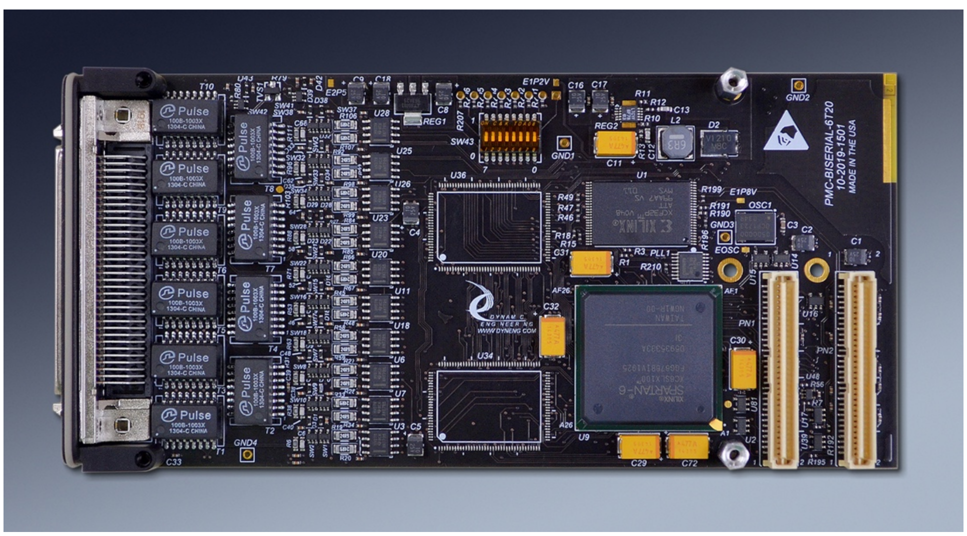
DYNAMIC ENGINEERING
150 DuBois St., Suite C, Santa Cruz, CA. 95060
831-457-8891
sales@dyneng.com
www.dyneng.com
Est. 1988

User Manual

PMC-BiSerial-6T20-HW1

20 Transformer coupled ports
2 RS-485 IO
Bi-directional Manchester Interface
PMC Module



Manual Revision 1p0 4/2/20
Corresponding Hardware: Revision 01
10-2019-1501
Corresponding Firmware: Revision 05

PMC-BiSerial-6T20 HW1
20 port Bi-Directional transformer coupled Manchester Encoded PMC Module

Dynamic Engineering
150 DuBois St., Suite C
Santa Cruz, CA 95060
831-457-8891
FAX: 831-457-4793

This document contains information of proprietary interest to Dynamic Engineering. It has been supplied in confidence and the recipient, by accepting this material, agrees that the subject matter will not be copied or reproduced, in whole or in part, nor its contents revealed in any manner or to any person except to meet the purpose for which it was delivered.

Dynamic Engineering has made every effort to ensure that this manual is accurate and complete. Still, the company reserves the right to make improvements or changes in the product described in this document at any time and without notice. Furthermore, Dynamic Engineering assumes no liability arising out of the application or use of the device described herein.

The electronic equipment described herein generates, uses, and can radiate radio frequency energy. Operation of this equipment in a residential area is likely to cause radio interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Dynamic Engineering's products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of Dynamic Engineering.

This product has been designed to operate with compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage.

Table of Contents

List of Figures

Product Description

PMC-BiSerial-6T20 is part of the PMC Module family of modular I/O components by Dynamic Engineering. PMC-BiSerial-6T20 is capable of providing multiple serial or parallel protocols. The 6T20 was ported from the 3T20 and updated to incorporate a Spartan VI FPGA, new RS485 drivers and termination switches, plus using industrial temperature components. The updated design is based on PMC-BiSerial-VI. Designs can be migrated between any of these platforms.

The HW1 protocol implemented is the same as HW1 implemented on the PMC-BiSerial-III/VI platform, 20 transformer coupled, Manchester encoded IO are provided with this version. The memory map for the 6T20-HW1 is the same as the 3T20-HW1 to allow easy of porting. The vendor and cardid were also retained.

Custom interfaces are available. Please send your requirements – We will redesign the state machines and create a custom interface protocol.

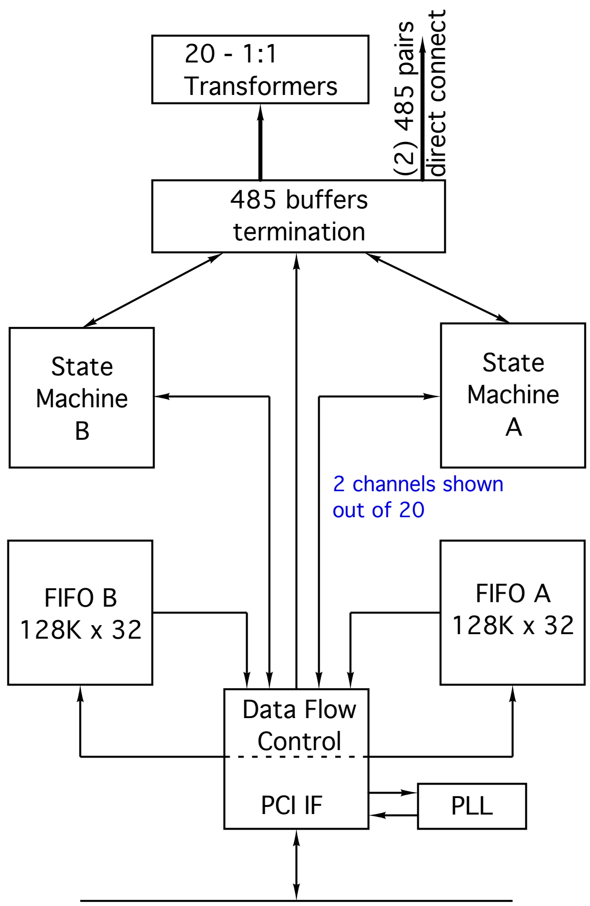


FIGURE 1 PMC BISERIAL-6T20 BASE BLOCK DIAGRAM

The standard configuration shown in Figure 1 has allowance for two extewrnal FIFOs. The larger memories are useful when large streams of data without flow control must be maintained. Usually the internal memory is sufficient. The HW1 model is Dual Port RAM based and does not use the external memory.

