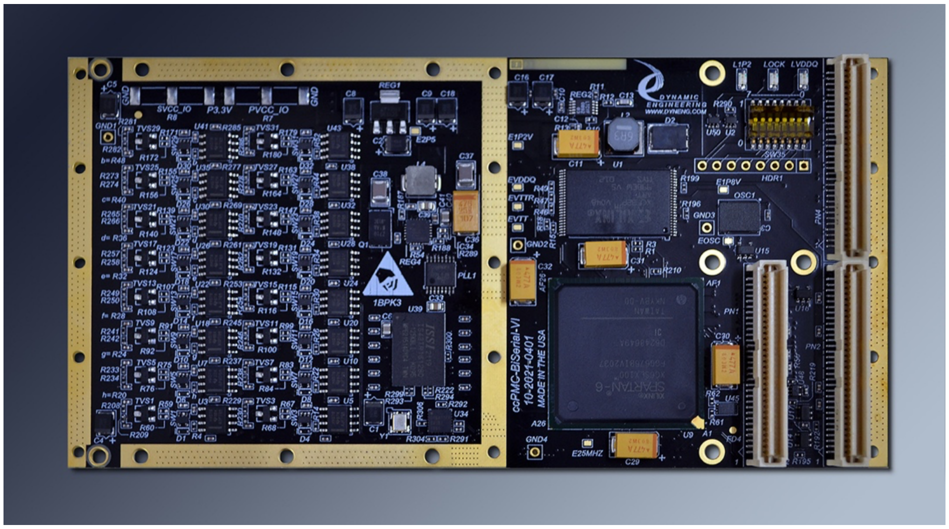
DYNAMIC ENGINEERING
150 DuBois, Suite B/C Santa Cruz, CA 95060
(831) 457-8891
https://www.dyneng.com sales@dyneng.com
Est. 1988

User Manual

ccPMC-BiSerial-VI-ORN1
SDLC & NRZ-L

2 Ports SDLC
2 Ports NRZ-L

Conduction Cooled PMC Module



Manual Revision 1p4 5/26/23
Corresponding PCB : 10-2021-0401/2

ccPMC-BiSerial-VI-ORN1
PMC Module

Dynamic Engineering
150 DuBois, Suite B/C
Santa Cruz, CA 95060
(831) 457-8891

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

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

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

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

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

Table of Contents

List of Figures

Product Description

ccPMC-BiSerial-VI-ORN1 is part of the PMC Module family of modular I/O components by Dynamic Engineering. ccPMC-BiSerial-VI is capable of providing multiple protocols. The ORN1 implementation ports the SDLC function from PMC-BiSerial-III and adds a new NRZ-L function. Two ports of each.

Other custom interfaces are available. We will redesign the state machines and create a custom interface protocol. Please see our web page for current protocols offered. Please contact Dynamic Engineering with your custom application.

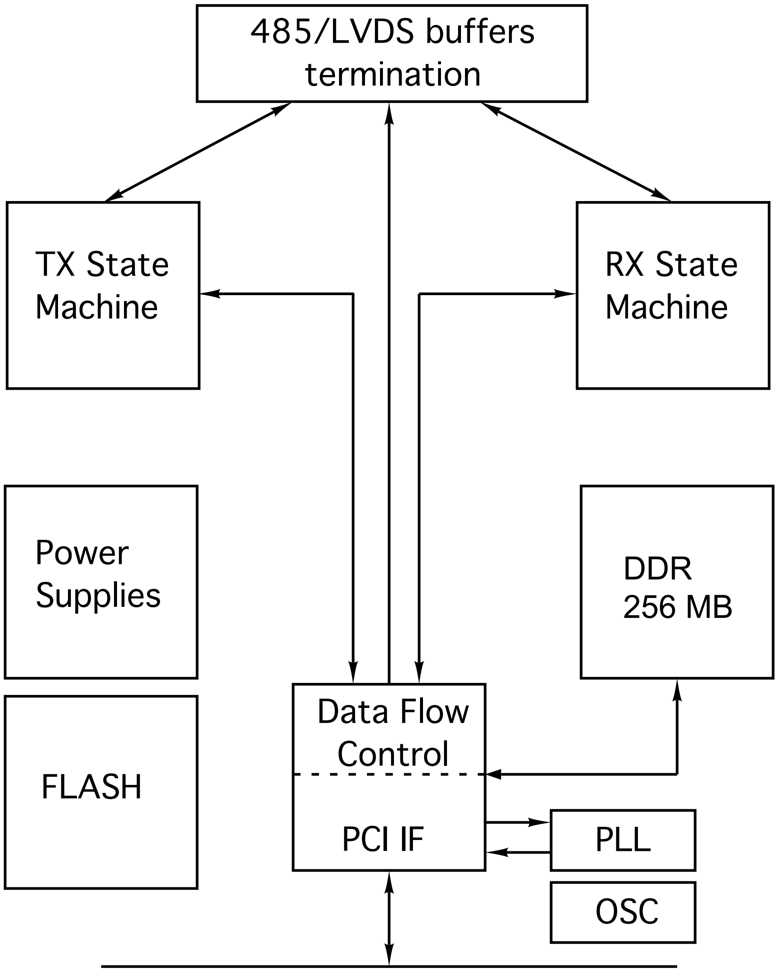


FIGURE 1 CCPMC BISERIAL-VI BASE BLOCK DIAGRAM

ccPMC-BiSerial-VI conforms to the PMC and CMC 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 uses a different one. Contact Dynamic Engineering for a copy of this specification. It is assumed that the reader is at least casually familiar with this document.

In standard configuration, ccPMC-BiSerial-VI is a Type 1 mechanical with only low-profile components on the back of the board and one position wide, with 10 mm inter-board height 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 ccPMC-BiSerial-VI, please let us know. We may be able to do a special build with a different height connector to compensate.

The block diagram shown in Figure 1 highlights the main features of the design. The FPGA is a Spartan-VI in the 676 package. The 100 size is typically used and other sizes can be supported. Industrial temperature parts are used throughout. DDR is available, and when the design requires larger memory configurations, integrated into the implementation. The PLL provides 4 clock references in addition to the PCI and oscillator inputs. The DDR utilizes a 100 MHz differential reference. This reference can also be used for other aspects of the design. The PLL is referenced to 40 MHz. The 40 MHz reference comes from a 32 MHz oscillator.

32 differential IO are provided each with separate termination and direction control. The IO transceivers can be installed with RS-485 or LVDS devices. Optional pull-up/pull-down resistors can also be installed to provide a logic ‘1’ when the lines are not driven. The terminations and transceivers are programmable through the Xilinx device to provide the proper mix of outputs and inputs and terminations needed for a specific protocol implementation. The terminations are programmable for all I/O.

The **SDLC** implementation has two 4 Kbyte Dual Port RAM (DPR) blocks implemented using the Xilinx internal block RAM per port. Each DPR is configured to have a 32-bit port on the PCI side, and a 16-bit port on the I/O side. See Figure 2 for a representation of the SDLC circuit.

The SDLC interface uses programmable PLL clock A as a reference frequency to sample the internal or external transmitter clock. PLL clock B is used to generate the Tx clock when using internal clock mode. Clock and data, in and out, comprise the four I/O lines of each SDLC port. The two DPRs are partitioned into one block each for transmit and receive. The RAM blocks are used as circular buffers that have independently specified start and stop addresses and separate transmit and receive interrupts.

