

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-Nvy1

Bi-directional Serial Data Interface PMC Module

Revision OR
Corresponding Hardware: Revision O1

PMC-BiSerial-NVY1
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.

©2001 by Dynamic Engineering.

Trademarks and registered trademarks are owned by their respective manufactures.
Manual Revision DR. Revised June 4, 2001.



Table of Contents

PRODUCT DESCRIPTION	6
THEORY OF OPERATION	9
ADDRESS MAP	12
PROGRAMMING	13
Register Definitions	15
bis_base	15
bis_tx	17
bis_rx	18
bis_stat0	20
bis_intc	21
bis_fifotx	21
bis_fiforx	21
bis_dir_term	22
bis_syncw1h	24
bis_syncw1l	24
bis_syncw2h	25
bis_syncw2l	25
bis_switch	26
Flag Insertion	27
Interrupts	27
Loop-back	28
PMC PCI PN1 INTERFACE PIN ASSIGNMENT	30
PMC PCI PN2 INTERFACE PIN ASSIGNMENT	31
BISERIAL FRONT PANEL IO PIN ASSIGNMENT	32
PMC PN4 USER INTERFACE PIN ASSIGNMENT	33
APPLICATIONS GUIDE	34



Interfacing	34
Construction and Reliability	35
Thermal Considerations	35
WARRANTY AND REPAIR	36
Service Policy	37
Out of Warranty Repairs	37
For Service Contact:	37
SPECIFICATIONS	38
ORDER INFORMATION	39
SCHEMATICS	39



List of Figures

FIGURE 1	PMC-BISERIAL BLOCK DIAGRAM	6
FIGURE 2	MANCHESTER DATA ENCODING	9
FIGURE 3	DATA PATH BLOCK DIAGRAM	11
FIGURE 4	PMC-BISERIAL INTERNAL ADDRESS MAP	12
FIGURE 5	PMC-BISERIAL BASE CONTROL REGISTER BIT MAP	15
FIGURE 6	PMC-BISERIAL TX CONTROL REGISTER BIT MAP	17
FIGURE 7	PMC-BISERIAL RX CONTROL REGISTER BIT MAP	18
FIGURE 8	PMC-BISERIAL STATUS REG 0 BIT MAP	20
FIGURE 9	PMC-BISERIAL DIRECTION TERMINATION CONTROL BIT MAP	22
FIGURE 10	PMC-BISERIAL RX SYNC WORD 1H REGISTER BIT MAP	24
FIGURE 11	PMC-BISERIAL RX SYNC WORD 1L REGISTER BIT MAP	24
FIGURE 12	PMC-BISERIAL RX SYNC WORD 2H REGISTER BIT MAP	25
FIGURE 13	PMC-BISERIAL RX SYNC WORD 1L REGISTER BIT MAP	25
FIGURE 14	PMC-BISERIAL SWITCH READ BIT MAP	26
FIGURE 15	PMC-BISERIAL PN1 INTERFACE	30
FIGURE 16	PMC-BISERIAL PN2 INTERFACE	31
FIGURE 17	PMC-BISERIAL FRONT PANEL INTERFACE	32
FIGURE 18	PMC-BISERIAL PN4 INTERFACE	33

Product Description

PMC-BiSerial-NVY1 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 NVY1 protocol implemented provides Manchester encoded data inputs and outputs.

Other 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.

