Pin Source Register.

3.7 DCE/DTE Mode

As all signals are bidirectional, the DCE or DTE mode will set the direction for each signal. For the transceivers to be configured as either DTE or DCE, set the DCE/DTE Enable bit in the Pin Source register (D31). The following table gives the input/output configuration for each signal: The DCD and AuxC direction is set in the Pin Source register fields, independent of DCE/DTE mode.

Signal	DTE	DCE
TxC	TxC Out	RxC In
RxC	RxC In	TxC Out
TxD	TxD Out	RxD In
RxD	RxD In	TxD Out
RTS	RTS Out	CTS In
CTS	CTS In	RTS Out
DCD	Direction controlled by Pin Source Reg D16:15	
AuxC	Direction controlled by Pin Source Reg D18:17	

3.8 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 SIO4BX2 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 PCI9056 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 PCI9056 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 PCI9056. 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 SIO4BX2 onboard FIFOs with the least amount of CPU overhead. The PCI9056 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 PCI9056 I/O Accelerator from PLX Technology. The PCI interface is compliant with the 5V, 66MHz 32-bit PCI Specification 2.2. The PCI9056 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 PCI9056 are not utilized in this design, it is beyond the scope of this document to duplicate the PCI9056 User's Manual. Only those features, which will clarify areas specific to the PCIe-SIO4BX2 are detailed here. Please refer to the PCI9056 User's Manual (See Related Publications) for more detailed information. Note that the BIOS configuration and software driver will handle most of the PCI9056 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 9056 contains many registers, many of which have no effect on the SIO4BX2 performance. The following section attempts to filter the information from the PCI9056 manual to provide the necessary information for a SIO4BX2 specific driver.

The SIO4BX2 uses an on-board serial EEPROM to initialize many of the PCI9056 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 SIO4BX2 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	0x9056	PCI9056
Sub-Vendor ID	0x10B5	PLX Technology
Sub-Device ID	0x3198	GSC SIO4B

The configuration registers also setup the PCI IO and Memory mapping for the SIO4BX2. The PCI9056 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 PCI9056 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 SIO4BX2 memory size is initialized to 4k Bytes. All other Local Registers initialize to the default values described in the PCI9056 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 SIO4BX2. All other Runtime Registers initialize to the default values described in the PCI9056 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 SIO4BX2 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 are 8 bits wide, the register access is still 32bit. 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. (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 PCIe-SIO4BX2:

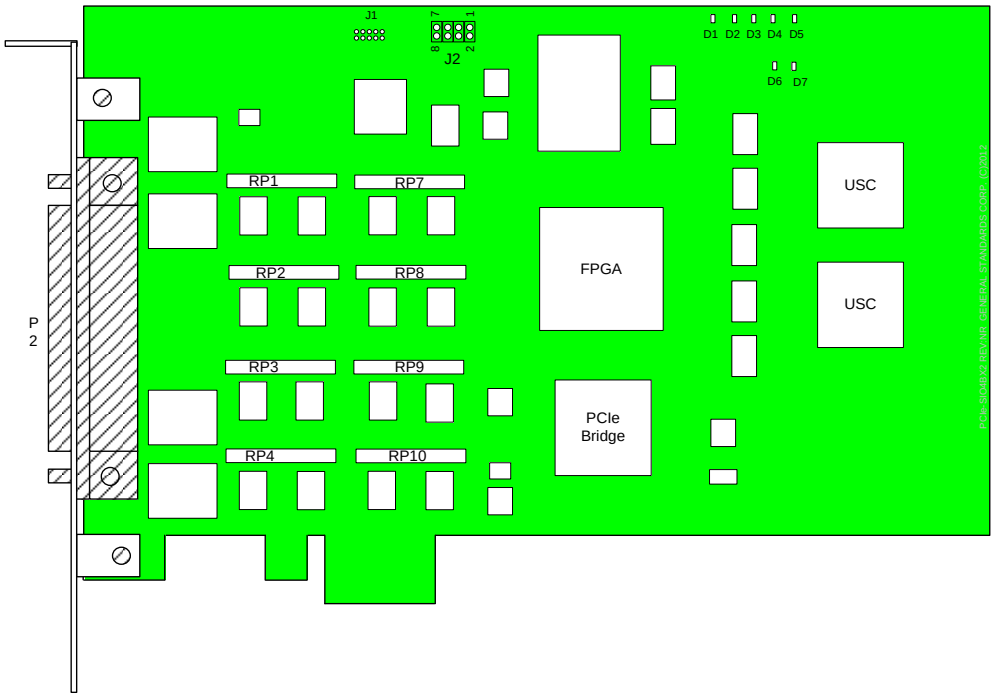


Figure 5-1: Board Layout – Top

5.1 Board ID Jumper J2

Jumper J2 allows the user to set the Board ID in the Board Status Register (See Section 2.1.3). This is useful to uniquely identify a board if more than one SIO4BX2 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. Refer to Figure 5-1 for Jumper J2 location.

J2 Jumper	Description	Notes
1 - 2	Board ID 1	Board ID 1 in Board Status Register (D0)
3 - 4	Board ID 2	Board ID 2 in Board Status Register (D1)
5 - 6	Board ID 3	Board ID 3 in Board Status Register (D2)
7 - 8	Board ID 4	Board ID 4 in Board Status Register (D3)

5.2 Termination Resistors

The PCIe-SIO4BX2 transceivers have built in termination resistors for the RS422/RS485 mode. The built in RS422/RS485 termination is a 120 Ohm parallel termination. If desired, the internal termination resistors may be disabled by setting bit D30 in the Pin Source Register.

The board is designed with socketed external parallel termination (if a different value than the internal termination is required). The external termination resistors are 8 pin SIPs. There are 8 termination SIPs – RP1-RP4, RP7-RP10. The external parallel resistors are for RS422/RS485 termination only. Refer to Figure 5-1 for resistor pack locations.

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

5.3 LEDs

Five bicolor LEDs (D1-D5) are accessible via software. Refer to Figure 5-1 for these LED locations.

LED D1 is controlled from the Board Control Register. LED_D1 Red is controlled by D25, and LED_D1 Green is controlled D24

The remaining 4 LEDs are controlled from D23:D20 of the four Channel Control Registers. Each Channel Control Register controls 1 LED. If D23:D22="10", the Red LED will turn off. Likewise, if D23:D22="11", the Red LED will turn on. D21:D20 controls the Green LED in the pair.

LED_D2 is controlled by Ch 4
LED_D3 is controlled by Ch 3
LED_D4 is controlled by Ch 2
LED_D5 is controlled by Ch 1

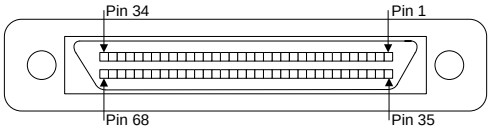
Additionally, if all the LED controls are set to 0 in all four of the Channel Control Registers (power up default), the LEDs will display the lower 4 bits of the firmware revision in Green LED_D2 to LED_D5.

LED D6 displays the PCIe link status. It will be off for “No PCIe Link” and on when the PCIe link is established. (Note: for Rev NR PCB, the D6 will be On = No Link / Off = Link).

The remaining LED (D7) displays the firmware status. This LED should flash at power up or after a PCIe reset, then it will turn off. The LED should be off during normal operation.

5.4 Interface Connector

User I/O Connector: 68-pin SCSI connector (female) (P2)
Part Number: AMP/TYCO 787170-7
Mating Connector: AMP/TYCO 749111-6 (or equivalent)