NRZ-L is implemented in two ports. Each port has Transmit, and Receive capabilities. To support the transmitter a Data FIFO (0x3FFF - 32 bit words), and a Packet FIFO with room for x3FF descriptors. The receive side has the same configuration.

Separate controls for Tx and Rx allow user selection of Clock Sense, Data Sense, MSB / LSB order, and interrupt enables. With the various options one can interface with NRZ or NRZL. In addition, the Tx side has a register to select the bit rate transmitted. The receiver auto detects the clock and does not require frequency selection. Both Transmitter and Receiver have programable times to control the time between transmissions or how much time to wait to detect the end of packet.

Packet descriptors are used by the transmitter to know how many bits to send in one packet. The receiver generates packet descriptors to document the number of bits stored in a received packet.

Status is provided for Overrun and Underrun. Interrupts are available for transmission or reception of a packet. Programmable Almost Full [Rx Data FIFO] and Almost Empty [Tx Data FIFO] are provided. Additional counts for Data and Packet FIFOs, Full and Empty status plus state machine Idle condition are provided.

Built in test is provided with R/W registers plus loop-back between the Data FIFOs.

All the data **I/O** lines are programmable to be register controlled or state-machine controlled. Any or all of the bits can be used as a parallel port instead of being dedicated to a specific I/O protocol. Thirty-two differential I/O are provided at the rear IO connector. The drivers and receivers conform to the RS-485 specification (exceeds RS-422 specification). The RS-485 input signals are selectively terminated with 100Ω. The termination resistors are in separate packages to allow flexible termination options for custom formats and protocols.

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

Interrupts are supported by PMC-BiSerial-VI-ORN1. All interrupts can be individually masked, and a master interrupt enable is also provided to disable all interrupts simultaneously. The current interrupt status is available whether an individual interrupt is enabled or not making it possible to operate in polled mode.

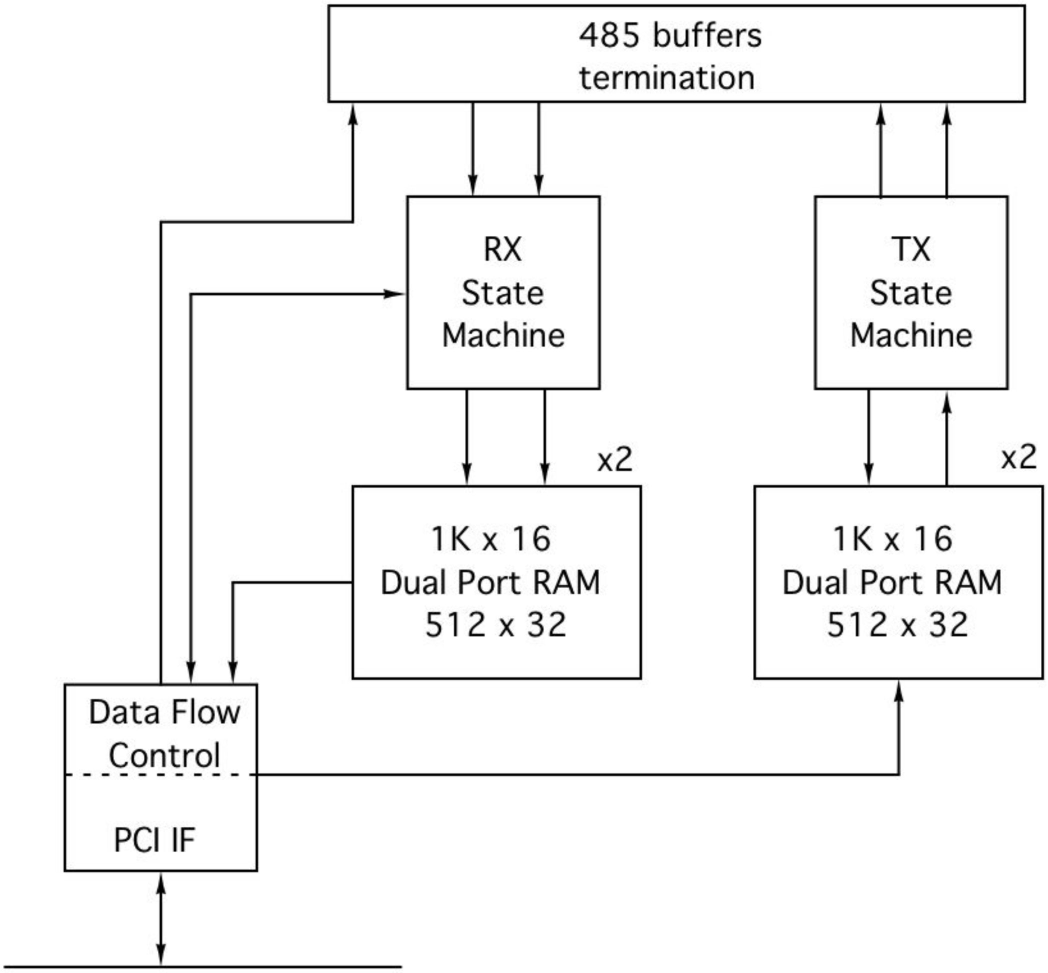


FIGURE 2 CCPMC BISERIAL-VI-ORN1 SDLC BLOCK DIAGRAM

Theory of Operation

ccPMC-BiSerial-III-ORN1 features a Xilinx FPGA. The FPGA contains all of the registers and protocol controlling elements of the BiSerial-VI design.

ccPMC-BiSerial-VI can support many protocols. ccPMC-BiSerial-VI-ORN1 supports two ports of full-duplex SDLC plus 2 full-duplex ports of NRZL.

SDLC Description:
This synchronous interface has separate clock and data inputs and outputs. Each message is delimited by eight-bit flag characters. The beginning flag and the ending flag enclose the SDLC frame. Both beginning and ending flags have the binary format 01111110.

The ending flag for one frame may serve as the beginning flag for the next frame. Alternatively, the ending zero of an ending flag may serve as the beginning zero of a beginning flag, thus forming the pattern '011111101111110'.

The transmitter may insert multiple flags between frames to maintain the active state of the link if time fill between message frames is required.

In order to avoid false flag detection from the data pattern, the SDLC interface uses zero insertion. If five consecutive ones appear anywhere in the data stream, a zero is inserted to avoid having six consecutive one bits. On the receive side, when five ones are received the sixth bit is monitored. If it is a zero, it is removed from the data stream, if it is a one then either a start/stop flag or an abort character (0xFE) has been detected.

Any ending flag may be followed by a frame, by another flag, or by an idle condition. The idle condition is signaled by a minimum of 15 consecutive one bits. As long as 1's continue to be sent, the link remains in the idle state.

To send a message, write the message data to the transmit memory, specify the start and stop addresses and configuration control bits, then enable the transmitter. The state-machine will load the start address, send the beginning flag character and send the data sequentially LSB first until the end address is reached and the ending flag is sent.

As soon as the beginning flag is sent, the sending status bit will be asserted. At that time the ending address will be latched in the transmitter and new addresses can be written for the next message to be sent. This message will be sent as soon as the current message completes. If a new transmit starting address is not written, the transmitter will continue reading data with the next address after the stop address of the current frame. A new transmit end address must be written to trigger sending an additional message-frame.