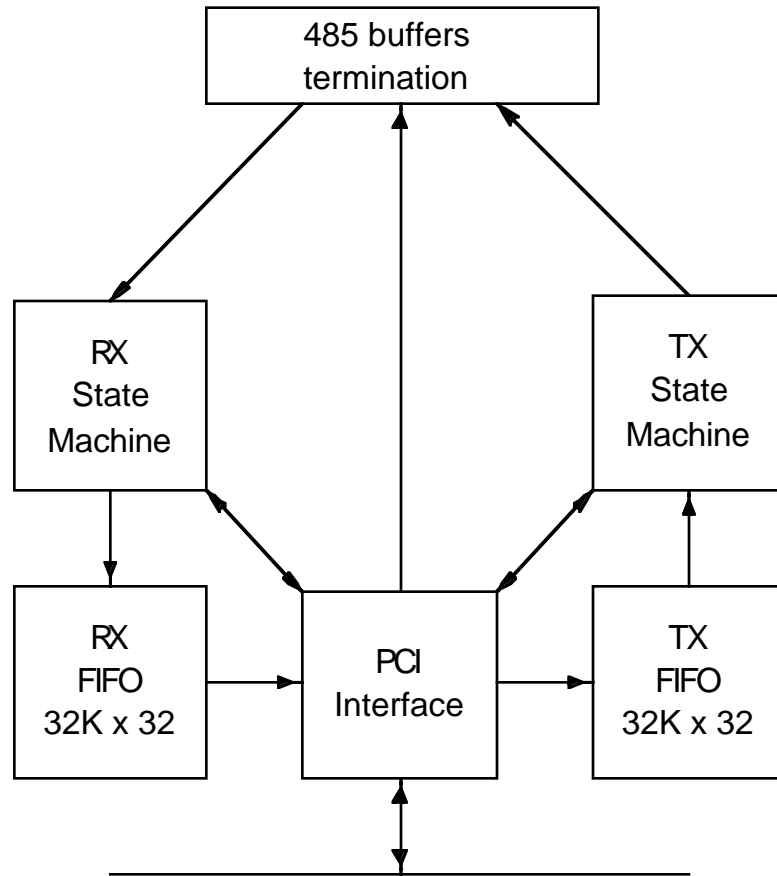


FIGURE 1 PMC-BISERIAL BLOCK DIAGRAM

Several clocking options are available with the NVY1 version. The PCI Clock, external reference input, or a division of the 62.208 MHz oscillator are the clock sources. The source can be selected with software. A 12-bit



frequency divider is provided for the oscillator to offer output data rates from 15.552 Mbps to 7.59375 Kbps. The acceptable input data rate is continuous from 5.2 to 12.4 Mbps. If a specific frequency is required that is not attainable with the standard choices we can install a different 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 130Ω . 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 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 32K 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 channels can receive continuous or bursted data in two paired data streams. As each pair of 12-bit words is received, the data is



stored into the Receive FIFO. The host can poll the FIFO flags or wait for the programmable FIFO Almost Full flag interrupt. The message can be read directly from the input FIFO.

The Output channel has a separate 32K x 32-bit FIFO. The FIFO is written as long words, and the transmitter can be configured to send the entire 32 bits or the lowest 1, 2, or 3 bytes. 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.

Smaller FIFO's are available as a special order.

Interrupts are supported by the PMC-BiSerial-NVY1. The interrupt occurs at the end of the transmitted message. In addition to this protocol interrupt there are interrupts associated with the programmable FIFO flags, the acquisition and loss of lock and the detection of incoherent clocks on the receiver. 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. 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 32K 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-NVY1 is designed for transferring data from one point to another with a manchester encoded serial protocol.

The PMC-BiSerial-NVY1 features a Xilinx FPGA. The FPGA contains all of the registers and protocol controlling elements of the BiSerial design. Only the transceivers, switches, and FIFO's 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 FIFO's. 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 PMC-BiSerial-NVY1 uses Manchester encoded data. At least one transition occurs in each data period allowing the clock to be recovered from the data stream. The timing

