

# **DYNAMIC ENGINEERING**

435 Park Dr., Ben Lomond, Calif. 95005  
831-336-8891 Fax 831-336-3840  
sales@dyneng.com  
www.dyneng.com  
Est. 1988

## **User Manual**

# **PMC-BiSerial**

## **Bi-directional Serial Data Interface PMC Module**

Revision A2  
Corresponding Hardware: Revision A

**PMC-BiSerial**  
Bi-directional Serial Data Interface  
PMC Module

Dynamic Engineering  
435 Park Drive  
Ben Lomond, CA 95005  
831- 336-8891  
831-336-3840 FAX

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 PMC Module carriers and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage.

©2000 by Dynamic Engineering.

Trademarks and registered trademarks are owned by their respective manufactures.  
Manual Revision A2. Revised August 27, 2000.



---

---

# Table of Contents

---

---

<b>PRODUCT DESCRIPTION</b>	<b>6</b>
<b>THEORY OF OPERATION</b>	<b>9</b>
<b>ADDRESS MAP</b>	<b>11</b>
<b>PROGRAMMING</b>	<b>12</b>
<b>Register Definitions</b>	<b>14</b>
bis_base	14
bis_tx	16
bis_txs	16
bis_rx	17
bis_rxs	17
bis_stat0	18
bis_stat1	19
bis_intc	19
bis_fifotx	19
bis_fiforx	19
bis_dir_term	20
bis_switch	21
<b>Interrupts</b>	<b>22</b>
<b>Loop-back</b>	<b>23</b>
<b>PMC PCI PN1 INTERFACE PIN ASSIGNMENT</b>	<b>24</b>
<b>PMC PCI PN2 INTERFACE PIN ASSIGNMENT</b>	<b>25</b>
<b>BISERIAL FRONT PANEL IO PIN ASSIGNMENT</b>	<b>26</b>
<b>PMC PN4 USER INTERFACE PIN ASSIGNMENT</b>	<b>27</b>
<b>APPLICATIONS GUIDE</b>	<b>28</b>
<b>Interfacing</b>	<b>28</b>



Construction and Reliability	29
Thermal Considerations	29
<b>WARRANTY AND REPAIR</b>	<b>30</b>
Service Policy	31
Out of Warranty Repairs	31
For Service Contact:	31
<b>SPECIFICATIONS</b>	<b>32</b>
<b>ORDER INFORMATION</b>	<b>33</b>
<b>SCHEMATICS</b>	<b>33</b>



---

---

# List of Figures

---

---

FIGURE 1	PMC-BISERIAL BLOCK DIAGRAM	6
FIGURE 2	PMC-BISERIAL STANDARD SERIAL PROTOCOL TIMING	9
FIGURE 3	PMC-BISERIAL INTERNAL ADDRESS MAP	11
FIGURE 4	PMC-BISERIAL BASE CONTROL REGISTER BIT MAP	14
FIGURE 5	PMC-BISERIAL TX CONTROL REGISTER BIT MAP	16
FIGURE 6	PMC-BISERIAL TX SPECIAL CONTROL REGISTER BIT MAP	16
FIGURE 7	PMC-BISERIAL RX CONTROL REGISTER BIT MAP	17
FIGURE 8	PMC-BISERIAL TX SPECIAL CONTROL REGISTER BIT MAP	17
FIGURE 9	PMC-BISERIAL STATUS REG 0 BIT MAP	18
FIGURE 10	PMC-BISERIAL STATUS REG 1 BIT MAP	19
FIGURE 11	PMC-BISERIAL DIRECTION TERMINATION CONTROL BIT MAP	20
FIGURE 12	PMC-BISERIAL SWITCH READ BIT MAP	21
FIGURE 13	PMC-BISERIAL PN1 INTERFACE	24
FIGURE 14	PMC-BISERIAL PN2 INTERFACE	25
FIGURE 15	PMC-BISERIAL FRONT PANEL INTERFACE	26
FIGURE 16	PMC-BISERIAL PN4 INTERFACE	27

## Product Description

PMC-BiSerial is part of the PMC Module family of modular I/O components by Dynamic Engineering. The PMC-BiSerial is capable of providing multiple serial protocols. The standard protocol implemented provides Data, Clock and Strobe interface.

Custom interfaces are available. We will redesign the state machines and create a custom interface protocol. That protocol will then be offered as a “standard” special order product. Please see our web page for current protocols offered. Please contact Dynamic Engineering with your custom application.