If the TX Clear control bit is enabled, when no more message frames are left to transmit; automatically disable the transmitter and the TX interrupt will be asserted.

If the TX clear control bit is not enabled, the transmitter will remain enabled after the last message, and the TX interrupt will still be asserted. When multiple frames are being sent, the frame done interrupt will be asserted at the end of each message-frame.

The TX interrupt will only occur after the last frame and the transmitter will wait, pointing at the next address after the end address. If additional data has been or is later written to the DPR, a new message can be started by entering a new end address (and optionally a new start address). The transmit state-machine will start the new message and continue sending data until the new end address has been reached. If the end address of the message is less than the start address of the message, when the end of memory is reached the transmitter will wrap around ⇔ from the start of memory until the end address is reached.

To receive a message the receiver must be enabled. Only the starting address of the receive buffer is specified. Data will be stored sequentially in the next address after the starting address, incrementing until the closing flag is detected. This will latch a receiver done interrupt status and can cause an interrupt if enabled. The last address data was stored into is written to the starting address location for that message-frame.

This allows any received message to be quickly accessed in the received data by reading the address pointer in the message start location, which points to the end address of the first message-frame. The memory location following the end of the first message-frame contains the end address of the second message-frame. This process can be repeated as many times as needed to find the message of interest.

At the end of each frame, the end address is also latched and can be read from the control register as a read-only field, but this will be overwritten as subsequent frames complete. The transmit interrupt is mapped to the first interrupt line of the selected port, the transmit frame done interrupt is mapped to the second interrupt line, the receive interrupt is mapped to the third interrupt line and the abort received interrupt is mapped to the fourth interrupt line of the selected port.

When a frame completes and no more message-frames are pending, the bus can stay active by continually sending flags or it can go idle by sending ones. The SDLC Idle After Frame Done control bit determines this behavior for the transmitter. If this bit is not set and the bus remains active by sending multiple flags, the Repeated Flags Share Zero control bit determines whether the transmitter sends a ‘0111111001111110’ or a ‘011111101111110’ pattern while waiting for a new message-frame to be requested. When the transmitter is disabled the bus defaults to a high state, which is equivalent to the idle condition.

The PLL is configured to supply a 48 MHz signal on its clock A output. This is used to sample the transmit reference clock to detect transitions. These transitions are used to determine when to drive the next data bit onto the transmit data I/O line. The transmitter clock reference can be supplied by an external source or an internal clock reference provided by PLL clock B.

For test purposes, a substitute external clock is created by routing the output from PLL clock B onto I/O configured as outputs. These clocks may be connected externally to any or all selected ports for loopback testing. A control bit in each port's control register is used to select between these two options. When the internal clock mode is selected the transmit clock line is configured as an output, but when the external clock mode is selected the transmit clock line is configured as an input. The transmit data line is always an output and the receive clock and data lines are always inputs.

NRZL is a common interface. Clock and Data on a differential IO standard. ORN1 has several programmable features to create an adaptable interface.

1. MSB or LSB first selection
2. Active Clock Edge selection – 50/50 data period with rising or falling edge centered for transmission
3. Standard or Inverted Data
4. Number of bits to transmit
5. Frequency of transmission
6. Auto Frequency Rx
7. Automatic programmable time between packets transmitted
8. Programmable end of packet detect for Rx
9. Interrupts and status to control operation
10. Full duplex support

To transmit the HW is programmed for the mode used in your system. Select MSB or LSB first transmission, Active edge of the transmitted clock, Standard or inverted data, and if you want to enable an interrupt request at the end of each transfer. These values are written to the Tx Control Register. If receiving the same choices are available with the Rx Control Register. The choices do not need to match. If performing loop-back you will want to have them match.

If sending multiple packets with HW control of the timing program the Tx Gap register with the requested delay.

Program the Tx Rate register to set the clock period when transmitting.

Load data into the Data FIFO. Write a Descriptor to the Packet FIFO.

The last step is repeated for each new transmission. No need to repeat the others.

In addition, make sure the Parallel Port Mux and Termination registers are programmed to support the NRZL port operation.

For reception the operation is symmetrical with 2 key differences.
1. Data and Descriptor are read from the port.
2. Meaning of the Rx GAP.   For the Tx GAP if left programmed to 0x00 [reset default] the HW will operate and ignore the gap since it is zero. For the Rx side the Gap time is based on the reference clock rather than the Tx 2X rate and is used to determine the time to wait to determine end of packet has been reached. Program to 2X the expected period or similar.

Address Map

BASE Map

Base	0x0000	Pointer to Base memory space	
BIS6_BASE	0x0000	0	Base control register
BIS6_BASE1	0x0004	1	Master Interrupt Enable
BIS6_STATUS	0x0008	2	Interrupt and other Status
BIS6_ID	0x000C	3	Switch, Revision register
BIS6_IO_DATA	0x0010	4	Data register 31 - 0
BIS6_IO_DIR	0x0014	5	Direction register 31 - 0
BIS6_IO_TERM	0x0018	6	Termination register 31 - 0
BIS6_IO_MUX	0x001C	7	Mux register 31 – 0
BIS6_IO_UCNTL	0x0020	8	Spare
BIS6_RES	0x0024	9	Reserved
BIS6_PLL_DATA	0x0028	10	PLL R/W port for FIFOs
BIS6_PLL_STATUS	0x002C	11	PLL programming status
BIS6_TEMP	0x0030	12	Temperature port
BIS6_IO_RDBK	0x0034	13	External I/O read register

SDLC Map

Port 0	0x1000	Pointer to base address for Port 0	
Port 1	0x4000	Pointer to base address for Port 1	
BIS6_SDLC_TX_MEM	0x0000	Dual-port TX RAM read/write port	
BIS6_SDLC_RX_MEM	0x1000	Dual-port RX RAM read/write port	
BIS6_SDLC_CNTL	0x2000	0	SDLC control port
BIS_SDLC_STAT	0x2000	0	SDLC control read-back with added status
BIS6_SDLC_CNTLB	0x2004	1	SDLC Master Interrupt enable port
BIS6_SDLC_STAT2	0x2008	2	SDLC Interrupt Status port

NRZ-L Map

Port 2	0x7000	Pointer to base address for Port 2
Port 3	0x7080	Pointer to base address for Port 3

BIS6_NRZL_CNTL	0x0000	NRZL Control port
BIS6_NRZL_CNTLB	0x0004	NRZL Master Interrupt Port
BIS6_NRZL_TXREG	0x0008	NRZL Tx Function Control Port
BIS6_NRZL_RXREG	0x000C	NRZL Rx Function Control Port
BIS6_NRZL_TXAMT	0x0010	NRZL Tx Almost Empty FIFO level
BIS6_NRZL_RXAFL	0x0014	NRZL Rx Almost Full FIFO level
BIS6_NRZL_TXCLK2X	0x0018	NRZL Tx transmit rate 2x reg
BIS6_NRZL_FIFO	0x001C	NRZL FIFO port – wr to Tx, rd from Rx
BIS6_NRZL_STAT	0x0020	NRZL status register [FIFO]
BIS6_NRZL_STAT2	0x0024	NRZL Interrupt Status port
BIS6_NRZL_RXCNTS	0x0028	NRZL Rx Pkt Cnt : Rx FIFO Cnt
BIS6_NRZL_TXCNTS	0x002C	NRZL Tx Pkt Cnt: Tx FIFO Cnt
BIS6_NRZL_PKT	0x0030	NRZL Packet FIFO wr to Tx, rd from Rx
BIS6_NRZL_TXGAP	0x0034	NRZL Tx Gap Timer definition
BIS6_NRZL_RXGAP	0x0038	NRZL Tx End of packet time definition

FIGURE 3 CCPMC-BISERIAL-VI-ORN1 INTERNAL ADDRESS MAP

The address map provided is for the local decoding performed within ccPMC-BiSerial-VI-ORN1. The addresses are all offsets from a base address, assigned by the system when the PCI bus is configured. The Base and Port offsets relative to the system assigned address are shown. The register and memory offsets are relative to the base and port offsets.