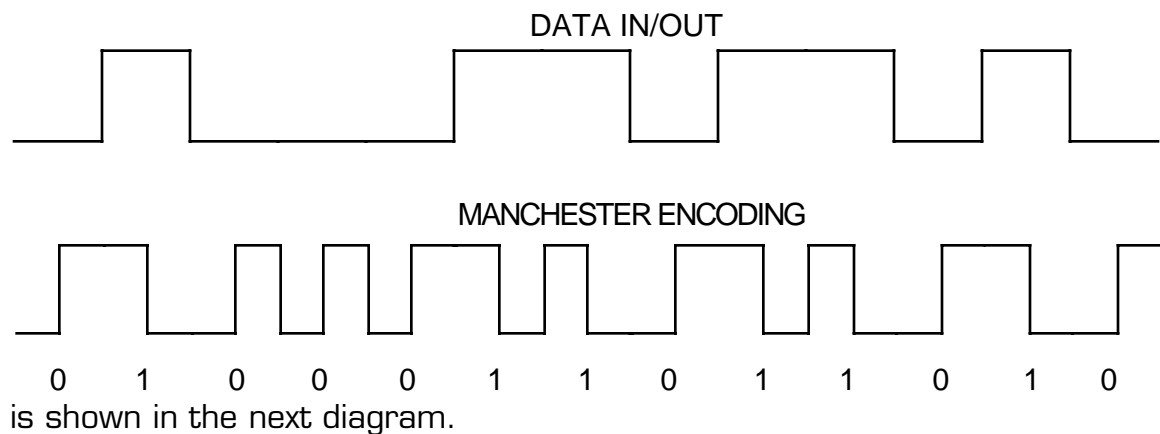


FIGURE 2

MANCHESTER DATA ENCODING



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 FIFO's 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.

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. One half bit period later the data value is inverted to encode the clock on the data stream. One half bit 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 FIFO's 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.

Clock reference sources include the PCI clock, a user input clock, and a division of the onboard oscillator. The divider is a 12-bit counter with programmable divisor. Please refer to the memory map for more details.

The receive function is more complex. The data is received in two separate streams which are assumed to be synchronous with each other. When the receiver is enabled the Manchester decoding circuit attempts to lock onto the proper bit period. Once a transition occurs in the data from a one to a zero or a zero to a one the decoder can distinguish between the mid-period transitions that encode the clock and the inter-period transitions that may or may not be present depending on the data pattern. The Manchester decoder recovers the clock from the data and the data proceeds to the sync detection circuit.

The sync detector searches for one of two 48-bit sync patterns and, if the pattern is found, inserts a four bit flag into the data stream replacing the least significant four bits in the third 12-bit word following the sync pattern. Once the sync pattern is found the alignment of the 12-bit words is determined. The two data streams are concatenated into one 32-bit wide data stream, with zeros inserted into bit positions 12-15 and 28-31, and the data is written into the FIFO's.

A simplified block diagram of the TX and RX paths is shown in the figure below.