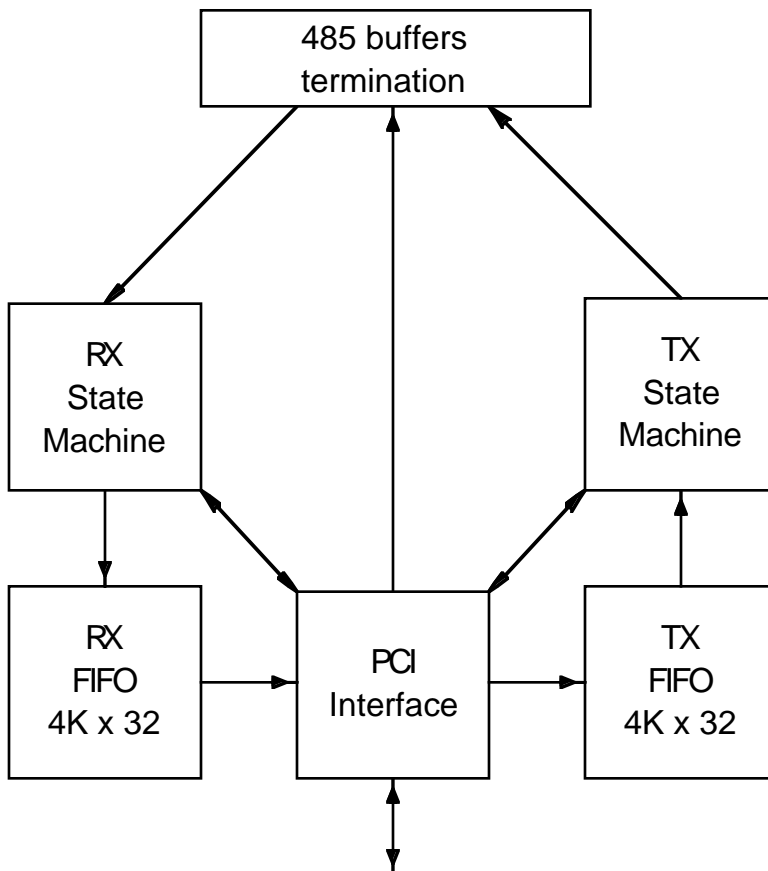
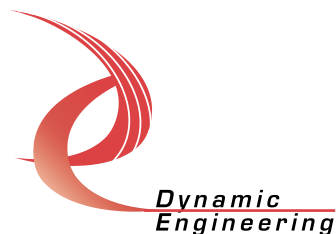


FIGURE 1

PMC-BISERIAL BLOCK DIAGRAM

Several clocking options are available with the standard -IO version. The PMC Clock, 10 MHz. oscillator and external reference input are the clock sources. The source can be selected with software along with the divisor.



A 12 bit counter is provided with a programmable divisor to offer a multitude of frequency options based on the three standard references. If a specific frequency is required that is not attainable with the standard choices we can install a "user" oscillator. Please be sure to select the proper source and clock divisors after reset to insure proper operation. Please refer to the programming section for details.

Differential I/O is available on 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  $130\Omega$ . The resistors are in discrete 1206 packages to allow individual termination options for custom formats and protocols. There are 20 transceivers for the IO. The transceivers are programmable to allow more outputs or more inputs as needed for a specific protocol implementation. The standard configuration is Data, Clock, and Strobe for Receive and Transmit plus a reference clock input. The transceivers are programmed through the Xilinx device for maximum flexibility. The terminations are also programmable to be active or not.

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

The PMC-BISERIAL 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 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. If your carrier has non-standard connectors [height] to mate with the PMC-BiSerial, please let us know. We may be able to do a special build with a different height connector to compensate.

The serial channels are each supported by a 4K by 32 bit FIFO. The FIFO supports long word reads and writes. A path exists for loop-back testing of each FIFO. The PMC data path is 32 bits wide. The Data path from the FIFO to the Xilinx is 8 bits wide. The hardware automatically performs the 4 data accesses and byte lane manipulation to make the internal port appear as a 32 bit port to the PMC bus. The design is optimized for the system configuration with minimal delay on the PCI write to TX FIFO path and PCI read from the RX FIFO path. The added delay for reading and writing to the internal FIFO ports only affects the loop-back path.



The serial receive channel receives data in 32 bit words which can come continuously. As each 32 bit word is received the data is stored into the Receive FIFO. The host can poll, wait for the message complete interrupt, or use the programmable FIFO Almost Full flag. The message can be read directly from the input FIFO.

The Output channel has a separate 4K x 32 FIFO. The FIFO can be written as long words. Data is sent MSB first. Data is sent when the transmitter is enabled and data is stored within the FIFO. Transmission completes when the FIFO is detected to be empty. The Strobe signals that data is active on the serial data output. The strobe is active low. The strobe and data switch on the falling edge and are valid on the rising edge of the TX clock.

Smaller and larger FIFOs are available as a special order.

Interrupts are supported by the PMC-BISERIAL. The interrupt occurs at the end of the message whether data is received or transmitted or both. The interrupts are individually maskable. The interrupt occurs on INTA. The FIFO status is available for the FIFO making it possible to operate in a polled mode. In addition to the protocol interrupts there are interrupts associated with the programmable FIFO flags. The Programmable Almost Full flag is used with the receive channel to allow user software to read data from the FIFO on an interrupt basis with long messages [ longer than the 16K of FIFO]. The Programmable Almost Empty flag is used with the transmitter to allow software to operate in an interrupt driven mode and to keep the TX FIFO from going empty.