VendorId = 0xDCBA, CardId = 0x0071
Flash design ID = 0x0001

FLASH Revision:
1p2 : original release 7/5/22

Programming

Programming the PMC-BiSerial-VI-ORN1 requires only the ability to read and write data from the host. The base address of the module refers to the first user address for the slot in which the PMC is installed. This address is determined during system configuration of the PCI bus.

Depending on the software environment it may be necessary to set-up the system software with the PMC-BiSerial-VI-ORN1 "registration" data.

For SDLC, In order to receive data the software is only required to initialize the receiver buffer start address and enable the Rx port. To transmit the software will need to load the message into the appropriate memory, set the transmitter buffer start and end address and any configuration parameters and enable the transmitter.

When a received message completes, the end address of the message will be written to the receiver buffer start address with the received data stored starting with the next address. The next message will be stored starting with the following address unless a new starting address has been written after the first message has begun. The end address of each received message can also be read from the address field of the channel control register, but this will be over-written when the next message completes.

Once the transmitter starts sending a message, a new end address (and optionally a new start address) can be written to send subsequent messages. Multiple messages can be loaded into the transmitter RAM and sent in any order desired.

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

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

SDLC is the primary function of Ports 0 and 1. If not in use the IO allocated to the ports can be used as a parallel port by selection in the Mux register. Similarly, the NRZ-L ports can be used for the primary function or swapped out for parallel port operation.

Non-assigned IO [SDLC or NRZ-L functions] can also be used for parallel port operation.

Base Definitions

BIS6_BASE

[$00] Base Control Register Port read/write

Base Control Register

DATA BIT	DESCRIPTION
31	TestClockSel
30-18	Spare
17	PLL Use Alternate ID
16	PLL Check ID
15	PLL Read Enable
14	PLL Reset
13	PLL Enable
12-5	reserved
4	ForceInt
3-0	reserved

FIGURE 4 BASE CONTROL REGISTER BIT MAP

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

PLL Enable: When this bit is set to a one, the PLL programmer module, used to program and read the PLL, is enabled. When this bit is zero, the PLL programmer is disabled.

PLL Reset: When this bit is set to a one, the PLL programmer will stop processing, if not stopped already, and return to its initial state. When this bit is zero, the PLL programmer is ready to accept control inputs.

PLL Read Enable: When this bit is set to a one and the PLL programmer is enabled, the programmer will perform a read of the PLL device internal registers. The 40 bytes of data obtained will be written into the PLL read FIFO as ten long-words. When this bit is zero and the PLL programmer is enabled, the programmer will write data into the PLL device or simply check for a response to the selected ID value depending on the PLL Check ID control bit.

PLL Check ID: When this bit is set to a one and the PLL programmer is enabled, the programmer will begin a write operation, but will stop after the device ID has been sent. If the ID was acknowledged successfully, the done status will be set and the error status will be cleared. If the ID was not acknowledged successfully, the done status will be cleared and the error status will be set. When this bit is zero and the PLL programmer is enabled, the PLL programmer will perform a write or read operation depending on the PLL Read Enable control bit.

PLL Use Alternate ID: When this bit is set to a one, the device ID sent will be the alternate ID: 0x6A. When this bit is zero, the normal ID: 0x69 will be sent to the PLL device.

TestClockSel when '1' selects having a test clock output on the IO. The Mux register and Direction registers will also need to be set to transmit and non-state-machine use to have the clock output. Approximately 5 MHz. For IO and termination testing. Not used in normal operation. When '0' standard operation with parallel port or programmed IO is applied to the IO based on the Mux settings.

ForceInt: When '1' this bit forces an interrupt request. This feature is useful for testing and software development. Note: requires the Master Interrupt Enable to be set [enabled].

BIS6_BASE1

[$04] Base Control Expansion Port read/write

Base1 Control Register

DATA BIT	DESCRIPTION
31-1	Spare
0	Interrupt Enable Master

FIGURE 5 BASE CONTROL REGISTER BIT MAP

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

Interrupt Enable Master: When '1' allows interrupts generated by ccPMC-BiSerial-VI-ORN1 to be driven onto the carrier (INTA). When '0' the interrupts can be individually enabled and used for status without driving the backplane. Polled operation can be performed in this mode. Please note: additional enables may be present in each port.

BIS6_STATUS

[$08] Status Port read only

Design Number / FLASH Revision

DATA BIT	DESCRIPTION
31	InterruptNoMask
30-12	reserved
11	PortInt3
10	PortInt2
9	PortInt1
8	PortInt0
7-5	reserved
4	ForceInt
3-1	reserved
0	InterruptMasked

FIGURE 5 DESIGN ID REGISTER BIT MAP

InterruptNoMask is set when a Port Interrupt or Force Interrupt is active. This bit can be used for polling if the Base Level Master Interrupt Enable is disabled.

InterruptMasked is set when any of the interrupt sources is active and the Master Interrupt Enable is also enabled.

ForceInt is set when the ForceInt bit in the Base Control register is set.

PortInt0 is set when Port 0 SDLC is requesting an interrupt.
PortInt1 is set when Port 1 SDLC is requesting an interrupt.
PortInt2 is set when Port 2 NRZL is requesting an interrupt.
PortInt3 is set when Port 3 NRZL is requesting an interrupt.

With any of these bits the associated port Interrupt Status should be read to determine the cause of the interrupt.

With the Windows driver this port is read when the system detects an interrupt potentially from this device. Based on the Port and Force interrupt status the secondary port interrupt or force interrupt routine handlers are launched.

BIS6_ID

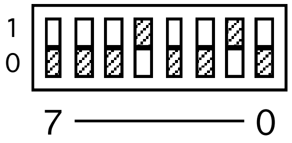
[$0C] Revision & Switch Port read only

User Switch Port

DATA BIT	DESCRIPTION
31-24	Design Type
23-16	FLASH Major Revision
15-8	FLASH Minor Revision
7-0	SW7-0

FIGURE 6 REVISION AND SWITCH PORT

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



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

BIS6_IO_DATA

[$10] Parallel Data Output Register read/write

Parallel Data Output Register

DATA BIT	DESCRIPTION
31-0	parallel output data

FIGURE 7 PARALLEL OUTPUT DATA BIT MAP

There are 32 potential output bits in the parallel port. The Direction, Termination, and Mux Control registers are also involved. When the direction is set to output, and the Mux control set to parallel port the bit definitions from this register are driven onto the corresponding parallel port lines.

This port is direct read/write of the register. The I/O side is read-back from the BIS6_IO_RDBK port. It is possible that the output data does not match the I/O data in the case of the Direction bits being set to input or the Mux control set to state-machine.