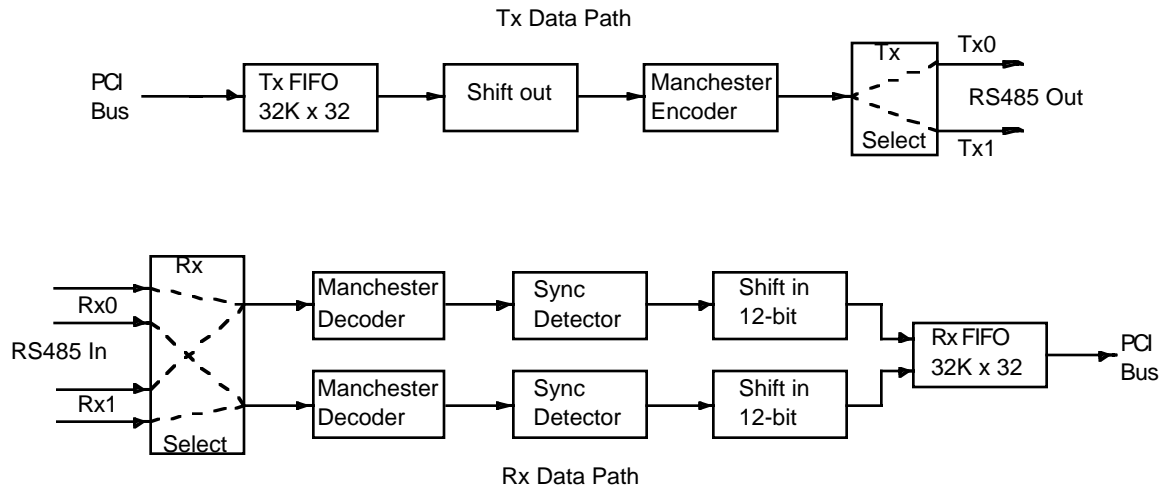


FIGURE 3

DATA PATH BLOCK DIAGRAM

Address Map

REGISTER	OFFSET	FUNCTION	TYPE
bis_base	EQU \$00	base control register	read/write
bis_tx	EQU \$04	transmit base control	read/write
bis_rx	EQU \$0c	receive base control	read/write
bis_stat0	EQU \$14	status register 0	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_syncw1h	EQU \$2C	sync word 1 - 24 upper bits	read/write
bis_syncw1l	EQU \$30	sync word 1 - 24 lower bits	read/write
bis_syncw2h	EQU \$34	sync word 2 - 24 upper bits	read/write
bis_syncw2l	EQU \$38	sync word 2 - 24 lower bits	read/write
bis_switch	EQU \$3C	read user switch	read

FIGURE 4

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 = 0x0008. Current revision = 0x00

Programming

Programming the PMC-BiSerial-NVY1 requires only the ability to read and write data from the host. The base address is determined by the PMC Carrier board. The base address refers to the first user address for the slot 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 "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 FIFO and state machine; load the sync word values, sync and store options, and scan length. Depending on these values, and the sync pattern received; data will be loaded into the FIFO's.

The interrupt service routine should be loaded and the mask should be set. The routine can be configured to respond to the acquisition or loss of sync lock or the programmable almost full interrupts, or the FIFO empty flag can be polled to indicate data is present in the FIFO. After the interrupt is received, the data can be retrieved. 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.

The TX interrupt indicates to the software that a message has been sent and that the message has terminated. If more than one interrupt is enabled, then the SW needs to read BIS_STATO to see which source caused the interrupt. Reading BIS_STATO will clear the interrupt status and accessing BIS_INTC will clear the TX 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 interrupt status bits to zero before Start is enabled. This will insure that the interrupt status values read by the interrupt service routine came from the current reception.

Before transmitting data, the FIFO's are enabled and the data loaded, the clock rate and the data size are selected. Once the complete message is loaded and the controls set properly, the start bit can be set to cause the transfer to begin. 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 32K bytes can be accommodated by polling or using



the programmable flag interrupts. To poll read the Status O register during the transfer and take appropriate action. T full, empty or programmable flags show 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
31-24	spare
23	Unit select
22	frx_ld
21	ftx_ld
20	fifo_en
19-18	spare
17	int_set
16	int_en_master
15	spare
14-12	clock selector
11-0	clock divisor

FIGURE 5

PMC-BISERIAL BASE CONTROL REGISTER BIT MAP

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

Unit select is used to select the active unit that is communicating with the board. When this bit is a zero unit 4 is selected(TX – IO4, Rx – IO12, 13); when unit select is a one, unit 5 is selected(TX – IO5, Rx – IO14, 15).

FRX_LD is tied to the RX FIFO WE2/_LD pin. FTX_LD is tied to the TX FIFO WE2/_LD pin. When the FIFO's 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 FIFO's 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 FIFO's 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 Selector

1XX	oscillator divided by 2^n
010	external
011	pci clock

The external clock is IO channel 0.

Divisor [11-0] are the clock divisor bits. The oscillator is divided by this value and then again by two to produce a square wave output. The resulting frequency is equal to $62.208 \text{ Mhz}/2^n$, $n \geq 2$.