## Theory of Operation

The PMC-BISERIAL is designed for transferring data from one point to another with a serial protocol.

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

The PMC-BISERIAL is a part of the PMC Module family of modular I/O products. It meets the PMC and CMC draft Standards. 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. In standard configuration, the PMC-BiSerial is a Type 1 mechanical with no components on the back of the board and one slot wide, with 10 mm inter-board height.

The PCI interface to the host CPU is controlled by a logic block within the Xilinx. The BISERIAL design requires one wait state for read or write cycles to any address other than the loop-back ports which require eight. The PMC-BISERIAL is capable of supporting 40 MBytes per second into and out of the FIFOs. The wait states refer to the number of clocks after the PCI core decode before the "terminate with data" state is reached. Two additional clock periods account for the 1 clock delay to decode the signals from the PCI bus and to convert the terminate with data state into the TRDY signal.

The serial I/O can support many protocols. The standard timing is shown in the next diagram. The clock is free running, the data is valid on the rising edge of the clock, and strobe frames the data.

A pair of state machines within the FPGA control all transfers between the FIFO and FPGA, and the FPGA and the data buffers. The TX state machine reads from the transmit FIFOs and loads the shift registers 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 FIFO.

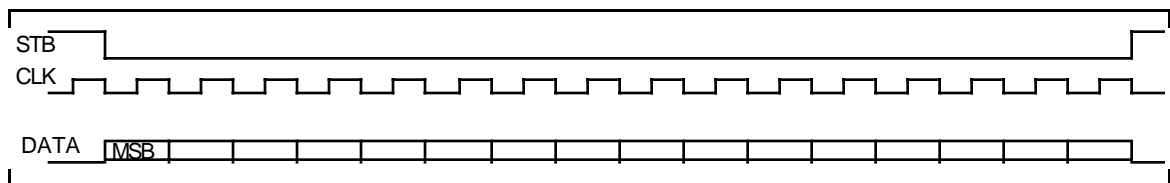


FIGURE 2

PMC-BISERIAL STANDARD SERIAL PROTOCOL TIMING



Data is read from the TX FIFO and loaded into the shift register. The MSB is then present at the output of the data buffer. The Strobe is activated at the same time. One half clock period later the rising edge of the data clock is driven to the output clock buffer. One half clock period later, the data is transitioned to the next value. The MSB-1 is now on the data lines. The process repeats until the first word is transferred. If the FIFOs are not empty from reading the first word then the process repeats for the second word. In the standard timing there are no inter-word gaps. The data stream is continuous from MSB to LSB for a compact serial transfer. Transfer rates up to 12 MHz are possible with the BiSerial design.

Clock reference sources include an on-board 10 MHz oscillator, the PCI clock and a user input clock. The reference clock is available at the base rate and after a divider. The divider is a 12 bit counter with programmable divisor. Please refer to the memory map for more details.

The receive function is very similar. When the Strobe is detected low then data is loaded into the receive shift register on the rising edge of the data clock. Once a word has been received the data is loaded into the receive FIFO. When the strobe goes inactive the transfer has been completed and an interrupt is generated to the host [if enabled]. In normal operation, once the receive state machine is enabled, it stays enabled until the strobe signals the end of the message. The enable is then auto-cleared.



## Address Map

Function	Offset	function	Type
bis_base	EQU \$00	base control register	read/write
bis_tx	EQU \$04	transmit base control	read/write
bis_txs	EQU \$08	transmit special fnct	read/write
bis_rx	EQU \$0c	receive base control	read/write
bis_rxs	EQU \$10	receive special fnct	read/write
bis_STAT0	EQU \$14	status register 0	read
bis_STAT1	EQU \$18	status register 1	read
bis_intc	EQU \$1C	interrupt clear	write
bis_fifotx	EQU \$20	transmit FIFO access	read/write
bis_fiforx	EQU \$24	receive FIFO access	read/write
bis_dir_term	EQU \$28	direction & termination	read/write
bis_switch	EQU \$44	read user switch	read

FIGURE 3

PMC-BISERIAL INTERNAL ADDRESS MAP

The address map provided is for the local decoding performed within the PMC-BiSerial. The addresses are all offsets from a base address. The carrier board that the PMC is installed into provides the base address.

The VendorId = 0x10EE. The CardId = 0x0005. Current revision = 0x02



## Programming

Programming the PMC-BISERIAL requires only the ability to read and write data from the host. The base address is determined by the PMC Carrier board. This documentation refers to the first user address for the slot that the PMC is installed in as the base address.