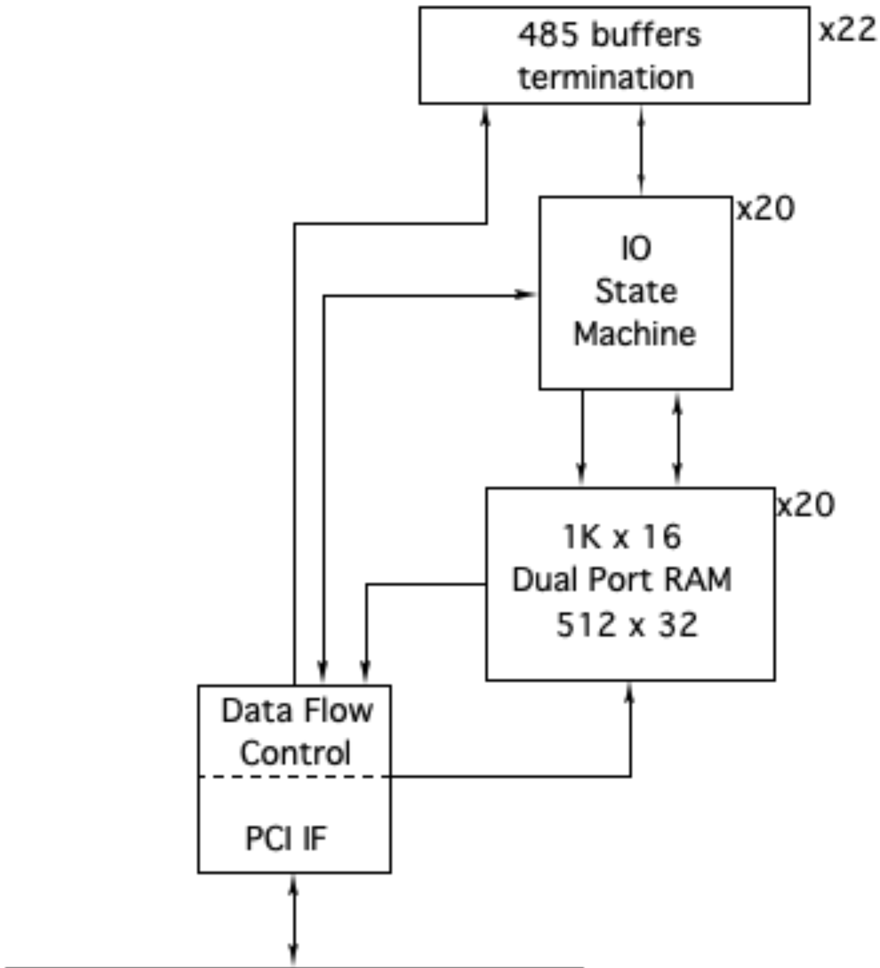
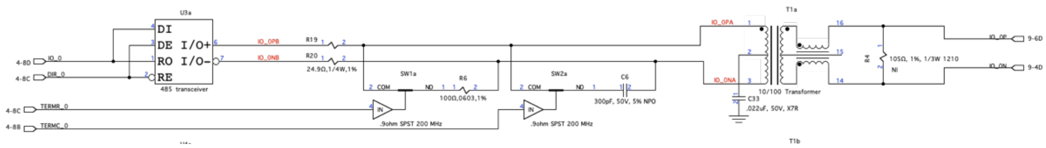


FIGURE 2 PMC BISERIAL-6T20-HW1 BLOCK DIAGRAM

The HW1 implementation has 20 Dual-Port RAMs (DPR) using the internal block RAM of the Xilinx. Each port has an associated DPR. Each DPR is configured to have a 32-bit port on the PCI side, and a 16-bit port on the IO side. When operating in the half-duplex mode the DPR is split in half to provide both a transmit, and a receive section. In the uni-directional mode the full DPR can be used for transmit or receive data.

The “extra” two 485 buffers are controlled via registers. Direction, termination and data ports are provided.

The IO from the state-machine are connected to the IO connector via transformers with termination options. HW1 has 25 ohm series resistors between the transceiver and transformer installed, plus the separately programmable parallel 100 ohm and 300 pF termination at the transceiver. The parallel load resistor [after transformer] are not installed. Please refer to the following schematic section:



The data rate is programmable. The base rates are 400 KHz and 5 MHz. Usually the 5 MHz rate is used with the uni-directional mode and the 400 KHz. in the half-duplex mode. The PLL can be selected for user defined transmit frequencies, while the receiver is always referenced to the on-board oscillator.

The data is Manchester encoded. The hardware uses a higher rate clock to separate the clock and data embedded within the Manchester data stream. The data I/O within the FPGA are programmable to be register controlled or state-machine controlled. Any or all of the bits can be used as a parallel port. The transformers will tend to block DC levels.

22 differential I/O are provided at the front bezel for the serial signals. The drivers and receivers conform to the RS-485 specification (exceeds RS-422 specification). The RS-485 input signals are selectively terminated with 100Ω. The termination resistors are discrete to allow flexible termination options for custom formats and protocols.   The terminations and transceivers are programmable through the Xilinx device to provide the proper mix of outputs and inputs and terminations needed for a specific protocol implementation. The terminations are programmable for all I/O.

All configuration registers support read and write operations for maximum software convenience; all addresses are long word aligned.

PMC-BiSerial-6T20 conforms to the PMC and CMC draft standards. This guarantees compatibility with multiple PMC Carrier boards. Because the PMC may be mounted on different form factors, while maintaining plug and software compatibility, system prototyping may be done on one PMC Carrier board, with final system implementation on a different one.

PMC-BiSerial-6T20 uses a 10 mm inter-board spacing for the front panel, standoffs, and PMC connectors. The 10 mm height is the "standard" height and will work in most systems with most carriers.

Interrupts are supported by PMC-BiSerial-6T20-HW1. An interrupt can be configured to occur at the end of a transmitted packet or message. An interrupt can be set at the end of a received packet. All interrupts are individually maskable, and a master interrupt enable is also provided to disable all interrupts simultaneously. The current status is available for the state-machines making it possible to operate in a polled mode. I2O interrupt processing is also available.

Theory of Operation

PMC-BiSerial-6T20-HW1 features a Xilinx FPGA. The FPGA contains all of the registers and protocol controlling elements of the design. Only the transceivers, and switches are external to the Xilinx device.

PMC-BiSerial-6T20 is a part of the PMC Module family of modular I/O products. It meets the PMC and CMC draft Standards. In standard configuration, the PMC BISERIAL-6T20 is a Type 1 mechanical with low-profile components on the back of the board and one slot wide, with 10 mm inter-board height. Contact Dynamic Engineering for a copy of this specification. It is assumed that the reader is at least casually familiar with this document and logic design.

The PCI interface to the host CPU is controlled by a logic block within the Xilinx. The PMC-BiSerial-6T20 design requires one wait-state for read or write cycles to any address. The HW1 design does not use DMA as the memories are small and numerous. PMC-BiSerial-6T20 is capable of implementing DMA.

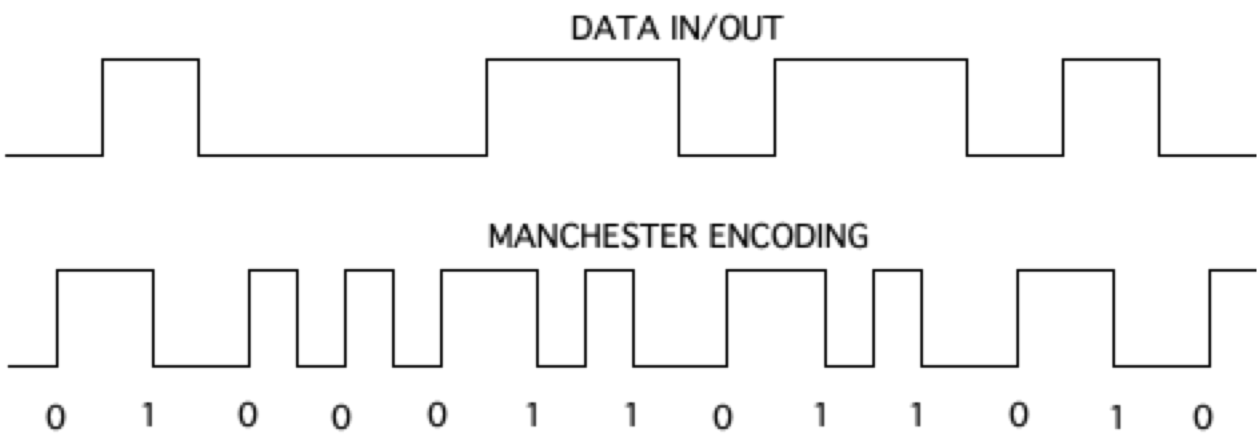


FIGURE 3 PMC-BISERIAL-6T20-HW1 MANCHESTER TIMING DIAGRAM

PMC-BiSerial-6T20 can support many protocols. PMC-BISERIAL-6T20-HW1 uses Manchester serial encoded data and clock. Data is sent in 16-bit words concatenated for multiple word transfers. The Manchester timing is shown in the Figure 3.