BIS6_IO_DIR

[$14] Direction Port read/write

Direction Control Port

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

FIGURE 8 DIRECTION CONTROL PORT

When set (‘1’) the corresponding bit in the parallel port is a transmitter. When cleared (‘0’) the corresponding bit is a receiver. The corresponding Mux control bits must also be configured for parallel port.

BIS6_IO_TERM

[$18] Termination Port read/write

Termination Control Port

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

FIGURE 9 TERMINATION CONTROL PORT

When set (‘1’) the corresponding I/O line will be terminated. When cleared (‘0’) the corresponding I/O line is not terminated. These bits are independent of the Mux control definitions. When a bit is set to be terminated; the analog switch associated with that bit is closed to create a parallel termination of approximately 100 Ω. In most systems the receiving side is terminated, and the transmitting side is not. These bits are not controlled by the state machines as some systems terminate in the cable.

BIS6_IO_MUX

[$1C] Mux Port read/write

Multiplexor Control Port

DATA BIT	DESCRIPTION
31-0	Parallel Port Mux Control bits

FIGURE 10 MUX CONTROL PORT

When set (‘1’) the corresponding bit is set to State-Machine control. When cleared (‘0’) the corresponding bit is set to parallel port operation. The Mux control definition along with the Data, Direction and Termination registers allows for a bit-by-bit selection of operation under software control. To use the defined ports [SDLC, NRZL] set to 0x3FFFF.

BIS6_IO_UCNTL

[$20] Upper Control Port read/write

Upper Bits Control Port

DATA BIT	DESCRIPTION
25-24	Mux 33, 32
17-16	Termination 33, 32
9-8	Direction 33, 32
1-0	Data 32, 32

FIGURE 11 UPPER CONTROL PORT

Unused for ccPMC models. PMC-BiSerial-VI has 34 transceivers. The upper control bits are concentrated within this register to cover the top 2 bits not controlled within the other control registers. The upper bits are only useable on the Bezel I/O connector. Pn4 has only 64 connections and doesn’t support the upper lines. The definitions are the same as the Data, Term, Dir and Mux port definitions for bit operation.

Data = Data transmitted when the Mux is set to ‘0’ and the direction is set to ‘1’. Termination when set to ‘1’ causes the parallel termination to be engaged. Setting the Mux control bits to ‘0’ creates a parallel port for those bits. Setting the Mux control bits to ‘1’ enables the state-machine to control the direction and data lines. The termination control is independent.

BIS6_IO_RDBKUPR

[$24] Upper Read-Back Port read only

I/O Upper Read-Back Port	
DATA BIT	**DESCRIPTION**
1-0	I/O Data 33-32

FIGURE 12 UPPER I/O READBACK PORT

Unused for ccPMC models. The I/O lines can be read at any time. The value is not filtered in any way. If the transceivers are set to TX by the parallel port or state-machine then the read-back value will be the transmitted value. If the transceivers are set to receive then the port values will be those received by the transceivers from the external I/O. The upper bits are presented on this port.

BIS6_PLL_DATA

[0x28] PLL Output FIFO Write/ PLL Input FIFO Read

PLL Output/Input FIFO Ports

Data Bit	Description
31-0	FIFO data word

FIGURE 13 PLL FIFO PORT

Writes to this port load PLL programming data into the PLL TX FIFO. This data is used to configure the PLL device. Reads from this port return data from the PLL RX FIFO. This data is the PLL device's internal register data that was read by the PLL programmer. Both FIFOs are 32 words deep and 32 bits wide.

The PLL is a separate device controlled by the Xilinx. The PLL has a fairly complex programming requirement which is simplified by using the Cypress® CyberClocks utility, and programming the resulting control words into the PLL using this PLL Control port. The interface can be further simplified by using the Dynamic Engineering driver to take care of the low-level bit manipulation requirements.

BIS6_PLL_STATUS

[0x2C] Base Interrupt Status – (read only)

Base Interrupt Status Register

Data Bit	Description
31-11	Spare
10	PLL Error
9	PLL Done
8	PLL Ready
7	Spare
6	PLL Read FIFO Data Valid
5	PLL Read FIFO Full
4	PLL Read FIFO Empty
3	Spare
2	PLL Write FIFO Data Valid
1	PLL Write FIFO Full
0	PLL Write FIFO Empty

FIGURE 14 PCIE-HOTLINK BASE STATUS PORT

PLL Write/Read FIFO Empty: When a one is read, it indicates that the corresponding FIFO contains no data; when a zero is read, there is at least one word in the FIFO. Although the FIFO is empty, there may still be one valid data word in the pipeline. The FIFO data valid bit indicates whether this is the case.

PLL Write/Read FIFO Full: When a one is read, it indicates that the corresponding FIFO is full; when a zero is read, there is room for at least one word in the FIFO.

PLL Write/Read FIFO Data Valid: When a one is read, there is valid data available; when a zero is read, there is no valid data available.

PLL Ready: When a one is read, the PLL programmer is idle and ready to accept a new command; when a zero is read, the programmer is actively sending data or reading data to/from the PLL device.

PLL Done: When a one is read, the programmer has successfully completed an input or output request; when a zero is read, this is not the case. This bit is latched and must be cleared by writing the PLL done bit back to this register.

PLL Error: When a one is read, an error occurred while processing a read or write request; when a zero is read, no error occurred. This bit is latched and must be cleared by writing the PLL error bit back to this register.

BIS6_TEMP

[$30] Temperture Port

Temperature Port

DATA BIT	DESCRIPTION
31-24	Spare
23-8	Lm75WriteData
7-5	Lm75Pointer [x80 = PTR only]
4	Lm75Read
3-1	spare
0	Lm75Write

FIGURE 15 PLL CONTROL

LM75B is a 400 KHz. I2C device. The 32 MHz reference is used to create an 80x reference to the controller. Write a null pointer to initialize [x81]. The Write and Read bits are auto cleared when the operation is complete. After the initial write completes do a dummy read of the data [x10]. Once the read bit is cleared repeat for data. See reference SW for an example. We use a flag to go through the initialization cycle once. Data is read back in the field shown. After shifting down the data is byte swapped to be in the proper order. Test the sign bit to see if a negative number.

0.125 C is the bit value.

BIS6_IO_RDBK

[$34] Read-Back Port read only

I/O Read-Back Port

DATA BIT	DESCRIPTION
31-0	I/O Data 31-0

FIGURE 16 I/O READBACK PORT

The I/O lines can be read at any time. The value is not filtered in any way. If the transceivers are set to TX by the parallel port or state-machine the read-back value will be the transmitted value. If the transceivers are set to receive the port values will be those received by the transceivers from the external I/O.

SDLC Definitions

BIS6_SDLC_CNTL

[$2000] SDLC Control/Status Register

SDLC Control/Status Register

DATA BIT	**DESCRIPTION**
31	Idle Detected/Clear (see note after description)
30	Abort Detected/Clear (see note after description)
29-25	spare
24	SDLC Internal Clock Select
23	Send an Abort (write only)
22	Load Transmit End Address (write only)
21	Load Transmit Start Address/SDLC Done
20	Load Receive Start Address/SDLC Sending Data
19	SDLC Idle After Frame Done
18-8	Address Input/ Receive End Address
7	Repeated Flags Share Zero
6	Received Abort Interrupt Enable
5	Receive Interrupt Enable
4	Transmit Frame Done Interrupt Enable
3	Transmit Interrupt Enable
2	Transmit Clear Enable
1	Receive Enable
0	Transmit Enable

FIGURE 17 SDLC CONTROL/STATUS REGISTER

Transmit Enable: When this bit is a one the transmitter is enabled to send data starting with the address stored in the transmitter start-address register and continuing until the data at the address in the transmitter end-address register has been sent. When this bit is a zero the transmitter is disabled.