Depending on the software environment it may be necessary to set-up the system software with the PMC-BiSerial "registration" data. For example in WindowsNT there is a system registry which is used to identify the resident hardware.

In order to receive data the software is only required to enable the RX state machine and FIFOs. If desired, the interrupt can be enabled and parity selected. Data will be loaded into the FIFOs as it is received.

The interrupt service routine should be loaded and the mask should be set. After the interrupt is received, the data pattern can be retrieved. If the data is of fixed length, then the data can be immediately read. If the message length is variable then the receive count should be accessed to determine the number of words to read. An efficient loop can then be utilized to fetch the data. New messages can be received even as the current one is read from the FIFO. It is important to read the count for the current message length prior to the completion of the next message. The read word count is stored in a register, which is over-written when the next message completes.

The end of transmission interrupt will indicate to the software that the message has been started and that the message has terminated. If both the TX and RX interrupts are enabled then the SW needs to read BIS\_STAT0 to see which source caused the interrupt. Reading BIS\_STAT0 will clear the interrupt status and accessing BIS\_INTC will clear the actual interrupt. The interrupt status can be read after the BIS\_INTC cycle. It is a good idea to read the status register to force the RX\_INT and TX\_INT bits to zero before Start is enabled. This will insure that the RX\_INT or TX\_INT=1 value read by the interrupt service routine came from the current reception.

Before transmitting data, the FIFOs are enabled and the data loaded. If the clock rate desired is something other than the default rate then the rate should be selected. Be sure to set the clock source and rate bits appropriately. Once the complete message is loaded and the controls set properly the start bit can be set to cause the transfer to begin. If a slow



clock rate is selected and a long message is sent then data can be loaded during transmission to save operational time. Care must be taken to insure that the FIFOs do not become empty. When the TX interrupt is received the transmission has been completed and another message can be loaded. All that needs to happen with a second message is to load the FIFO and set the start bit.

Messages longer than 16K bytes can be accommodated by special ordering HW with larger FIFOs, by polling or using the programmable flag interrupts. To poll read the Status 0 register during the transfer and take appropriate action when the full, empty or programmable flag shows that there is data to read or space to write. The PAE and PAF flags are implemented to provide an almost empty interrupt to allow the TX side to operate in an interrupt driven mode with longer messages. Similarly the PAF can be used to provide an almost full interrupt for the receive side- to allow interrupt driven long message capability.

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



## Register Definitions

### bis\_base

\$00 BISERIAL Control Register Port read/write

CONTROL BASE	
DATA BIT	DESCRIPTION
31-23	spare
22	FRX_LD
21	FTX_LD
20	FIFO_EN
19-18	spare
17	INT_SET
16	INT_EN_MASTER
15	spare
14-13	clock pre-selector
12	clock post-selector
11-0	clock divisor

FIGURE 4

PMC-BISERIAL BASE CONTROL REGISTER BIT MAP

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

FRX\_LD is tied to the RX FIFO WE2/\_LD pin. FTX\_LD is tied to the TX FIFO WE2/\_LD pin. When the FIFOs are taken out of reset it is possible to set-up the FIFO to accept commands to program the way the programmable almost empty and programmable almost full signals operate. ***In the standard transfer mode these pins are set hi before FIFO\_EN is set to use as a second WE control pin.*** If the PAE and PAF flags are used then the FIFOs will require programming.

FIFO\_EN when '1' takes the FIFO out of reset. To create a reset be sure to leave in the '0' state for the reference clock to capture the reset. This can be an issue if a slow transmission rate is chosen. To guarantee reset the PCI clock can be used as a reference for both the TX and RX FIFOs temporarily then set back to the original settings.

INT\_SET is used for test and software development purposes to create an interrupt request. 1 = assert interrupt request. 0 = normal operation. Useful to stimulate interrupt acknowledge routines for development.

INT\_EN\_MASTER when '1' gates all interrupts through to the PCI host.



When '0' the interrupts can be used for status without interrupting the host.

#### Clock Pre-Selector

00	oscillator
01	oscillator
10	external
11	pci clock

The clock pre-selector is used to select which reference clock to use with the divisor hardware. [The clock source] The external clock is IO channel 0.

Divisor [11-0] are the clock divisor select bits. The clock source is divided by a counter and the select bits pick which clock is used to drive the IO read-back registers. The reference clock for the counter is selected with the CLK Pre-Selector. The output frequency is  $\{\text{reference} / [2^{(n+1)}]\}$ .  $N \geq 1$ . The reference oscillator is 10 MHz. in frequency. The counter divides by N+1 due to counting from 0 -> n before rolling over. The output is then divided by 2 to produce a square wave output.

Post Selector when '1' sets clock out to clock divided, when '0' sets clock out to pre-selector reference value.

Please note that the 485 buffers are rated for 12 MHz. With most systems the larger divisors will be used. The smaller divisors are provided for use with external oscillators and the external clock line.