State machines within the FPGA control all transfers between the internal DPR and FPGA logic, and the FPGA and the data buffers. The TX state machine reads from the transmit memory and loads the shift-register before sending the data. The RX state machine receives data from the data buffers and takes care of moving data from the shift-register into the RX memory.

Data is read from the TX memory. The first two locations are control words. The control words are stored for state-machine use. The first data word is read and loaded into the output shift-register and the CRC generator. The shift-register is enabled to shift the data out. As the bits are shifted out of the shift-register the data is encoded for Manchester compatibility. When the last data word has been loaded into the CRC and shift-register, the hardware completes the CRC processing to be prepared for the last load to the shift-register. Once the CRC has been transmitted the hardware checks to see if more data is to be sent or if this was the last packet in the message. There are several options including using a software CRC instead of the hardware generated one, adding a post-amble pattern etc. Please refer to the register bit definitions for more details.

The receive function uses a free running shift register coupled with the receive state-machine to capture the data. When the receiver detects the idle pattern followed by 4 Manchester '0's the receiver starts to capture data. The data is read in and stored into the DPR. The embedded length is used to determine where the CRC should be. The CRC is calculated as the data is received and checked against the CRC received with the packet. An error bit is set if the two do not match. Manchester errors within the packet are detected, and used to abort processing of the message. After a packet has been received the Post Amble is tested to see that it follows the proper protocol. The Manchester and Post Amble errors also set status bits.

This document is somewhat restricted as to the technical content allowed in describing the electrical interface. The document "Point-to-Point Data Bus Protocol Specification – C72-1199-069" provides a more complete description of the interface.

Address Map

BIS3_BASE	0x0000	0	base control register offset
BIS3_ID	0x0004	1	ID Register offset
BIS3_START_SET	0x0008	2	Start Register Set Bit offset
BIS3_START_CLR	0x000C	3	Start Register Clear Bit offset
BIS3_START_RDBK	0x0008	2	Start Register Read Back offset
BIS3_IO_DATA	0x0010	4	Data Register 31 - 0 with direct read-back
BIS3_IO_DIR	0x0014	5	Direction Reg 31 - 0 with direct read-back
BIS3_IO_TERMR	0x0018	6	Termination Reg 31 - 0 with direct read-back
BIS3_IO_MUX	0x001C	6	MUX Register 31 - 0 with direct read-back
BIS3_STAT_FIFO	0x0024	9	FIFO and switch status Place Holder for future use
BIS3_PLL_CMD	0x0028	10	PLL control register and read-back of PLL data
BIS3_PLL_RDBK	0x002C	11	PLL control register read-back
BIS3_SM_CNTL_0	0x0040	16	chan 0 state-machine control read-write port
BIS3_SM_CNTL_1	0x0044	17	chan 1 state-machine control read-write port
BIS3_SM_CNTL_2	0x0048	18	chan 2 state-machine control read-write port
BIS3_SM_CNTL_3	0x004C	19	chan 3 state-machine control read-write port
BIS3_SM_CNTL_4	0x0050	20	chan 4 state-machine control read-write port
BIS3_SM_CNTL_5	0x0054	21	chan 5 state-machine control read-write port
BIS3_SM_CNTL_6	0x0058	22	chan 6 state-machine control read-write port
BIS3_SM_CNTL_7	0x005C	23	chan 7 state-machine control read-write port
BIS3_SM_CNTL_8	0x0060	24	chan 8 state-machine control read-write port
BIS3_SM_CNTL_9	0x0064	25	chan 9 state-machine control read-write port
BIS3_SM_CNTL_10	0x0068	26	chan 10 state-machine control read-write port
BIS3_SM_CNTL_11	0x006C	27	chan 11 state-machine control read-write port
BIS3_SM_CNTL_12	0x0070	28	chan 12 state-machine control read-write port
BIS3_SM_CNTL_13	0x0074	29	chan 13 state-machine control read-write port
BIS3_SM_CNTL_14	0x0078	30	chan 14 state-machine control read-write port
BIS3_SM_CNTL_15	0x007C	31	chan 15 state-machine control read-write port
BIS3_SM_CNTL_16	0x0080	32	chan 16 state-machine control read-write port
BIS3_SM_CNTL_17	0x0084	33	chan 17 state-machine control read-write port
BIS3_SM_CNTL_18	0x0088	34	chan 18 state-machine control read-write port
BIS3_SM_CNTL_19	0x008C	35	chan 19 state-machine control read-write port
BIS3_IO_RDBK	0x00C0	48	External IO read register
BIS3_IO_RDBKUPR	0x00C4	49	External IO read register UPPER data bits
BIS3_IO_TERMC	0x00C8	50	Termination Control register for Capacitors - 6T20
BIS3_INT_STAT	0x00CC	51	Interrupt Status and clear register
BIS3_I2OAR	0x00D4	53	I2O Address Storage register

BIS3_SM_mem_0	0x00800	ch 0 state-machine DPR read-write port	17-10 = 0x01
BIS3_SM_mem_1	0x01000	ch 1 state-machine DPR read-write port	17-10 = 0x02
BIS3_SM_mem_2	0x01800	ch 2 state-machine DPR read-write port	17-10 = 0x03
BIS3_SM_mem_3	0x02000	ch 3 state-machine DPR read-write port	17-10 = 0x04
BIS3_SM_mem_4	0x02800	ch 4 state-machine DPR read-write port	17-10 = 0x05
BIS3_SM_mem_5	0x03000	ch 5 state-machine DPR read-write port	17-10 = 0x06
BIS3_SM_mem_6	0x03800	ch 6 state-machine DPR read-write port	17-10 = 0x07
BIS3_SM_mem_7	0x04000	ch 7 state-machine DPR read-write port	17-10 = 0x08
BIS3_SM_mem_8	0x04800	ch 8 state-machine DPR read-write port	17-10 = 0x09
BIS3_SM_mem_9	0x05000	ch 9 state-machine DPR read-write port	17-10 = 0x0A
BIS3_SM_mem_10	0x05800	ch 10 state-machine DPR read-write port	17-10 = 0x0B
BIS3_SM_mem_11	0x06000	ch 11 state-machine DPR read-write port	17-10 = 0x0C
BIS3_SM_mem_12	0x06800	ch 12 state-machine DPR read-write port	17-10 = 0x0D
BIS3_SM_mem_13	0x07000	ch 13 state-machine DPR read-write port	17-10 = 0x0E
BIS3_SM_mem_14	0x07800	ch 14 state-machine DPR read-write port	17-10 = 0x0F
BIS3_SM_mem_15	0x08000	ch 15 state-machine DPR read-write port	17-10 = 0x10
BIS3_SM_mem_16	0x08800	ch 16 state-machine DPR read-write port	17-10 = 0x11
BIS3_SM_mem_17	0x09000	ch 17 state-machine DPR read-write port	17-10 = 0x12
BIS3_SM_mem_18	0x09800	ch 18 state-machine DPR read-write port	17-10 = 0x13
BIS3_SM_mem_19	0x0a000	ch 19 state-machine DPR read-write port	17-10 = 0x14

FIGURE 4 PMC BISERIAL-6T20-HW1 INTERNAL ADDRESS MAP

The address map provided is for the local decoding performed within the PMC BISERIAL-6T20. The address definitions are the same as for the PMC-BiSerial-3T20. The references have been left using the original naming convention to help with porting from the previous card to this one. We checked operation against the original XP test suite before moving to the new Win10 and Linux models.

The addresses are all offsets from a base address, which is assigned by the system when the PCI bus is configured.

Programming

Programming PMC-BiSerial-6T20-HW1 requires the ability to read and write data from the host. The base address is determined during system configuration of the PCI bus. The base address refers to the first user address for the position in which the PMC is installed.

Depending on the software environment, it may be necessary to set-up the system software with the PMC BISERIAL-6T20 "registration" data.