Receive Enable: When this bit is a one the receiver is enabled to receive data and store it in the dual-port RAM starting with the address stored in the receiver start-address register if it is the first message since the receiver was enabled, or in the next 16-bit address after the end-address of the last message if it is not. When this bit is a zero the receiver is disabled.

Transmit Clear Enable: When this bit is a one the transmit enable bit will be cleared when the transmitted message completes and there is not another message pending. When this bit is a zero the transmitter will remain enabled, but no more data will be sent until a new end address is loaded.

Transmit Interrupt Enable: When this bit is a one the transmitter interrupt is enabled. The interrupt will occur at when the transmit state-machine reaches the end address stored in the transmitter end-address register and there is not another message pending. When this bit is a zero the interrupt status will still be latched, but will not cause an interrupt to occur. The transmit interrupt is mapped to the first interrupt line in its channel block.

Transmit Frame Done Interrupt Enable: When this bit is a one the transmit frame done interrupt is enabled. This interrupt will occur when each message frame completes regardless of whether another message is pending. When this bit is a zero the interrupt status will still be latched, but will not cause an interrupt to occur. The transmit frame done interrupt is mapped to the second interrupt line in its channel block.

Receive Interrupt Enable: When this bit is a one the receiver interrupt is enabled. The interrupt will occur at the end of a message transmission, which is determined by the detection of a SDLC flag character (0x7e) after the message has started. When this bit is a zero the interrupt status will still be latched, but will not cause an interrupt to occur. The receive interrupt is mapped to the third interrupt line in its channel block.

Received Abort Interrupt Enable: When this bit is a one, the received abort interrupt is enabled. This interrupt will occur when an SDLC abort character (0x7f) is received. When this bit is a zero the abort interrupt status will still be latched, but will not cause an interrupt to occur. The received abort interrupt is mapped to the fourth interrupt line in its channel block.

Repeated Flags Share Zero: When this bit is a one and the transmitter is sending repeated flag characters, the last zero in each flag will also serve as the first zero in the next flag. This is only true for two successive flags, the last flag before data is sent will be sent entirely. When this bit is a zero, all eight bits of each flag will be sent regardless of adjacent characters.

Address Input/Receive End Address: This field is used with the three load address bits to specify address boundaries for the transmitter and receiver data buffers. When this field is read, it represents the address in which the last received data word from the last message-frame is stored. Note that this is a 16-bit address, bit 0 indicates which half of the appropriate long-word the last 16-bit word was stored (0 -> lower half, 1 -> upper half).

SDLC Idle After Frame Done: When this bit is a one, the SDLC link will go to the idle state (minimum of 15 consecutive ones) when message transmission completes. The link will remain high until a new message is requested. When this bit is zero and the transmitter remains enabled, the transmitter will send repeated flags until a new message is requested.

Load Receive Start Address/SDLC Sending Data: When this bit is a one the value in the address input field is loaded into the receiver start-address register. When this bit is a zero no action is taken. When this bit is read as a one, the transmitter is actively sending data. At this time new addresses can be written for the next message-frame to be sent. A new transmitter end address is required to queue a new message-frame. New transmit or receive start addresses are optional. If new start addresses are not written, the transmitter and/or receiver will continue reading/storing data at the next address after the end address of the last message frame. When this bit is a zero, the link is either idle, aborted or sending repeated flags.

Load Transmit Start Address/SDLC Frame Done: When this bit is a one the value in the address input field is loaded into the transmitter start-address register. When this bit is a zero no action is taken. When this bit is read as a one, it indicates that the last message has completed. This bit is latched and will be cleared by any write to this control register. An interrupt can be configured to occur when this bit goes high by asserting the transmit frame done interrupt enable. When this bit is read as a zero, a message-frame has not completed since the last write to the SDLC control register.

Load Transmit End Address (write only): When this bit is a one the value in the address input field is loaded into the transmitter end-address register. When this bit is a zero no action is taken.

Send an Abort (write only): When this bit is set to a one the transmit state-machine will send an abort character (0xfe) provided a transmission is currently in progress. When this bit is a zero normal operation will continue.

SDLC Internal Clock Select: When this bit is a one, the transmitter will use the internal transmit clock as a reference for sending SDLC data. When this bit is a zero, an external received clock will be used as the reference for data transmission. *If using external clock with loop-back be sure to connect the reference clock IO and set the control Mux to enable the reference clock outputs.*

Abort Detected/Clear: When an abort character is detected by the receiver, this status bit will be latched and can be cleared by writing a one back in this bit position. When this bit is a zero, no abort has been detected since the latch was last cleared.

Idle Detected/Clear: When an idle bus state is detected by the receiver, this status bit will be latched and can be cleared by writing a one back in this bit position. When this bit is a zero, the bus has not idled since the latch was last cleared.

Note: Writing Abort Clear or Idle Clear disables updating any other bits in the control register. When these latched bits are cleared it must be the only action performed with that register access.

BIS6_SDLC_CNTLB

[$2004] Control Register Expansion

SDLC Master Interrupt Control

DATA BIT	DESCRIPTION
31-2	Spare
1	ForceInt
0	Master Interrupt Enable

FIGURE 18 SDLC CONTROL EXPANSION REGISTER

Master Interrupt Enable when set gates the interrupts from the associated SDLC port to the device level [base] interrupt processing. When ‘0’ the status bits can be used to poll and no interrupt will be generated. Please note: the individual interrupt enables also need to be set.

ForceInt when set will cause an interrupt request to be generated. This bit is masked by the Master Interrupt Enable for the port and device.

BIS6_SDLC_STAT2

[$2008] SDLC Interrupt Status Register

Interrupt Status and Clear Register

DATA BIT	DESCRIPTION
31	InterruptActive
30-5	Spare
4	ForceInt
3	Received Abort Interrupt Latched
2	Rx Interrupt Latched
1	Transmit Frame Done Latched
0	Tx Interrupt Latched

FIGURE 19 SDLC INTERRUPT STATUS REGISTER

Please refer to the Control register definitions for the Latched interrupts. ForceInt is set when set in the BIS6_SDLC_CNTLB. Latched bits are cleared by writing to this port with the corresponding bit set.

BIS6_SDLC_MEM

Each SDLC port has 2 Dual Port RAM implementations. The host side of the memory is 32 bits wide and 1K deep – 4K bytes. The SDLC base address is the starting address for the TX memory. The Rx memory is the same size and starts offset by the 4K bytes x1000.

The Tx SDLC function reads from the internal port of the TX memory. The Rx SDLC function writes to the RX internal port.

The memory is accessible via target read and write operations from both ports. It is permissible to write and then read from the Tx memory or Rx memory spaces. The ATP uses this feature as part of testing the hardware.

See the Description for SDLC operation for a discussion of how the memory operates and interacts with the SDLC function.

See the SDLC base register to set the TX and RX operational addresses. The same field is used for both with control bits selecting which address is being loaded.

The SDLC transfer is in terms of 16 bit words. In cases of non-LW aligned data lengths the Start and End addresses are encoded with the last word to send in bit 0. The programmed addresses are word addresses. For example x7FE would be the full memory of words.