Please note that the 485 buffers are rated for 12 MHz. With most systems the larger divisors will be used.



bis_tx

[04] BiSerial Tx Control Register Port read/write

CONTROL TX	
DATA BIT	DESCRIPTION
31-11	spare
10-9	idle pattern
8	invert output
7	tx1_en (test mode)
6	tx0_en (test mode)
5-4	data size
3	tx_testmode
2	int_en_pae
1	int_en_tx
0	start_tx

FIGURE 6

PMC-BISERIAL TX CONTROL REGISTER BIT MAP

Idle pattern defines the pattern that is output between actual messages. This is a two bit pattern that is repeated continuously until another message is sent.

Invert output inverts the output data.

Tx1_en enables the Tx data onto IO5 when tx_testmode is enabled.

TxO_en enables the Tx data onto IO4 when tx_testmode is enabled.

Data size selects the size of the TX data word. A value of '00' outputs only the lowest 8 bits of the TX data word, '01' outputs 16-bits, '10' outputs 24 bits, and '11' outputs all 32 bits.

Tx_testmode is selected to enable the Tx1_en and TxO_en bits, when this bit is zero, the unit select bit in the control register controls the output line.

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_rx

[\$0C] BiSerial Rx Control Register Port read/write

CONTROL RX	
DATA BIT	DESCRIPTION
31-22	spare
21-12	scan sequence length
11	invert input
10-9	Store options
8	Lock status source
7-6	Sync options
5	rx_testmode
4	int_en_lock_loss
3	int_en_lock_acquire
2	int_en_paf
1	int_en_rx_clk_err
0	start_rx

FIGURE 7

PMC-BISERIAL RX CONTROL REGISTER BIT MAP

Scan sequence length specifies the number of bits from the beginning of one sync word to the beginning of the next sync word.

Invert input inverts the input data.

Store options determines when data is stored in the FIFO's. If bit 9 is a zero data is stored before sync is detected, when a valid sync is seen the data is aligned to the sync word and data continues to be stored. If bit 9 is a one no data is stored before the first sync is seen. If bit 10 is a one, data is not stored between the first sync and sync lock (three consecutive syncs detected). Therefore, if this field is '00' all data is stored, if '01' data is stored only after a sync is seen and if '11' no data is stored until sync lock is achieved.

Lock status source determines the source for the lock status bit and the lock acquisition and loss interrupts. If this bit is a zero, the lock from the upper and lower data streams is or-ed and if it is a one these locks are and-ed.

Sync options determines how the sync detection is coordinated in the two data streams. If this field is '00' no syncs are detected, if '01' a sync must be seen simultaneously in both data streams to be detected, if '10' the sync detectors operate independently, and if '11' a sync seen by either stream will be detected by both.



rx_testmode is used in the Rx fifo test to select the PCI bus as the data source for receive data.

int_en_lock_loss when '1' enables the interrupt that is generated when lock is lost after being achieved. Requires the master enable to create a system level interrupt.

int_en_lock_acquire when '1' enables the interrupt that is generated when lock is achieved. Requires the master enable to create a system level interrupt.

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_clk_err when '1' enables the receive clock error interrupt. A clock error condition occurs when the two data streams are not synchronized or if only one is present. Requires the master enable to create a system level interrupt.

start_rx is used to enable the receive state machine to receive messages.



bis_stat0

[S14] BiSerial Status Port read only

STATUS		
DATA BIT	DESCRIPTION	
15	int_request	1 = interrupt request after int_en mask
14	spare	undefined
13	fae_int	1 = programmable almost empty int rqst
12	tx_int	1 = tx int rqst
11	ftx_ffn	0 = tx fifo full 1 = not full
10	ftx_faen	0 = almost empty 1 = not almost empty
9	ftx_mtn	0 = tx fifo empty, 1 = not empty
8	slock_int	1 = sync lock acquired interrupt asserted
7	sunlock_int	1 = sync lock lost interrupt asserted
6	relkerr_int	1 = rx clock error interrupt asserted
5	faf_int	1 = fifo almost full interrupt asserted
4	rx_lock	1 = rx sync lock condition present
3	rx_clkerr	1 = rx clock error detected
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 8

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, the FIFO still has room.