In order to receive data, software is only required to enable the Rx channel and set the frequency parameters. To transmit the software will need to load the message into the appropriate channel Dual Port RAM, set the frequency and mode and enable the transmitter.

The interrupt service routine should be loaded and the interrupt mask set. The interrupt service routine can be configured to respond to the channel interrupts on an individual basis. After the interrupt is received, the data can be retrieved. An efficient loop can be implemented to fetch the data. New messages can be received even as the current one is read from the Dual-Port RAM.

The TX interrupt indicates to the software that a message has been sent and completed. If more than one interrupt is enabled, SW needs to read the status to see which source caused the interrupt. The status bits are latched, and are explicitly cleared by writing a one to the corresponding bit. It is a good idea to read the status register and write that value back to clear all the latched interrupt status bits before starting a transfer. This will insure the interrupt status values read by the interrupt service routine came from the current transfer.

Refer to the Theory of Operation section above and the Interrupts section below for more information regarding the exact sequencing and interrupt definitions.

The VendorId = 0x10EE. The CardId = 0x0045. Current FLASH revision = 0x0104 [design 1 revision 4 ported from version 3 of PMC-BiSerial-3T20-HW1]

Register Definitions

BIS3_BASE

[$00] BiSerial-6T20 Base Control Register Port read/write

Base CONTROL Register

DATA BIT	DESCRIPTION
31-5	Spare
4	Reference Clock Select
3	I2O CLR
2	I2O EN
1	Interrupt Set
0	Interrupt Enable Master

FIGURE 5 PMC BISERIAL-6T20 BASE CONTROL REGISTER BIT MAP

All bits are active high and are reset on power-up or reset command.

Interrupt Enable Master when '1' allows interrupts generated by the PMC-BiSerial-6T20-HW1 to be driven onto the carrier [INTA]. When '0' the interrupts can be individually enabled and used for status without driving the backplane. Polled operation can be performed in this mode.

Interrupt Set when '1' and the Master is enabled, forces an interrupt request. This feature is useful for testing and software development.

I2O EN when ‘1’ allows the I2O interrupts to be activated. Interrupt requests are routed to the address stored in the I2O Address Register [I2OAR]. When ‘0’ the I2O function is disabled.

I2O CLR when toggled high [‘1’] will cause the current data stored in the I2O collection register to be cleared. It is recommended that the register clear bit be used immediately before enabling I2O operation to prevent previously stored events from causing interrupts.

Reference Clock Select when ‘1’ selects the PLL clock A to be used as the transmitter reference frequency. When ‘0’ the on-board oscillator is used. The Oscillator is set to 32 MHz. A local DCM provides a 40 MHz reference for the PLL to be compatible with the 3T20 version. With a 40 MHz reference, the Hi Tx speed is 5 MHz and the Low Tx speed is 400 KHz. If the PLL is selected the ratio of reference to speed [÷8, ÷100] will be constant, and based on the new reference frequency.

BIS3_ID

[$04] BiSerial-6T20 FLASH status/Driver Status Port read only

Design Number / FLASH Revision	
DATA BIT	**DESCRIPTION**
31-16	Design/Driver ID
15-0	FLASH revision

FIGURE 6 PMC BISERIAL-6T20 INTERRUPT ENABLE REGISTER BIT MAP

The Design / Driver ID for the HW1 project is 0x0001. The FLASH ID will be updated as features are added or revisions made. See the title page for the current FLASH revision.

BIS3_START_SET

BIS3_START_RDBK

[$08] BiSerial-6T20 Start Set Control Register Port read/write

Start Set Register	
DATA BIT	**DESCRIPTION**
31-0	Channels to activate [write only]
31-0	Channels that are active [read only]

FIGURE 7 PMC BISERIAL-6T20 START SET REGISTER

To start a channel, write a '1' to the corresponding bit. To clear a channel use the Start Clear register. Read back from this port reflects the channels which are active. Please note that TX channel "start" can be cleared by the channel state-machines. Channels 19-0 correspond to bits 19-0. The upper bits should be masked when read.

BIS3_START_CLR

[$0C] BiSerial-6T20 Start Clear Control Register Port write only

Start Clear Register	
DATA BIT	**DESCRIPTION**
31-0	Clear the active Start Bits

FIGURE 8 PMC BISERIAL-6T20 START CLEAR REGISTER

Writing a ‘1’ to a channel clear bit will cause that channels Start Bit to be cleared. The Channel will complete the current operation and then abort processing. Reading from the RDBK register will show the active channels. The state-machine may be running at a significantly slower rate than the PCI bus. There may be some delay in sensing that the start abort has been set for a particular channel.

The delay can be estimated to be the period of the clock in use and 12 periods. For Tx the clock rate is 2x the data rate. For Rx the clock rate is 8x the data rate. At low speed in Tx the delay would be 12x [1/800 Khz] => 15 uS or so. These are worst case delays.

Please note that the “ready_busy” bit can be used to check when an aborted channel is ready for a new start command. Please refer to the channel control registers.

Channels 19-0 active in this design.

BIS3_IO_DATA

[$10] BiSerial-6T20 Parallel Data Output Register read/write

Parallel Data Output Register	
DATA BIT	**DESCRIPTION**
31-0	parallel output data

FIGURE 9 PMC BISERIAL-6T20 PARALLEL OUTPUT DATA BIT MAP

There are 22 potential output bits in the parallel port. The Direction, Termination, and Mux Control registers are also involved. When the direction is set to output, and the Mux control set to parallel port the bit definitions from this register are driven onto the corresponding parallel port lines. The upper 2 bits are not DC blocked by transformers and suitable for level based controls.

This port is direct read-write of the register. The IO side is read-back from the BIS3_IO_RDBK port. It is possible that the output data does not match the IO data in the case of the Direction bits being set to input or the Mux control set to state-machine.

BIS3_IO_DIR

[$14] BiSerial-6T20 Direction Port read/write

Direction Control Port	
DATA BIT	**DESCRIPTION**
31-0	Parallel Port Direction Control bits

FIGURE 10 PMC BISERIAL-6T20 DIRECTION CONTROL PORT

When set '1' the corresponding bit in the parallel port is a transmitter. When cleared '0' the corresponding bit is a receiver. The corresponding mux control bits must also be set to parallel port. 21-0 are active bits in this design. With the transformers, DC signals will be blocked for 19-0. The receivers default mode [tri-stated input] is '1'.

BIS3_IO_TERMR

[$18] BiSerial-6T20 Termination Port read/write

Direction Control Port	
DATA BIT	**DESCRIPTION**
31-0	Parallel Port Resistive Termination Control bits

FIGURE 11 PMC BISERIAL-6T20 TERMINATION CONTROL PORT

When set '1' the corresponding bit is terminated. When cleared '0' the corresponding bit is not terminated. These bits are independent of the mux control definitions. When a bit is set to be terminated; the analog switch associated with that bit is closed to create a parallel termination of approximately 100 Ω. In most systems the receiving side is terminated, and the transmitting side is not. The drivers can handle termination on both ends.

21-0 are implemented in the 6T20. 21-20 are the discretes and 19-0 are the transformer coupled ports.

BIS3_IO_MUX

[$1C] BiSerial-6T20 Mux Port read/write

Direction Control Port	
DATA BIT	**DESCRIPTION**
31-0	Parallel Port Mux Control bits

FIGURE 12 PMC BISERIAL-6T20 MUX CONTROL PORT

When set '1' the corresponding bit is set to State-Machine control. When cleared '0' the corresponding bit is set to parallel port operation. The Mux control definition along with the Data, Direction and Termination registers allows for a bit-by-bit selection of operation under software control. 21-0 are implemented in the 6T20 design. The discretes should be cleared to register control for use.