The Windows driver supports multiple reads and writes by passing a structure with the offset, array and length to either write or read. The driver also supports single word accesses. See the driver manual for more information.

NRZ-L Definitions

BIS6_NRZL_CNTL

[$0000] Control Register

NRZL Control Register

DATA BIT	DESCRIPTION
31-2	Spare
1	FIFO Loopback
0	NRZL Reset

FIGURE 20 NRZL CONTROL REGISTER

Set NRZL Reset to reset the FIFOs [Data and Packet] plus reset the state-machines. Control registers are not reset. Clear for normal operation.

Set FIFO Loopback to cause data in the TX data FIFO to loop to the RX data FIFO. Flow control is employed. TX and RX state-machines should be disabled to prevent conflicts when using this mode. Clear for normal operation. Data transferred to the RX FIFO is no longer available to transmit.

BIS6_NRZL_CNTLB

[$0004] Control Register Expansion

NRZL Master Interrupt Control

DATA BIT	DESCRIPTION
31-2	Spare
1	ForceInt
0	Master Interrupt Enable

FIGURE 21 NRZL CONTROL EXPANSION REGISTER

Master Interrupt Enable when set gates the interrupts from the associated NRZL port to the device level [base] interrupt processing. When ‘0’ the status bits can be used to poll and no interrupt will be generated. Please note: the individual interrupt enables also need to be set.

ForceInt when set will cause an interrupt request to be generated. This bit is masked by the Master Interrupt Enable for the port and device.

BIS6_NRZL_TXCNTL

[$0008] Tx Control Register

NRZL Tx Specific Control

DATA BIT	DESCRIPTION
31-5	Spare
4	Tx Interrupt Enable
3	Tx Clk Inv
2	Tx Data Inv
1	Tx MsbLsb
0	Tx Enable

FIGURE 22 NRZL TX CONTROL REGISTER

Tx Enable when set ‘1’ enables the transmitter to send data. The state machine will wait for the Data FIFO to be not empty and for the Tx Packet FIFO to have a descriptor before sending data. Set to ‘0’ when using FIFO loop-back mode.

Tx MsbLsb determines the order of the data transmitted. For MSB first operation set this bit. For LSB first leave cleared. See Data and Packet FIFO definitions for more on the order of the bits – how to load and how to define the length.

Tx Data Inv when set inverts the stored data bit by bit for transmission. When cleared data is transmitted as stored [not inverted].

Tx Clk Inv when set changes the sense of the clock to be falling edge stable. When Cleared the rising edge is used by the receiver. Transmitted data is very close to 50-50 duty cycle. Change on the falling edge, stable on the rising or the opposite based on this bit.

Tx Interrupt Enable when set gates the latched status for the end of transmission to be gated through to the master interrupt enable stage of interrupt generation. When cleared the status can still be read for polling but does not generate an interrupt request. Set each time a packet is completed [last bit sent].

BIS6_NRZL_RXCNTL

[$000C] Rx Control Register

NRZL Rx Specific Control

DATA BIT	DESCRIPTION
31-5	Spare
4	Rx Interrupt Enable
3	Rx Clk Inv
2	Rx Data Inv
1	Rx MsbLsb
0	Rx Enable

FIGURE 23 NRZL RX CONTROL REGISTER

Rx Enable when set ‘1’ enables the receive to capture data. The state machine will wait edges to be detected, capture bits and store words. When a packet is completed based on the programmed gap time the current descriptor is stored. Set to ‘0’ when using FIFO loop-back mode.

Rx MsbLsb determines the order of the data received. For MSB first operation set this bit. For LSB first leave cleared. See Data and Packet FIFO definitions for more on the order of the bits – how to read and interprete.

Rx Data Inv when set inverts the received data bit by bit. When cleared data is stored as received [not inverted].

Rx Clk Inv when set changes the sense of the clock to be falling edge stable. When cleared the rising edge is used by the receiver. A reference clock based sampling state-machine is used to detect edges and store the received bits. This bit selects the edge the detector is searching for.

Rx Interrupt Enable when set gates the latched status for the end of reception to be gated through to the master interrupt enable stage of interrupt generation. When cleared the status can still be read for polling but does not generate an interrupt request. Set each time a packet is completed [last word stored].

BIS6_NRZL_TXAMT

[$0010] Tx Almost Empty

NRZL Tx Almost Empty

DATA BIT	DESCRIPTION
31-16	Spare
15-0	Tx Almost Empty

FIGURE 24 NRZL TX ALMOST EMPTY

Tx Almost Empty sets the count for the Tx Data FIFO Almost Empty comparison. The count is used as a “less than” comparison. If set to x10 when the FIFO is x0F and lower the bit will be set. The status is in the Status Register. Full width register with 16 bits assigned to the Almost Empty function to match the depth of the FIFO.

BIS6_NRZL_RXAFL

[$0014] Rx Almost Full

NRZL Rx Almost Full

DATA BIT	DESCRIPTION
31-16	Spare
15-0	Rx Almost Full

FIGURE 25 NRZL RX ALMOST FULL

Rx Almost Full sets the count for the Rx Data FIFO Almost Full comparison. The count is used as a “Greater than” comparison. If set to x400 when the FIFO is x401 and higher the bit will be set. The status is in the Status Register. Full width register with 16 bits assigned to the Almost Full function to match the depth of the FIFO.

BIS6_NRZL_TXCLK2X

[$0018] Tx Clock Rate x2

NRZL Tx Clock Rate

DATA BIT	DESCRIPTION
31-16	Spare
15-0	Tx Clock Rate x2, Program with N

FIGURE 26 NRZL TX CLOCK RATE

TX Clock Rate is used to set the state machine rate for the transmitter. It is set to 2x the desired clock rate – for example setting to 10 MHz provides 5 MHz clock at the transmitter. The PLLC rate is counted from 0 to N in a free running manner. A pulse is generated for each loop. The pulse is used to control shifting, loading, and clock output generation. Ref Freq / N+1 = 2x the desired frequency. With a 100 MHz reference on PLLC and a divisor of 9 the reference pulse rate will be 10 MHz and the output clock rate 5 MHz. Output clock is burst mode – only present when data is being transferred.

BIS6_NRZL_FIFO

[$001C] Data FIFO

NRZL FIFO

DATA BIT	DESCRIPTION
31-0	FIFO DATA

FIGURE 27 NRZL DATA FIFO

Writing to the Data FIFO address stores data to transmit. Reading from the address retrieves data stored from reception.

Transmitted and Received data are both stored based on the packet descriptor. In the Tx case, the host provides the descriptor. In the Rx case the HW generates the descriptor and stores into the Rx Packet FIFO to let the host know how much data to read.

Descriptors are bit lengths. Data is stored LWs first Remainder last. The remainder is LSB aligned for both Tx and Rx whether LSB first or MSB first operation, The HW automatically picks off the apparent MSB for the remainder and sends the last bits starting from there[MSB mode]. The receiver automatically shifts the last word down to be LSB aligned [LSB mode].

LW
LW
LW
Remainder LSB aligned

Any number of bits can be sent within the bit count provided by the descriptor. See the packet FIFO discussion for more on this topic.

The Tx and Rx FIFOs are separate and 16K-1 x 32 bits each.