## bis\_tx

\$04 BISERIAL TX Control Register Port read/write

CONTROL TX	
DATA BIT	DESCRIPTION
31-4	spare
3	clock_en
2	int_en_pae
1	int_en_tx
0	start_tx

FIGURE 5

PMC-BISERIAL TX CONTROL REGISTER BIT MAP

Clock\_en when '1' gates the transmit clock to be driven to TX\_CLK. The gate is provided to allow FIFO loop back testing without driving the clock onto the transmission line. When '0' the TX\_CLK is held in at '0'. The enable should normally be set.

INT\_EN\_TX is the Interrupt Enable bit for the Transmit channel. The default state is off. If enabled and the master interrupt enable is also enabled then an interrupt is requested when the transmission is complete. The interrupt is cleared by writing to bis\_intc.

INT\_EN\_PAE = 1 to enable the FIFO Programmable Almost Empty interrupt. When enabled an interrupt is generated when the data level falls to the programmed level. The interrupt is cleared by reading the status register [bis\_stat0]

Start\_tx when set will start a transmission assuming that there is data in the TX FIFO. Start\_TX is auto-cleared when the transmission is complete

## bis\_txs

\$08 BISERIAL TX Special Control Register Port read/write

CONTROL TX Special	
DATA BIT	DESCRIPTION
31-0	spare

FIGURE 6

PMC-BISERIAL TX SPECIAL CONTROL REGISTER BIT MAP



## bis\_rx

\$0C BISERIAL RX Control Register Port read/write

CONTROL RX	
DATA BIT	DESCRIPTION
31-4	spare
3	testmode
2	int_en_paf
1	int_en_rx
0	start_rx

FIGURE 7

PMC-BISERIAL RX CONTROL REGISTER BIT MAP

Testmode when '1' selects the PCI clock for the RX FIFO reference instead of the RX\_CLK. Normally set to '0'. Set to '1' for FIFO loop-back.

Int\_en\_paf when '1' enables the RX FIFO Programmable Almost Full flag interrupt. The interrupt becomes active when the data in the RX FIFO reaches a user programmed point of almost full. Requires the master enable to create a system level interrupt.

Int\_en\_rx is used to enable the receive interrupt. The default is disabled. When '1' and the master interrupt enable is also enabled then an interrupt is requested when the Strobe returns to the off state [1]. The interrupt is cleared by writing to bis\_intc.

Start\_rx is used to enable the receive state machine to receive messages. Cleared at the end of each message.

## bis\_rxs

\$10 BISERIAL RX Special Control Register Port read/write

CONTROL RX Special	
DATA BIT	DESCRIPTION
31-0	spare

FIGURE 8

PMC-BISERIAL TX SPECIAL CONTROL REGISTER BIT MAP



## bis\_stat0

[\$14] BISERIAL Status Port [read only]

Data Bit	Status	
15	int_request	1 = interrupt request after int_en mask
14-13	gnd	set to 0
12	fae_int	1 = programmable almost empty int rqst
11	tx_int	1 = tx int rqst
10	ftx_ffn	0 = tx fifo full 1 = not full
9	ftx_faen	0 = almost empty 1 = not almost empty
8	ftx_mtn	0 = tx fifo empty, 1 = not empty
7	unused	
6-5	gnd	set to 0
4	faf_int	1 = fifo almost full interrupt asserted
3	rx_int	1 = rx interrupt asserted
2	frx_ffn	0 = rx fifo full, 1 = not full
1	frx_fafn	0 = almost full 1 = not almost full
0	frx_mtn	0 = rx fifo empty 1 = not empty

FIGURE 9

PMC-BISERIAL STATUS REG 0 BIT MAP

The FIFO flags are active low. When the empty bit is low then the FIFO is empty. When the empty flag is high then the FIFO has at least one piece of data stored. When the Full Flag is set [low] the FIFO is full. When not set then the FIFO still has room.

When int\_request is set then the at least one of the maskable interrupts is active and the master enable is set to allow interrupts.

Tx\_int when set indicates that the transmission has completed. Rx\_INT when set indicates that a message has been received. To clear, write to bis\_intc with any pattern. The status bit is a separate registered version of the interrupt request. The status bit is cleared when the status register is read. If the interrupt request has not been cleared then the status will become set again because the request is still present. Once the bis\_intc has been written to the interrupt will clear and the status read will clear and stay cleared. An alternate method of clearing the tx\_int is to dis-able the interrupt enable for TX. Similarly the rx\_int\_en can be used to clear the pending interrupt by disabling the interrupt.

## **bis\_stat1**

[\$18] BISERIAL Status Port [read only]

Data Bit	Status
15-0	received word count

FIGURE 10

PMC-BISERIAL STATUS REG 1 BIT MAP

WORD\_CNT 15..0 is the word count for the number of words received for a particular reception. The count is updated at the end of the reception. The count is valid once the RX interrupt bit is set for a reception. Each "count" = one 32 bit word stored in memory. The count remains valid until the next message completes.