BIS3_IO_UCNTL

[$20] BiSerial-6T20 Upper Control Port read/write

Upper Bits Control Port	
DATA BIT	**DESCRIPTION**
25-24	Mux 33,32
17-16	Termination 33,32
9-8	Direction 33,32
1-0	Data 32,32

FIGURE 13 PMC BISERIAL-6T20 UPPER CONTROL PORT

Unused this version – Discretes are in the main registers

BIS3_IO_RDBK

[$C0] BiSerial-6T20 IO Read-Back Port read only

IO Read-Back Port	
DATA BIT	**DESCRIPTION**
31-0	IO Data 31-0

FIGURE 14 PMC BISERIAL-6T20 IO READBACK PORT

The IO lines can be read at any time. The value is not filtered in any way. If the transceivers are set to TX by the parallel port or state-machine then the read-back value will be the transmitted value. If the transceivers are set to receive then the port values will be those received by the transceivers from the external IO. 21-0 are implemented in the 6T20.

BIS3_IO_TERMC

[$C8] BiSerial-6T20 Capacitive Termination Port read/write

Direction Control Port	
DATA BIT	**DESCRIPTION**
31	Test Output Clock Select
30-20	Unused
19-0	Parallel Port Capacitive Termination Control bits

FIGURE 15 PMC BISERIAL-6T20 CAPACITIVE TERMINATION CONTROL PORT

Test Output Clock Select is used to test the Capacitive termination. When set '1' all 20 transformer coupled channels output a 33MHz clock. This bit is independent of mux control, the resistive termination control and the capacitive termination control. When cleared '0' the transformer coupled channels are set to either State-Machine control or parallel port operation based on the Mux control definition.

Parallel Port Capacitive Termination Control bits when set '1', the corresponding bit is terminated with capacitor. When cleared '0' the corresponding bit is not terminated. These bits are independent of the mux control definitions. When a bit is set to be terminated; the analog switch associated with that bit is closed to create a parallel termination of approximately 300pF.

19-0 Are implemented in the 6T20, which are the transformer coupled channels.

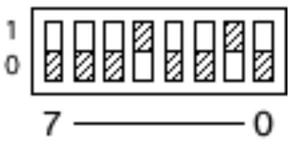
BIS3_STAT_FIFO

[$24] BiSerial-6T20 FIFO Status Port read only

FIFO Status & Switch Register	
DATA BIT	**DESCRIPTION**
23-16	sw7-0

FIGURE 16 PMC BISERIAL-6T20 FIFO STATUS

The Switch Read Port has the user bits. The user bits are connected to the eight dip-switch positions. The switches allow custom configurations to be defined by the user and for the software to identify a particular board by its switch settings and to configure it accordingly.



The Dip-switch is marked on the silk-screen with the positions of the digits and the '1' and '0' definitions. The numbers are hex coded. The example shown would produce 0x12 when read [and shifted down].

The rest of the FIFO Status port status bits are reserved for designs that require the use of external FIFO's.

BIS3_PLL_CMD, PLL_RDBK

[$28, 2C] BiSerial-6T20 PLL Control

PLL Command Register, PLL CMD Read-back

DATA BIT	DESCRIPTION
3	PLL Enable
2	PLL S2
1	PLL SCLK
0	PLL SDAT

FIGURE 17 PMC BISERIAL-6T20 PLL CONTROL

The register bits for PLL Enable, PLL S2, PLL SCLK are unidirectional from the Xilinx to the PLL – always driven. SDAT is open drain. The SDAT register bit when written low and enabled will be reflected with a low on the SDAT signal to the PLL. When SDAT is taken high or disabled the SDAT signal will be tri-stated by the Xilinx, and can be driven by the PLL. The SDAT register bit when read reflects the state of the SDAT signal between the Xilinx and PLL and can be in a different state than the written SDAT bit. To read back the contents of the CMD port use the RDBK port.

PLL enable: When this bit is set to a one, SDAT is enabled. When set to '0' SDAT is tri-stated by the Xilinx.

PLL sclk/sdata output: These signals are used to program the PLL over the I2C serial interface. SCLK is always an output whereas sdata is bi-directional.

PLL s2 output: This is an additional control line to the PLL that can be used to select additional pre-programmed frequencies.

The PLL is a separate device controlled by the Xilinx. The PLL has a fairly complex programming requirement which is simplified by using the Cypress® frequency descriptor software, and then programming the resulting control words into the PLL using the PLL Control ports. The interface can be further simplified by using the Dynamic Engineering Driver to take care of the programming requirements.

BIS3_SM_CNTL31-0

[$8C-40] BiSerial-6T20 State Machine Control Registers

State Machine Control Registers

DATA BIT	DESCRIPTION
31	Ready_Busy [read only]
30	Manchester Error Status / CLR
29-20	Address Pointer [read only]
19	Post Amble Status / CLR
18	CRC Error Status / CLR
17-8	End of Message
5	INTEN
4	CLREN
3	Bi_Uni
2	IDLE Pattern Transmit
1	HI_LOW Speed
0	TX/RX Operation

FIGURE 18 PMC BISERIAL-6T20 STATE MACHINE CONTROL REGISTERS

Each state-machine has a separate control register to govern the operation of the channel. In addition, the TX channels have control via the data in the Dual Port RAM associated with that channel.

TX/RX when set '1' indicates transmit operation. When cleared '0' indicates receiver operation.

HI_LOW when set '1' indicates operation at the "high speed" = 5 MHz nominal. In Low speed mode '0' the system operates at 400 KHz.

Bi_Uni when set '1' indicates that bidirectional operation is requested. When cleared '0' unidirectional operation is selected. In Bidirectional operation the memory is set to operate with the lower half allocated to TX and the upper half to RX. The counter will roll over to the correct boundaries [end of memory to 1/2, and 1/2 – 1 to start or end of memory to start] based on the mode of operation. Continuous operation is possible.

In BiDirectional mode the transmitter is only enabled when transmitting. When the transmission is completed the hardware automatically clears the TX bit and restarts in RX mode [BiDir only] without clearing the start bit. This allows a response, and wait for data without software intervention.

In Unidirectional mode the transmitter is enabled whenever there is data to send or the idle pattern is sent [if enabled]. The transmitter will send as many packets as the hardware is programmed to send, and at the end of the message clear the start bit. The hardware will remain in transmit or receive mode until changed by software.

IDLE when '1' and in transmit mode commands the state-machine to insert the idle pattern when not sending data from the Dual Port RAM. In receive or BiDirectional mode this bit should be set to zero.. When in transmit mode, and this bit is cleared the transmitter is tri-stated between messages sent.

CLREN when set, and in transmit mode allows the start bit to be cleared at the end of a message sent. Note that the start bit is not cleared until the last packet of the message is sent. For Rx this bit has no effect.

Please note that in BiDirectional mode this bit should be set and the packet and end of message addresses should be the same to cause a single packet to be sent and the TX bit to be reset.

INTEN when set in RX or TX mode allows the channel to create an interrupt request. The INTEN signal is applied after the holding register and before the interrupt request to the PCI bus.

In TX mode there is a control bit in the DPR command that controls the interrupt to the packet level. In RX mode the interrupt is set for each packet received. In either case the interrupt is generated by the state machine, and captured by the interrupt holding register. By reading BIS3_INT_STAT the interrupt source(s) can be checked.

If **INTEN** is not set then the Interrupt status register can be used to poll for status. With individual INTEN bits each channel can be operated in polled or interrupt driven modes. To use the interrupt method the master interrupt enable must also be enabled.

End of Message is the address to test the address pointer against for the end of message. A message is a group of packets. The packet length is embedded in the message. Please refer to the Memory section for more details on the packet length control. As each packet is sent the hardware tests the end of message address to determine if there are more packets to send.