Note: RS422/RS485 mode or RS232 mode is set on a per channel basis

Pin #	RS422/RS485		RS232		Pin #	RS422/RS485		RS232	
	DTE	DCE	DTE	DCE		DTE	DCE	DTE	DCE
1	AUXC1+		Unused (Hi)		35	AUXC3+		Unused (Hi)	
2	AUXC1-		AUXC1		36	AUXC3-		AUXC3	
3	DCD1+		Unused (Hi)		37	DCD3+		Unused (Hi)	
4	DCD1-		DCD1		38	DCD3-		DCD3	
5	CTS1+	RTS1+	Unused (Hi)		39	CTS3+	RTS3+	Unused (Hi)	
6	CTS1-	RTS1-	CTS1	RTS1	40	CTS3-	RTS3-	CTS3	RTS3
7	RXD1+	TXD1+	Unused (Hi)		41	RXD3+	TXD3+	Unused (Hi)	
8	RXD1-	TXD1-	RXD1	TXD1	42	RXD3-	TXD3-	RXD3	TXD3
9	RXC1+	TXC1+	Unused (Hi)		43	RXC3+	TXC3+	Unused (Hi)	
10	RXC1-	TXC1-	RXC1	TXC1	44	RXC3-	TXC3-	RXC3	TXC3
11	RTS1+	CTS1+	Unused (Hi)		45	RTS3+	CTS3+	Unused (Hi)	
12	RTS1-	CTS1-	RTS1	CTS1	46	RTS3-	CTS3-	RTS3	CTS3
13	TXD1+	RXD1+	Unused (Hi)		47	TXD3+	RXD3+	Unused (Hi)	
14	TXD1-	RXD1-	TXD1	RXD1	48	TXD3-	RXD3-	TXD3	RXD3
15	TXC1+	RXC1+	Unused (Hi)		49	TXC3+	RXC3+	Unused (Hi)	
16	TXC1-	RXC1-	TXC1	RXC1	50	TXC3-	RXC3-	TXC3	RXC3
17	SGND1		SGND1		51	SGND3		SGND3	
18	SGND2		SGND2		52	SGND4		SGND4	
19	CTS2+	RTS2+	Unused (Hi)		53	CTS4+	RTS4+	Unused (Hi)	
20	CTS2-	RTS2-	CTS2	RTS2	54	CTS4-	RTS4-	CTS4	RTS4
21	RXD2+	TXD2+	Unused (Hi)		55	RXD4+	TXD4+	Unused (Hi)	
22	RXD2-	TXD2-	RXD2	TXD2	56	RXD4-	TXD4-	RXD4	TXD4
23	RXC2+	TXC2+	Unused (Hi)		57	RXC4+	TXC4+	Unused (Hi)	
24	RXC2-	TXC2-	RXC2	TXC2	58	RXC4-	TXC4-	RXC4	TXC4
25	RTS2+	CTS2+	Unused (Hi)		59	RTS4+	CTS4+	Unused (Hi)	
26	RTS2-	CTS2-	RTS2	CTS2	60	RTS4-	CTS4-	RTS4	CTS4
27	TXD2+	RXD2+	Unused (Hi)		61	TXD4+	RXD4+	Unused (Hi)	
28	TXD2-	RXD2 -	TXD2	RXD2	62	TXD4-	RXD4-	TXD4	RXD4
29	TXC2+	RXC2+	Unused (Hi)		63	TXC4+	RXC4+	Unused (Hi)	
30	TXC2-	RXC2-	TXC2	TXC2	64	TXC4-	RXC4-	TXC4	RXC4
31	DCD2+		Unused (Hi)		65	DCD4+		Unused (Hi)	
32	DCD2-		DCD2		66	DCD4-		DCD4	
33	AUXC2+		Unused (Hi)		67	AUXC4+		Unused (Hi)	
34	AUXC2-		AUXC2		68	AUXC4-		AUXC4	

Table 1- Front Panel (P2) IO Connections

CHAPTER 6: ORDERING OPTIONS

6.0 Ordering Information

PCIe – SIO4BX2 - <*Temperature*>

Option	Valid Selections	Description
Temperature	<blank>	0ºC to +70ºC – Commercial (Standard)
	I	-40ºC to +85ºC – Industrial

6.1 Interface Cable

General Standards Corporation can provide an interface cable for the SIO4BX2 board. This standard cable is a twisted pair cable for increased noise immunity. Several standard cable lengths are offered, or the cable length can be custom ordered to the user's needs. Versions of the cable are available with connectors on both ends, or the cable may be ordered with a single connector to allow the user to adapt the other end for a specific application. A standard cable is available which will breakout the serial channels into eight DB25 connectors. Shielded cable options are also available. Please consult our sales department for more information on cabling options and pricing.