## **bis\_intc**

[\$1c] BISERIAL Interrupt Clear Port

The user can, by accessing this port, cause the BISERIAL to clear the pending interrupt requests. Any data pattern can be written.

## **bis\_fifotx**

[\$20] BISERIAL TX FIFO write-read port

The BISERIAL supports 32 bit writes to the transmit data FIFOs. Data is aligned D31-0. Normally this port is only written to. For loop-back testing the contents of the FIFO can be read from the "Xilinx" side of the FIFO. The reference clock must be set to the PCI source for the loop-back to work. The engineering kit contains software which performs a TX FIFO loop-back. Once data is read from the FIFO it is no longer available for transmission.

## **bis\_fiforx**

[\$24] BISERIAL RX FIFO write-read port

The BISERIAL supports 32 bit reads from the receive data FIFOs. Data is aligned D31-0. Normally this port is only read from. For loop-back testing the contents of the FIFO can be written through the "Xilinx" side of the FIFO. The testmode bit must be set to cause the PCI clock to be used as the clock reference on the RX FIFO Write port. The engineering kit contains software which performs an RX FIFO loop-back. Once data is read from the FIFO it is no longer available.



## bis\_dir\_term

[\$28 BiSerial direction and termination Port read/write]

CONTROL REGISTER DIR_TERM	
DATA BIT	DESCRIPTION
6-0	DIRection 10-0 0 = read, 1 = drive
23-16	TERMination 10-0 1 = terminated

FIGURE 11

PMC-BISERIAL DIRECTION TERMINATION CONTROL BIT MAP

The direction for each of the 20 differential pairs is controlled through this port. The port defaults to '0' which corresponds to tri-stating the drivers. The output and input pins are separated and independently connected to the Xilinx to allow loop-back testing.

Pull-up and Pull-down resistors built into some '485 interface devices may make the signal appear to be driven [if open] when in the tri-stated mode. Enabling the termination on a tristated line will yield approximately 2.5V on each side of the tri-stated driver.

The base design of the PMC\_BiSerial\_IO sets the direction of the signals to be set to output except for the signal group 12-15 which are inputs and IO0 which is also an input. The table is shown for future use when we plan to add in a read/write parallel port on the unused transmission lines. Currently the forced bits are read-write but have no effect.

CONTROL	CORRESPONDING IO BIT(S)
DIR_0..3	IO_0..3.
DIR4	IO_4..7
DIR5	IO_8..11
DIR6	IO_12..15
DIR7	IO_16..19

Parallel termination resistors are supplied on each differential pair along with a switch to allow the user to select which lines are terminated and where. In some systems it will make sense to terminate the lines in the cable and in others it will make sense to use the onboard terminations.

The terminations for the receive groups should be set to terminate with the user software in most cases. [term0 and term6]



<u>CONTROL</u>	<u>CORRESPONDING IO BIT(S)</u>
TERM_0..3	IO_0..3.
TERM4	IO_4..7
TERM5	IO_8..11
TERM6	IO_12..15
TERM7	IO_16..19

### bis\_switch

[\$44 BiSerial Switch Read Port read only]

	<b>CONTROL</b>	<b>REGISTER 1</b>
	<b>DATA BIT</b>	<b>DESCRIPTION</b>
	5	UB5
	4	UB4
	3	UB3
	2	UB2
	1	UB1
	0	UB0

FIGURE 12

PMC-BISERIAL SWITCH READ BIT MAP

The Switch Read Port has the user bits. The user bits are connected to 6 switch positions. The switches allow custom configurations to be defined by the user and for the software to “know” how to configure the read/write capabilities of each IO line. Please note that the lower 6 bits of the switch are implemented [8 positions on switch]. The silk-screen is marked with the ‘0’ and ‘1’ definitions.



## Interrupts

PMC-BiSerial interrupts are treated as auto-vectored. When the software enters into an exception handler to deal with a PMC-BiSerial interrupt the software must read the status register(s) to determine the cause(s) of the interrupt, clear the interrupt request(s) and process accordingly.

The PMC-BiSerial state machines generate an interrupt request when a transmission or reception is complete and the INTEN bits in the control registers are set. The transmission is considered complete when the strobe line is deactivated. The interrupt is mapped to INTA on the PMC connector. INTA may be mapped to a different interrupt in your system. For example in our NT systems it is mapped to interrupt B. The source of the interrupt is obtained by reading BIS\_STAT0. The status remains valid until the status register is read. The interrupt status is auto-cleared when the status register is accessed.

The interrupt level seen by the CPU is determined by the rest of the system. The master interrupt can be disabled or enabled through the bis\_base register. The individual enables for TX and RX are controllable through bis\_tx and bis\_rx.