The end of packet address includes the command word through the CRC. Starting with '0' at the lower command word, and counting n 16 bit data words plus the CRC plus 1 = the end of packet address. When the end of message address matches the end of packet address the hardware recognizes that the last packet processing should occur. The command word is 32 bits and takes 2 locations. The offset [for post increment], label, length, and CRC take 1 each for a total of 6 –1 = 5 [count from zero] plus the data length. For a message with 8 locations the end of packet address would be "D".

The **Address Pointer** is stored when an interrupt condition happens during a read. The address location stored is the address on the State-machine side of the Dual Port RAM. In receive mode an interrupt is generated each time a packet is received or a Manchester error is found. If the length is not incrementing to match the packet length received then a Manchester error has occurred cutting the packet short. The hardware will recover and look for the next packet. The address stored is the next address where data would be stored independent of LW boundaries. The next address used will be on a long word boundary. The value will be valid until the next interrupt condition occurs.

Manchester Error, CRC Error, Post Amble Error bit are set when the associated error is detected.

Manchester error is set when an illegal Manchester encoding happens when properly coded data is expected. For example if a message is programmed to be of one length and a shorter length is received the Manchester bit will be set because the data will become the idle pattern [too wide bit periods] or fixed at 0 or 1.

CRC error is set when the received CRC does not match the calculated CRC.

Post Amble error is set when the Post Amble pattern is not detected at the end of a packet.

All three bits are cleared by writing with a '1' in the respective bit positions.

Ready_Busy when set indicates that the hardware is ready for a new start command. When cleared the hardware is executing a command. For example: in tx mode with the idle pattern turned on and unidirectional mode, the hardware will be ready while sending the idle pattern. The transmitter is active, but filling time and can accept a new start command. Once start is set, the hardware will begin transmission the status will change to Busy.

BIS3_INT_STAT

[$CC] BiSerial-6T20 Interrupt Status and Clear Register

Interrupt Status and Clear Register	
DATA BIT	**DESCRIPTION**
31-0	Channel Interrupt or Clear bit

FIGURE 19 PMC BISERIAL-6T20 INTERRUPT STATUS REGISTER

Each bit is set when an interrupt occurs on the associated channel. Each bit can be cleared by writing to the register with the same bit position set '1'. You do not need to rewrite with a '0' – the clearing action happens during the write.

This register is in parallel with the I2O interrupts. Usually only one or the other will be in use at a time. Both can be used if desired. Interrupt conditions are captured and processed in both places. I20 interrupts are used with the channels. 19-0 are implemented in the 6T20 version.

BIS3_I2OAR

[$D4] BiSerial-6T20 I20 Address Register

I2O Address Register	
DATA BIT	**DESCRIPTION**
31-0	Address

FIGURE 20 PMC BISERIAL-6T20 I2O ADDRESS REGISTER

The physical address where the I2O interrupt status should be written to is stored in this register. When active interrupts are detected the I2O sequence is started. The PCI bus is requested, the hardware waits for the grant and then writes the captured status to the stored address. Please note that this is the direct hardware address, and not an indirect [translated] address.

The active bits are auto cleared and the process re-enabled for new active interrupts. Interrupts that occur during an I2O cycle are stored until the hardware is re-enabled and cause a second immediate processing cycle. The receiving hardware must be able to handle multiple interrupt status writes in close succession. A FIFO is ideal for the receiving hardware implementation.

BIS3_SM_MEM19-0

[$0x800 – 0x10000] BiSerial-6T20 Dual Port RAM address space

Each port has Dual Port RAM [DPR] associated with it. The DPR is configured to have a 32 bit port on the PCI side and a 16 bit port on the IO side. Each DPR is 1K x 16 on the IO side and 512 x 32 on the PCI side.

When using BiDirectional mode, the memory is further divided with an upper half and a lower half. The lower half is used for transmit and the upper half for receive data. The first 256 locations are used for TX and the upper 256 for receive in BiDirectional mode. In Unidirectional Mode the memory is all allocated to either RX or TX, and starts at offset 0x00.

The DPR is used to store the packet or packets of data to be transmitted or that have been received. When transmitting the data should be loaded prior to starting. It is possible to load additional data while transmitting if the software has tight control over the system timing.

In transmit the first 32 bits are the control word. The packet follows: Label, length, data, CRC. The CRC will normally be Set to 0x00 and the hardware instructed to create and insert the CRC.

Location IO	Value
0	lower command
1	upper command
2	label
3	length
4	First data
…	
N	last data
N+1	CRC or zero data

Location PCI	
0	upper lower command
1	length label
2	data 1 data 0
…	
N	CRC data last or data last data last –1 XXXX CRC

When the CRC falls on a non 32 bit boundary; the last location is padded, and the hardware will automatically skip that location to start on the new LW boundary.

The upper and lower command words are used to provide control on a packet-by-packet basis.

31	spare
30	Interrupt Enable
29	Post Amble Generation
28	CRC in Hardware
27-26	spare
25-16	spare
15-10	spare
9-0	End of Packet Address

The end of packet address and the end of message address are the word addresses on the IO side. The address is relative to the start of each DPR section – always starts at 0x00 and ends at 0x3ff for the 1K space.

The hardware will transmit until the end of packet address is detected. The hardware will then either stop or start a new packet based on the end of message address. The end of message address is not checked until the end of packet, and is checked as an absolute rather than a >= to allow roll over addressing to be used.

At the end of the packet if the Interrupt Enable bit is set an interrupt request will be generated. Each packet can have a different setting for this bit. Once at the end of message, interrupt on every packet, alternate packets etc.

At the end of the packet the hardware can append the post amble as defined in the specification. If the bit is set the post amble will be added to the packet before going to tristate, or to the idle pattern if completed, or sending the next packet if more packets are queued. If the bit is not set the post amble is ignored and the next process started earlier.

If the CRC in hardware bit is set then the CRC is generated by the hardware and appended to the message. If the bit is not set then the CRC is loaded from the location in memory immediately following the data. In either case a CRC is sent. To create a CRC error on a given packet set the bit to “memory supplied” and provide a bogus CRC value for the data. With the per packet control one packet can be made bad and the rest good to test error detection and response. Due to the processing required the Hardware option will be the “normal” option.

Due to preprocessing and hardware timing constraints the address the hardware is working with leads the data currently being sent. The following example will help to define the End of packet and end of message address as well as the CRC processing.

```
i = 0x00;
*pmcbis3SM_mem_0  = 0x7000000d; // stop after 1 pkt enable interrupt, post amble
generation, crc hardware generation, 1 packet stopping at address d
i = 0x01;
*(pmcbis3SM_mem_0 + i) = 0x00080400; // length label 0008 0400
i = 0x02;
*(pmcbis3SM_mem_0 + i) = 0x5555aaaa; // data 2 data 1
i = 0x03;
*(pmcbis3SM_mem_0 + i) = 0x0000ffff; // data 4 data 3
i = 0x04;
*(pmcbis3SM_mem_0 + i) = 0xaaaa1234; // data 6 data 5
i = 0x05;
*(pmcbis3SM_mem_0 + i) = 0x5555aaaa; // data 8 data 7
i = 0x06;
*(pmcbis3SM_mem_0 + i) = 0x00000000; // XXXX CRC
```

The message length is 8. The length is the number of 16 bit words to be sent within the packet. The length does not include the command words, label, length or CRC – data 1->data 8 are included in the length.

The end of packet address does include everything including the command word. Starting with 0 at the lower command word and counting 16 bit words address 0x0d" is one past the CRC. In this case the End of Message is also set to address 0x0D [control register for channel]; so this would be a 1 packet case.

The CRC is calculated by a slightly non-standard process. The CRC is calculated on the message not including the CRC and the command words. The hardware has a parallel CRC calculation allowing each 16 bit word to be transmitted to be added to the CRC in one clock. At the end of the message having the CRC available to transmit in 1 clock is helpful in meeting the timing requirements of the transmission.