6.2 Device Drivers

General Standards has developed many device drivers for The SIO4BX2 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.

6.3 Custom Applications

Although the SIO4BX2 board provides extensive flexibility to accommodate most user applications, a user application 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 SIO4BX2 boards. Please consult our sales department with your specifications to inquire about a custom application

APPENDIX A: PROGRAMMABLE OSCILLATOR PROGRAMMING

The 4 on-baord clock frequencies are supplies via two Cypress Semiconductor CY22393 Programmable Clock Generatosr. In order to change the clock frequencies, this chip must be reprogrammed. This document supplies the information necessary to reprogram the on-board clock frequencies. GSC has developed routines to calculate and program the on-board oscillator for a given set of frequencies, so it should not be necessary for the user need the following 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 (CLOCK RAM) 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 dat sheet.

For the SIO4BX2, a second programmable oscillator has been added to assure that each channel has a dedicated PLL. (The older SIO4BX uses 3 PLLs in a single CY22393 to generate all four clocks). To implement this, a second CLOCK RAM block was added. CLOCK RAM1 programs the first CY22393 (using CLKA=Ch1_Clk, CLKB=Ch2_Clk, CLKC=Ch3_Clk), and CLOCK_RAM2 programs the second CY22393 (using CLKD=Ch4_Clk). Since the original SIO4BX (with a single CY22393) used CLKD for Ch4_Clk, the same code can be made to support both schemes by simply programming CLKD of the first CY22393.

Each CLOCK RAM block is accessed through 2 registers – Address Offset at local offset 0x00A0 and Data at local ffset at 0x00A4 (CLOCK RAM1) or 0x00AC (CLOCK RAM2). 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 Data1 Register

Provides access to the CLOCK RAM1 pointed to by the RAM Addr Register.

0x00AC – RAM Data2 Register

Provides access to the CLOCK RAM2 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. These post-dividers will further divide down the input clock from the programmable oscillator to provide for slow baud rates. 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.

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 RAM is defined as follows: RAM Address 0x08–0x57 correspond directly to the CY22393 registers.

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
0x41	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

APPENDIX B: FIRMWARE REVISIONS / FEATURES REGISTER

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 (0xE5100100)

D31:16 HW Board Rev 0xE510 PCIe-SIO4BX2 Rev NR

- 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
  - **5** = PCIe
  - 6 = XMC
- D23:D20 HW Board (sub-field of form factor)
  - 0 = PCIe4-SIO8BX2
  - **1** = PCIe-SIO4BX2
- D19:D16 HW Board Rev (lowest rev for firmware version)
  - **0**=NR

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

0x00 – Initial release from PMC66-SIO4BXR v117

Feature Register - Local Offset 0xFC (0x00197AF4)

D31:D21	Unused
D20	**1** - Rx Status byte inserted in FIFO
D19:D18	Timestamp 01 = single external clock **10** = single internal clock
D17:D16	FPGA Reprogram field **01** = Present 00 = Not Present
D15:D14	Configurable FIFO space **01** - Rx/Tx select. Up to 32k deep FIFOs
D13	**1** = FIFO Test Bit
D12	**1** = FW Type Reg
D11:D8	FW Feature Level (Set at common code level) 0x01 = RS232 support, Pin Source Change 0x02 = Multi-Protocol support 0x03 = Common Internal/External FIFO Support 0x04 = FIFO Latched Underrun/Overrun/Level 0x05 = Demand mode DMA Single Cycle for Tx 0x06 = DMA_Single_Cycle_Dis, updated Pin_Src 0x07 = Rx Underrun Only, Reset Status 0x08 = Clock to 50Hz with 10Hz resolution 0x09 = No Legacy Support (No Clock Control Register) **0x0A** = Falling Int fix
D7	**1** = DMA Single Cycle Disable
D6	**1** = Board Reset, FIFO present bits
D5	**1** = FIFO Size/Counters present
D4	**1** = FW ID complies with this standard
D3:D0	Clock Oscillator 0x0 = Fixed 0x1 = ICD2053B (1 Osc) 0x2 = ICD2053B (4 Osc) 0x3 = CY22393 (4 Osc) **0x4** = 2 x CY22393 (6 Osc)