The individual enables operate before the interrupt holding latches, which store the request for the CPU. Once the interrupt request is set, the way to clear the request is to reset the board, access bis\_intc, or disable the interrupt. Toggling the interrupt enable low will clear the interrupt. The interrupt enable can be set back to enabled immediately. TX\_INT\_EN enables and clears the TX interrupt and RX\_INT\_EN enables and clears the RX interrupt request. The master enable is a mask and can be used to disable the interrupt from reaching the CPU, but still leaves the internal interrupt request hardware active which is useful for polled operation.

If operating in a polled mode and making use of the interrupts for status then the master interrupt should be disabled and the Rx, TX, and programmable level interrupts enabled as needed. When bis\_stat0 shows an interrupt pending the appropriate FIFO action can take place and the enable toggled to remove the interrupt request. One extra read of the bis\_stat0 to make sure that the interrupt request is cleared before starting the next transfer. The first read of bis\_stat0 register does clear the interrupt status. The source of the status is still pending [interrupt request] after the status read and before the enable toggle or bis\_intc operation which removes the source of the status. The status can become



set again before the SW has a chance to clear the source. It is necessary to do one extra read for clearing purposes.

If interrupt driven then the bis\_intc operation can occur before the status is read to clear the request and save the necessity of the second status read. If operating with RX and TX interrupts enabled and random timing, it is recommended that the individual enables be used to clear the interrupt source. It is possible that one interrupt is closely followed by another leading to the second being cleared by the response to the first before the status bit captures the event leading to a lost interrupt request. By using the individual interrupt enables to clear the interrupt, the issue is avoided because the software is only clearing the interrupt that triggered the interrupt cycle. If a second interrupt comes as described in the pathological case then it will remain after the clearing operation. The second status read may capture the event or not depending on the timing. If the second interrupt is captured by the status register then the software can either recognize the second event or choose to ignore it. If recognized the software can save an interrupt cycle by immediately processing the new information. If the information was ignored then a new interrupt request will be created. The host can treat the second interrupt as a new event.

Power on initialization will provide a cleared interrupt request and interrupts disabled.

## Loop-back

The Engineering kit has reference software which includes an external loop-back test. The test requires an external cable with the following pins connected.

Data+	7 - 22
Data-	41 - 56
Clk+	9 - 24
Clk-	43 - 58
Strobe+	11 - 21
Strobe-	45 - 55



## PMC PCI Pn1 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn1 Interface on the PMC-BiSerial-IO. 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 13

PMC-BISERIAL PN1 INTERFACE

## PMC PCI Pn2 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn2 Interface on the PMC-BiSerial-IO. 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 14

PMC-BISERIAL PN2 INTERFACE



# BiSerial Front Panel IO Pin Assignment

The figure below gives the pin assignments for the PMC Module IO Interface on the PMC-BiSerial. Also, see the User Manual for your carrier board for more information. GND\* is a plane which is tied to GND through a 1206 0% resistor. AC or open are options – contact Dynamic Engineering.

IO_0p [ref clk+]	IO_0m [ref clk-]	1	35
IO_1p	IO_1m	2	36
GND*	GND*	3	37
IO_2p	IO_2m	4	38
GND*	GND*	5	39
IO_3p	IO_3m	6	40
IO_4p [TX_DATA+]	IO_4m [TX_DATA-]	7	41
GND*	GND*	8	42
IO_5p [TX_CLK+]	IO_5m [TX_CLK-]	9	43
GND*	GND*	10	44
IO_6p [TX_STB+]	IO_6m [TX_STB-]	11	45
IO_7p	IO_7m	12	46
GND*	GND*	13	47
IO_8p	IO_8m	14	48
GND*	GND*	15	49
IO_9p	IO_9m	16	50
IO_10p	IO_10m	17	51
GND*	GND*	18	52
IO_11p	IO_11m	19	53
GND*	GND*	20	54
IO_12p [RX_STB+]	IO_12m [RX_STB+]	21	55
IO_13p [RX_DATA+]	IO_13m [RX_DATA-]	22	56
GND*	GND*	23	57
IO_14p [RX_CLK+]	IO_14m [RX_CLK-]	24	58
GND*	GND*	25	59
IO_15p	IO_15m	26	60
GND*	GND*	27	61
IO_16p	IO_16m	28	62
GND*	GND*	29	63
IO_17p	IO_17m	30	64
GND*	GND*	31	65
IO_18p	IO_18m	32	66
GND*	GND*	33	67
IO_19p	IO_19m	34	68

FIGURE 15

PMC-BISERIAL FRONT PANEL INTERFACE



## PMC Pn4 User Interface Pin Assignment

The figure provides the pin assignments for the PMC-BiSerial Module routed to Pn4. Also, see the User Manual for your carrier board for more information.