When int_request is set then 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. 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 a write to bis_intc has been performed, 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. The other interrupt sources are edge triggered. Therefore, once the status is read the interrupt will be cleared and the interrupt will not reoccur until the generating condition goes inactive and then active again.

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 FIFO's. 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 FIFO's. 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 DIR_TERM REGISTER			
DATA BIT		DESCRIPTION	
7-0		DIRection	7-0 0 = read 1 = drive
23-16		TERMination	7-0 1 = terminated

FIGURE 9

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_NVY1 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.

<u>CONTROL</u>	<u>CORRESPONDING IO BIT(S)</u>
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_syncw1h

[\$2C] BiSerial Sync Word 1H Register Port read/write

RX SYNC 1H	
DATA BIT	DESCRIPTION
31-24	spare
23-0	upper 24 bits of sync word 1 (ping sync)

FIGURE 10

PMC-BISERIAL RX SYNC WORD 1H REGISTER BIT MAP

The Sync word registers allow the user to specify the sync patterns that the circuit looks for to establish data framing and control data storage. This register contains the upper 24 bits of the 48-bit ping sync word.

bis_syncw1l

[\$30] BiSerial Sync Word 1L Register Port read/write

RX SYNC 1L	
DATA BIT	DESCRIPTION
31-24	spare
23-0	lower 24 bits of sync word 1 (ping sync)

FIGURE 11

PMC-BISERIAL RX SYNC WORD 1L REGISTER BIT MAP

The Sync word registers allow the user to specify the sync patterns that the circuit looks for to establish data framing and control data storage. This register contains the lower 24 bits of the 48-bit ping sync word.

bis_syncw2h

[\$34] BiSerial Sync Word 2H Register Port read/write

RX SYNC 2H	
DATA BIT	DESCRIPTION
31-24	spare
23-0	upper 24 bits of sync word 2 (scan sync)

FIGURE 12

PMC-BISERIAL RX SYNC WORD 2H REGISTER BIT MAP

The Sync word registers allow the user to specify the sync patterns that the circuit looks for to establish data framing and control data storage. This register contains the upper 24 bits of the 48-bit scan sync word.

bis_syncw2l

[\$38] BiSerial Sync Word 1L Register Port read/write

RX SYNC 2L	
DATA BIT	DESCRIPTION
31-24	spare
23-0	lower 24 bits of sync word 2 (scan sync)

FIGURE 13

PMC-BISERIAL RX SYNC WORD 1L REGISTER BIT MAP

The Sync word registers allow the user to specify the sync patterns that the circuit looks for to establish data framing and control data storage. This register contains the lower 24 bits of the 48-bit scan sync word.



bis_switch

[\$3C] BiSerial Switch Read Port read only

USER CONTROL SWITCH REGISTER	
DATA BIT	DESCRIPTION
5	UB5
4	UB4
3	UB3
2	UB2
1	UB1
0	UB0

FIGURE 14

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 [positions 7 & 8 are unused]. The silk-screen is marked with the ‘0’ and ‘1’ definitions.

Flag Insertion

When a sync sequence is first recognized and when subsequent sync sequences are recognized with the appropriate delay (specified by the scan sequence length field in the RX control register), a four-bit pattern will be inserted. This pattern replaces the least significant four bits in the third 12-bit word following the sync word. When a ping sync (sync word 1) is detected a '0110' pattern will be inserted, when a scan sync (sync word 2) is detected a '1001' pattern will be inserted.

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 to determine the cause(s) of the interrupt, clear the interrupt request(s) and process accordingly.