At the start of a packet the CRC is cleared. The bitwise [1's complement] inversion of the label is then added. The length and data are added in without inversion. Then a word of 0x0000 is added. The remaining value is then transmitted immediately following the data. The hardware loads the CRC in parallel with the shift register used to transmit the data. The CRC is available one clock after the last data parallel load. The additional 0x0000 calculation is performed and then the data loaded at the appropriate transmit clock edge.

Using the example above the CRC is defined to be 0xDA8B.

The data from the shift register is then Manchester encoded and transmitted.

In the example, the CRC falls onto a non-long word boundary. The hardware will increment past the odd word and look for the next command on the next long word boundary. Please note that the memory locations are counting as long words – 0, 4, 8, C etc. The compiler will convert “i” to a long word count in the example above. The next command will be located at the 8th long word.

When receiving the data is loaded into the DPR. There is no equivalent to the command word on the receive side. When doing loop-back testing the data will be offset in the receive channel compared to the transmit channel.

In Bidirectional mode the receive data is loaded starting from the half way point of the memory. In Unidirectional mode the receiver loads data starting at 0x00. Each packet is loaded starting on a long word boundary. If a packet ends on a non-long word boundary one location is skipped.

In receive mode the receiver stays enabled until the software disables the receiver. In receive mode an interrupt request is generated on each packet received. The interrupt enable can be used to disable a channel’s interrupt request capabilities.

The Manchester decoder is designed to accommodate a range of input frequencies. The difference between the high and low speed mode is too great for the decoder to handle without selecting the proper speed of operation.

The receiver will look for a valid pre-amble before accepting data. When a valid pre-amble is found the data is captured and stored into the DPR. The CRC is calculated in hardware based on the data received and then compared against the CRC received with the message. If the CRC’s do not match then the CRC error bit is set.

The receiver restarts by looking for a new pre-amble. The idle pattern or a tri-stated bus will be detected and not stored. When a new message is detected and received it will be stored starting at the next LW location. The messages will continue to increment up the memory. Eventually roll over causing part of the message to be written to low memory or high memory [depending on mode]. As long as the data has been read by the time the roll over occurs no data will be lost.

Messages are at least 4 words long with the label, length, 1 data word plus the CRC. At 5 MHz the minimum time between messages is 12.8 uS [discounting the pre and post amble time]. An additional 3.2 uS is added for each additional word in the message. The specification allows for up to 62 words. The hardware can handle longer messages than allowed for by the specification. The length field is 16 bits and the size of the memory is 1K.

The receiver uses the length embedded in the message to determine how much data to expect and when to expect the CRC to be received. The host software can read the length and determine the end address of the packet. The host software can also read the CRC status to determine if the message is valid. It is important that the host keep up with the hardware on an interrupt basis to make sure the CRC is checked for each message before the next packet is received. Once set, the CRC stays set until explicitly cleared by the software. If the CRC is bad, and the host does not keep up then more than one message will have to be considered bad. The Manchester error and Post Amble error bits have the same restrictions. The three bits are located in the same register and can be checked in one operation.

Interrupts

PMC BISERIAL-6T20 interrupts are treated as auto-vectored. When software enters into an exception handler to deal with a PMC BISERIAL-6T20 interrupt, the status register must be read to determine the cause(s) of the interrupt, clear the interrupt request(s) and process accordingly. Power on initialization will provide a cleared interrupt request and interrupts disabled.

For example, the PMC BISERIAL-6T20 Tx state machine(s) generates an interrupt request when a transmission is complete, and the Tx int enable and Master interrupt enable bits are set. The transmission is considered complete when the last bit is output from the output shift register.

The interrupt is mapped to INTA on the PMC connector, which is mapped to a system interrupt when the PCI bus configures. The source of the interrupt is obtained by reading BIS3_INT_STAT. The status remains valid until that bit in the status register is explicitly cleared.

When an interrupt occurs, the Master interrupt enable should be cleared, and the status register read to determine the cause of the interrupt. Next perform any processing needed to remove the interrupting condition, clear the latched bit and set the Master interrupt enable bit high again.

The individual enables operate after the interrupt holding latches, which store the interrupt conditions for the CPU. This allows for operating in polled mode simply by monitoring the BIS3_INT_STAT register. If one of the enabled conditions occurs, the interrupt status bit will be set, but unless the Master interrupt, and the channel interrupt enable is set, a system interrupt will not occur.

I2O interrupts are also available. Program the Address where the interrupt status should be written to in the I2OAR, Clear any stored interrupts in the I2O register, and program the I2O enable to be set. The hardware will collect interrupt conditions, and write them to the address stored in the I2OAR. The interrupts will still need to be processed at the hardware level.

Loop-back

The reference software includes an external loop-back test. The HW1 version of PMC-BiSerial-6T20 utilizes a 68 pin SCSI II front panel connector. The test requires an external cable with the following pins connected.

SIGNALs	+	-	+	-
Ch 0,1	1	35	2	36
Ch 2,3	3	37	4	38
Ch 4,5	9	43	10	44
Ch 6,7	11	45	12	46
Ch 8,9	15	49	16	50
Ch 10,11	19	53	20	54
Ch 12,13	21	55	22	56
Ch 14,15	25	59	26	60
Ch 16,17	29	63	30	64
Ch 18,19	31	65	32	66
Additional IO used for parallel Port				
Ch 20, 21	33	67	34	68

PMC PCI Pn1 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn1 Interface on PMC-BISERIAL-6T20. See the User Manual for your carrier board for more information. Unused pins may be assigned by the specification and not needed by this design.

	-12V[unused]	1	2
GND	INTA#	3	4
		5	6
BUSMODE1#	+5V	7	8
		9	10
GND -		11	12
CLK	GND	13	14
GND -		15	16
	+5V	17	18
	AD31	19	20
AD28-	AD27	21	22
AD25-	GND	23	24
GND -	C/BE3#	25	26
AD22-	AD21	27	28
AD19	+5V	29	30
	AD17	31	32
FRAME#-	GND	33	34
GND	IRDY#	35	36
DEVSEL#	+5V	37	38
GND	LOCK#	39	40
		41	42
PAR	GND	43	44
	AD15	45	46
AD12-	AD11	47	48
AD9-	+5V	49	50
GND -	C/BE0#	51	52
AD6-	AD5	53	54
AD4	GND	55	56
	AD3	57	58
AD2-	AD1	59	60
	+5V	61	62
GND		63	64

FIGURE 21 PMC BISERIAL-6T20 PN1 INTERFACE

PMC PCI Pn2 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn2 Interface on PMC-BISERIAL-6T20. See the User Manual for your carrier board for more information. Unused pins may be assigned by the specification and not needed by this design.

+12V[unused]		1	2
		3	4
	GND	5	6
GND		7	8
		9	10
		11	12
RST#	BUSMODE3#	13	14
	BUSMODE4#	15	16
	GND	17	18
AD30	AD29	19	20
GND	AD26	21	22
AD24		23	24
IDSEL	AD23	25	26
	AD20	27	28
AD18		29	30
AD16	C/BE2#	31	32
GND		33	34
TRDY#		35	36
GND	STOP#	37	38
PERR#	GND	39	40
	SERR#	41	42
C/BE1#GND		43	44
AD14	AD13	45	46
GND	AD10	47	48
AD8		49	50
AD7		51	52
		53	54
	GND	55	56
		57	58
GND		59	60
		61	62
GND		63	64

FIGURE 22 PMC BISERIAL-6T20 PN2 INTERFACE

PMC-BiSerial-6T20 Front Panel IO Pin Assignment

The figure below gives the pin assignments for the PMC Module IO Interface on PMC-BISERIAL-6T20. Also, see the User Manual for your carrier board for more information. For customized version, or other options, contact Dynamic Engineering.