IO_0p	REFCLK+	IO_0m	REFCLK-	1	2
IO_1p		IO_1m		3	4
IO_2p		IO_2m		5	6
IO_3p		IO_3m		8	9
IO_4p	TX_DATAp	IO_4m	TX_DATAm	9	10
IO_5p	TX_CLKp	IO_5m	TX_CLKm	11	12
IO_6p	TX_STBp	IO_6m	TX_STBm	13	14
IO_7p		IO_7m		15	16
IO_8p		IO_8m		17	18
IO_9p		IO_9m		19	20
IO_10p		IO_10m		21	22
IO_11p		IO_11m		23	24
IO_12p	RX_STBp	IO_12m	RX_STBm	25	26
IO_13p	RX_DATAp	IO_13m	RX_DATAm	27	28
IO_14p	RX_CLKp	IO_14m	RX_CLKm	29	30
IO_15p		IO_15m		31	32
IO_16p		IO_16m		33	34
IO_17p		IO_17m		35	36
IO_18p		IO_18m		37	38
IO_19p		IO_19m		39	40
				41	42
				43	44
				45	46
				47	48
				49	50
				51	52
				53	54
				55	56
				57	58
				59	60
				61	62
				63	64

FIGURE 16

PMC-BISERIAL PN4 INTERFACE

# Applications Guide

## Interfacing

The pinout 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 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. Alternatively, the use of OPTO-22 isolation panels is recommended.

**Keep cables short.** Flat cables, even with alternate ground lines, are not suitable for long distances. PMC-BISERIAL 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 BiSerial pin definitions. It is suggested that this standard 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 were conceived and engineered for rugged industrial environments. The PMC-BiSerial is constructed out of 0.062 inch thick FR4 material.

Through hole and surface mounting of components are used. IC sockets use gold plated screw machine pins. High insertion and removal forces are required, which assists in the retention of components. If the application requires unusually high reliability or is in an environment subject to high vibration, the user may solder the corner pins of each socketed IC into the socket, using a grounded soldering iron.

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 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/°C for uniform heat. This is based upon the temperature coefficient of the base FR4 material of 0.31 W/m-°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 BISERIAL 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

Dynamic Engineering warrants this product to be free from defects in workmanship and materials under normal use and service and in its original, unmodified condition, for a period of one year from the time of purchase. If the product is found to be defective within the terms of this warranty, Dynamic Engineering's sole responsibility shall be to repair, or at Dynamic Engineering's sole option to replace, the defective product. The product must be returned by the original customer, insured, and shipped prepaid to Dynamic Engineering. All replaced products become the sole property of Dynamic Engineering.

Dynamic Engineering's warranty of and liability for defective products is limited to that set forth herein. Dynamic Engineering disclaims and excludes all other product warranties and product liability, expressed or implied, including but not limited to any implied warranties of merchandisability or fitness for a particular purpose or use, liability for negligence in manufacture or shipment of product, liability for injury to persons or property, or for any incidental or consequential damages.

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.



## 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. The current minimum repair charge is \$100. 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  
435 Park Dr.  
Ben Lomond, CA 95005  
831-336-8891  
831-336-3840 fax  
support@dyneng.com



## Specifications

Host Interface:	PCI Mezzanine Card
Serial Interface:	RS-485 TX_Data, TX_CLK, TX_STB, RX_Data, RX_CLK, RX_STB, REFCLKI
TX CLK rates generated:	12 bit divisor with 10 MHz, 33 MHz. and REFCLK input rates. Other rates available with special oscillator installation
Software Interface:	Control Registers, Status Ports, FIFO
Initialization:	Hardware Reset forces all registers to 0.
Access Modes:	LW boundary Space (see memory map)
Wait States:	1 for all addresses except FIFO loop-back which requires 8.
Interrupt:	Tx interrupt at end of transmission Rx interrupt at end of reception Programmable Almost Empty Programmable Almost Full
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. Programmable parts are socketed.
Temperature Coefficient:	0.89 W/°C for uniform heat across IP
Power:	Max. <b>TBD</b> mA @ 5V



## Order Information

PMC-BISERIAL-IO	PMC Module with 1 Tx and 1 Rx serial channel, Programmable data rates Standard protocol support, RS-485 drivers and receivers 32 bit data interface
Eng Kit-PMC-BISERIAL	HDEterm68 - 68 position screw terminal adapter HDEcabl68 - 68 IO twisted pair cable Technical Documentation, 1. PMC-BISERIAL Schematic 2. PMC-BISERIAL Reference test software Data sheet reprints are available from the manufacturer's web site reference software.

**Note:** *The Engineering Kit is strongly recommended for first time PMC-BISERIAL buys.*

## Schematics

Schematics are provided as part of the engineering kit for customer *reference only*. This information was current at the time the printed circuit board was last revised. This revision letter is shown on the front of this manual as "Corresponding Hardware Revision." This information is not necessarily current or complete manufacturing data, nor is it part of the product specification.

All information provided is Copyright Dynamic Engineering