The PMC-BiSerial TX state machine generates an interrupt request when a transmission is complete and the INTEN bit in the control registers is 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. 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_STATO. 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 conditions 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 access bis_intc and bis_statO, or reset the board. Toggling the interrupt enable low will clear the TX interrupt. The interrupt enable can be set back to enabled immediately. TX_INT_EN enables and clears the TX 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 status, TX, and programmable level interrupts enabled as needed. When bis_statO shows an interrupt pending the appropriate FIFO action can take place. One extra read of the bis_statO can be made to make sure that the interrupt request is cleared before starting the next transfer. The first read of bis_statO register does clear the interrupt status. The source of the status may still be pending [TX 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. In this case 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 the TX interrupt enabled and random timing, it is recommended that the individual enable be used to clear this 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 enable to clear the TX 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 above, 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. Select either Input1 or InputO, but not both at the same time as this will double terminate the data lines and degrade the signal



amplitude.

<u>SIGNAL</u>	<u>OUTPUT</u>	<u>INPUT0</u>	<u>INPUT1</u>
Data0+	7	21	24
Data0-	41	55	58
Data1+	9	22	26
Data1-	43	56	60



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 15

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 16

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 0Ω resistor. DC, AC or open are options. For customized version, or other 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_2n	4	38
GND*	GND*	5	39
IO_3p	IO_3n	6	40
IO_4p [TX0_DATA+]	IO_4n [TX0_DATA-]	7	41
GND*	GND*	8	42
IO_5p [TX1_DATA+]	IO_5n [TX1_DATA-]	9	43
GND*	GND*	10	44
IO_6p	IO_6n	11	45
IO_7p	IO_7n	12	46
GND*	GND*	13	47
IO_8p	IO_8n	14	48
GND*	GND*	15	49
IO_9p	IO_9n	16	50
IO_10p	IO_10n	17	51
GND*	GND*	18	52
IO_11p	IO_11n	19	53
GND*	GND*	20	54
IO_12p [RX0L_ DATA+]	IO_12n [RX0L_ DATA-]	21	55
IO_13p [RX0U_ DATA+]	IO_13n [RX0U_ DATA-]	22	56
GND*	GND*	23	57
IO_14p [RX1L_ DATA+]	IO_14n [RX1L_ DATA-]	24	58
GND*	GND*	25	59
IO_15p [RX1U_ DATA+]	IO_15n [RX1U_ DATA-]	26	60
GND*	GND*	27	61
IO_16p	IO_16n	28	62
GND*	GND*	29	63
IO_17p	IO_17n	30	64
GND*	GND*	31	65
IO_18p	IO_18n	32	66
GND*	GND*	33	67
IO_19p	IO_19n	34	68

FIGURE 17

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	TX0_DATAp	IO_4m	TX0_DATAn	9	10
IO_5p	TX1_DATAp	IO_5m	TX1_DATAn	11	12
IO_6p		IO_6m		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	RX0L_DATAp	IO_12m	RX0L_DATAn	25	26
IO_13p	RX0U_DATAp	IO_13m	RX0U_DATAn	27	28
IO_14p	RX1L_DATAp	IO_14m	RX1L_DATAn	29	30
IO_15p	RX1U_DATAp	IO_15m	RX1U_DATAn	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 18

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 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 - 32 bit
Serial Interface:	RS-485 TX0_Data, TX1_Data, TX_STB, RX0L_Data, RX0U_Data, RX1L_Data, RX1U_Data, REFCLKI
Tx Data rates generated:	15.552 MHz to 7.59375 KHz with 62.208 MHz oscillator, 33 MHz and REFCLK input rates. Other rates are available with different oscillator installation.
Rx Data rates accepted:	Continuous from 5.2 to 12.4 Mbps with 62.208 MHz oscillator.
Software Interface:	Control Registers, Status Port, FIFO's
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 Clock Error Rx Sync Lock Acquired Rx Sync Lock Lost Tx Programmable Almost Empty Rx 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 PMC
Power:	Max. TBD mA @ 5V



Order Information

PMC-BiSerial-NVY1

PMC Module with 2 TX and 2 paired Rx serial channels
Programmable TX data rates
Manchester encoded data inputs and outputs
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-NVY1 Reference test software
Data sheet reprints are available from the manufacturer's web site
reference software: C source code requires WinRT.

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

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