IO_0p	IO_0m	1	35
IO_1p	IO_1m	2	36
IO_2p	IO_2m	3	37
IO_3p	IO_3m	4	38
IO_4p	IO_4m	9	43
IO_5p	IO_5m	10	44
IO_6p	IO_6m	11	45
IO_7p	IO_7m	12	46
IO_8p	IO_8m	15	49
IO_9p	IO_9m	16	50
IO_10p	IO_10m	19	53
IO_11p	IO_11m	20	54
IO_12p	IO_12m	21	55
IO_13p	IO_13m	22	56
IO_14p	IO_14m	25	59
IO_15p	IO_15m	26	60
IO_16p	IO_16m	29	63
IO_17p	IO_17m	30	64
IO_18p	IO_18m	31	65
IO_19p	IO_19m	32	66
IO_20p	IO_20m	33	67
IO_21p	IO_21m	34	68

FIGURE 23 PMC BISERIAL-6T20 FRONT PANEL INTERFACE

Channels 19-0 are transformer coupled. Channels 21-20 are direct coupled.

Applications Guide

Interfacing

The pin-out tables are displayed with the pins in the same relative order as the actual connectors. The pin definitions are defined with noise immunity in mind. The pairs are chosen to match standard SCSI II/III cable pairing to allow a low-cost commercial cable to be used for the interface.

Some general interfacing guidelines are presented below. Do not hesitate to contact the factory if you need more assistance.

Watch the system grounds. All electrically connected equipment should have a fail-safe common ground that is large enough to handle all current loads without affecting noise immunity. Power supplies and power-consuming loads should all have their own ground wires back to a common point.

Power all system power supplies from one switch. Connecting external voltage to the PMC-BISERIAL-6T20 when it is not powered can damage it, as well as the rest of the host system. This problem may be avoided by turning all power supplies on and off at the same time. With the transformer coupling and fairly wide differential voltage maximum ratings this is usually not an issue. It can become an issue with long cables runs from equipment not sharing a common reference.

PMC-BiSerial-6T20 does incorporate diode protection on the switches to prevent reverse power into the board when the board is not energized and the system is. TVS devices are installed on the discrete ports which are not transformer protected. The transformers do offer protection against DC offsets when powered on or unpowered.

Keep cables short. Flat cables, even with alternate ground lines, are not suitable for long distances. PMC-BiSerial-6T20 does not contain special input protection. The connector is pinned out for a standard SCSI II/III cable to be used. The twisted pairs are defined to match up with the PMC-BiSerial-VI pin definitions. It is suggested that this or similar cable be used for most of the cable run.

Terminal Block. We offer a high quality 68-screw terminal block that directly connects to the SCSI II/III cable. The terminal block can mount on standard DIN rails. HDEterm68 [http://www.dyneng.com/HDEterm68.html]

We provide the components. You provide the system. Safety and reliability can be achieved only by careful planning and practice. Inputs can be damaged by static discharge, or by applying voltage outside of the RS-485 devices rated voltages.

Construction and Reliability

PMC Modules are conceived and engineered for rugged industrial environments. The PMC-BISERIAL-6T20 is constructed out of 0.062 inch thick High Temp FR4 material. The PC Boards are ROHS compliant. Dynamic Engineering has selected gold immersion processing to provide superior performance, and reliability. [to avoid tin whisker issues].

Standard processing is “leaded”. Add –ROHS to the part number if ROHS processing and components is needed.

The PMC connectors are rated at 1 Amp per pin, 100 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable.

The PMC is secured against the carrier with four screws attached to the 2 stand-offs and 2 locations on the front panel. The four screws provide significant protection against shock, vibration, and incomplete insertion.

The PMC Module provides a low temperature coefficient of 2.17 W/^{O}C for uniform heat. This is based upon the temperature coefficient of the base FR4 material of 0.31 W/m-^{O}C, and taking into account the thickness and area of the PMC. The coefficient means that if 2.17 Watts are applied uniformly on the component side, then the temperature difference between the component side and solder side is one degree Celsius.

Thermal Considerations

The PMC-BiSerial-6T20 design consists of CMOS circuits. The power dissipation due to internal circuitry is very low. It is possible to create a higher power dissipation with the externally connected logic. If more than one Watt is required to be dissipated due to external loading then forced air cooling is recommended. With the one-degree differential temperature to the solder side of the board external cooling is easily accomplished.

Warranty and Repair

Please refer to the warranty page on our website for the current warranty offered and options.

http://www.dyneng.com/warranty.html

Service Policy

Before returning a product for repair, verify as well as possible that the suspected unit is at fault. Then call the Customer Service Department for a RETURN MATERIAL AUTHORIZATION (RMA) number. Carefully package the unit, in the original shipping carton if this is available, and ship prepaid and insured with the RMA number clearly written on the outside of the package. Include a return address and the telephone number of a technical contact. For out-of-warranty repairs, a purchase order for repair charges must accompany the return. Dynamic Engineering will not be responsible for damages due to improper packaging of returned items. For service on Dynamic Engineering Products not purchased directly from Dynamic Engineering, contact your reseller. Products returned to Dynamic Engineering for repair by other than the original customer will be treated as out-of-warranty.

Out of Warranty Repairs

Out of warranty repairs will be billed on a material and labor basis. Customer approval will be obtained before repairing any item if the repair charges will exceed one half of the quantity one list price for that unit. Return transportation and insurance will be billed as part of the repair and is in addition to the minimum charge.

For Service Contact:

Customer Service Department
Dynamic Engineering
150 DuBois Street, Suite C
Santa Cruz, CA 95060
831-457-8891
support@dyneng.com

Specifications

Host Interface:	[PMC] PCI Mezzanine Card - 32 bit, 33 MHz
IO Interface:	20 manchester encoded transformer coupled serial interfaces. 16 bit word size, MSB first, multiple words, CRC, embedded length, label. 2 direct connect register bits are provided.
Tx Data rates generated:	32 MHz oscillator used to generate 80 MHz. 800 KHz, 3.2 MHz, 10 MHz and 40 MHz are generated to provide TX and RX reference rates. 5 MHz and 400 KHz IO frequencies supported via software selection. PLL for custom transmit frequencies supplied.
Tx Options	Tristate or transmit IDLE pattern between messages, append post amble pattern, calculate and append CRC in hardware, interrupt on a packet basis.
Rx Data rates accepted:	Continuous at 5 MHz. or 400 KHz software programmable
Software Interface:	Control Registers, Status Ports, Dual-Port RAM, Driver Available
Initialization:	Hardware Reset forces all registers to 0.
Access Modes:	LW boundary Space (see memory map)
Wait States:	1 for all addresses
Interrupt:	Tx and Rx interrupts at end of packet transmission/reception Software interrupt I2O interrupts
DMA:	No DMA Support implemented at this time
Onboard Options:	All Options are Software Programmable
Interface Options:	68 pin twisted pair cable 68 screw terminal block interface
Dimensions:	Standard Single PMC Module.
Construction:	FR4 Multi-Layer Printed Circuit, Through Hole and Surface Mount Components.
Temperature Coefficient:	2.17 W/^{O}C for uniform heat across PMC
Power:	Max. **TBD** mA @ 5V
Temperature range	[-40 + 85] Industrial Temperature standard

Order Information

PMC-BiSerial-6T20-HW1	PMC Module with 20 Manchester encoded serial ports, plus 2 bit parallel port. RS-485 IO. 32-bit data interface
-ROHS	Add RoHS processing
-CC	Add conformal coating
HDEterm68	68 pin break out compatible with SCSI connector on PMC-BiSerial-6T20. https://www.dyneng.com/HDEterm68.html
HDEcabl68	SCSI connector available in various lengths https://www.dyneng.com/HDEcabl68.html