BIS6_NRZL_STATUS

[$0020] NRZL STATUS Register

NRZL Status Register

DATA BIT	DESCRIPTION
31-14	'0'
13	RX SM IDLE
12	TX SM IDLE
11	RX PKT FIFO FULL
10	RX PKT FIFO MT
9	TX PKT FIFO FULL
8	TX PKT FIFO MT
7	'0'
6	RX FIFO FULL
5	RX FIFO AFL
4	RX FIFO MT
3	'0'
2	TX FIFO FULL
1	TX FIFO AMT
0	TX FIFO MT

FIGURE 28 NRZL DATA FIFO

Bits marked '0' will return '0' when read. These bits are spare for future additions.

TX FIFO MT = '1' when the TX Data FIFO is Empty. Otherwise '0'.
TX FIFO AMT = '1' when the TX Data FIFO count is below the programmed TX AMT count. Otherwise '0'.
TX FIFO FULL = '1' when the TX Data FIFO is FULL. Otherwise '0'.

RX FIFO MT = ‘1’ when the RX Data FIFO is Empty. Otherwise ‘0’.
RX FIFO AFL = ‘1’ when the RX Data FIFO count is above the programmed RX AFL count. Otherwise ‘0’.
RX FIFO FULL = ‘1’ when the RX Data FIFO is FULL. Otherwise ‘0’.

TX PKT FIFO MT = ‘1’ when the TX Packet FIFO is Empty. Otherwise ‘0’.
TX PKT FIFO FULL = ‘1’ when the TX Packet FIFO is FULL. Otherwise ‘0’.

RX PKT FIFO MT = ‘1’ when the RX Packet FIFO is Empty. Otherwise ‘0’.
RX PKT FIFO FULL = ‘1’ when the RX Packet FIFO is FULL. Otherwise ‘0’.

TX and RX SM IDLE bits are set when the respective state machines are in the IDLE state. Can be polled to know when a disabled SM is back to the idle state for further processing. Alternatively, disable and then reset the port to force back to the idle state.

BIS6_NRZL_STAT2

[$0024] NRZL Interrupt Status Register

Interrupt Status and Clear Register

DATA BIT	DESCRIPTION
31	InterruptActive
30-6	Spare
5	ForceInt
4	Rx Packet Over Flow Latched
3	Rx Data Over Flow Latched
2	Tx Under Run Latched
1	Rx Interrupt Latched
0	Tx Interrupt Latched

FIGURE 29 NRZL INTERRUPT STATUS REGISTER

Latched bits are cleared by writing to this port with the corresponding bit set.

Tx Interrupt is set each time the Tx state-machine finishes processing a descriptor. The bit is latched.

Rx Interrupt is set after loading the descriptor for a reception. The end of the packet is determined by the lack of a clock transition within the programmed Rx Gap time.

Tx Under Run is set if the transmitter does not have enough data to complete a descriptor.   Data can be loaded on the fly – however it must be in the FIFO when time to read that word to send.

Rx Data Over Flow is set if the Rx Data FIFO is full when it is time to write more data.

Rx Packet Over Flow is set if there is no room to write the new descriptor when ready to load.

The UnderRun and OverFlow error bits should be monitored and corrected if they occur. If under running – preload more data before starting [loading the descriptor]. If over flowing read more often or make use of the burst read utility provided in the SW package.

Force Int is set when the Force Int control bit is set to allow a single read to determine the cause(s) of an interrupt from the port. Clear this bit in the control register.

BIS6_NRZL_RXCNTS

[$0028] Rx FIFO Counts

NRZL Rx FIFO Counts

DATA BIT	DESCRIPTION
31-16	Rx Packet FIFO Count
15-0	Rx Data FIFO Count

FIGURE 30 NRZL RX FIFO COUNTS

Rx Data FIFO Count field returns the number of LW in the Rx data FIFO. Use associated Packet FIFO Descriptor to crack the packed data into complete LW and remainder for non-longword aligned transfers. 0x3FFF is the current max count.

Rx Packet FIFO Count is the number of descriptors stored into the Rx Packet FIFO. 0x3FF is the current max count.

BIS6_NRZL_TXCNTS

[$002C] Tx FIFO Counts

NRZL Tx FIFO Counts

DATA BIT	DESCRIPTION
31-16	Tx Packet FIFO Count
15-0	Tx Data FIFO Count

FIGURE 31 NRZL TX FIFO COUNTS

Tx Data FIFO Count field returns the number of LW in the Tx data FIFO. 0x3FFF is the current max count.

Tx Packet FIFO Count is the number of descriptors stored into the Tx Packet FIFO. 0x3FF is the current max count.

BIS6_NRZL_PKT

[$0030] Packet FIFO

NRZL Packet FIFO

DATA BIT	**DESCRIPTION**
31-29	“0”
28-5	Descriptor LW count
4-0	Descriptor Remainder

FIGURE 32 NRZL PACKET FIFO

Writing to the Packet FIFO address stores the Descriptor to program the length to transmit. Reading from the address retrieves the descriptor for the data stored in the Data FIFO.

If the transmitter is enabled and data is stored the act of writing the descriptor will also launch the transmit state-machine.

Descriptors are bit lengths. The D4-0 represent the remainder – the part of the message sent that is not on a LW boundary. 0x20 would be 1 LW or 32 bits sent. 1 LW loaded into the Data FIFO.

LW D31-0

For LSB or MSB data shorter than 1 LW the descriptor would be 0x1F – 0x1. The data will be LSB aligned. The upper portion is not sent and can be anything. Padding with ‘0’ may provide some benefits when tracing.

For more than 1 LW in length the data is stored with at least 2 LW with the complete LW sent first and the remainder sent last.
LW
…
LW
Remainder LSB aligned

The Tx and Rx Packet FIFOs are separate and 1K-1 x 32 bits each.

Example: 3 LW loaded, LSB first

```
76543210
FEDCBA98
13121110
```

The Descriptor would be 3 x 32 bits = x60
Data would be transmitted ‘b 0000 1000 0100 1100 0010 1010 0110 1110…
Gaps are for ease of reading and not present in the data stream

Example: 16 bits to send MSB

0x0000ABCD loaded into Data FIFO

Descriptor = 16 bits = 0x10
Data transmitted 1010 1011 1100 1101

Gaps are for ease of reading and not present in the data stream

Reading a Descriptor tells the host what to do with the data. Divide the descriptor by 32 and read the integer value in LW. The remainder is the portion of the next LW with valid data. Alternatively, the Host can read the FIFO count and move data to host memory using the descriptors to break into separate messages after. With multiple small packets this can be a more efficient method.

Note: Data in the Data FIFO is separated by packet. Messages are always read [Tx] stored [Rx] starting on a LW boundary [bit wise].

BIS6_NRZL_TXGAP

[$0034] Tx Gap Register

NRZL Tx Gap Register

DATA BIT	DESCRIPTION
31-24	“0”
23-0	Programmed time between packets

FIGURE 33 NRZL TX GAP

The Tx Gap register is used to program the time between packets sent. If 0x00 the state-machine skips the gap time. The time is counted by the clock enables from the 2X clock. If programmed to 10 MHz as in the previous example the delay would be 100 nS per count.

The full register is present and can be read back. This register can be changed at any time as the value is synchronized to the reference clock [PLLC] before comparing against the local count. If the count is decreased during while a gap is being counted the counter may have already gone past the count and have to loop around 1 time before the following gaps are operating with the programmed time. Best to change when Tx is in the Idle state.

The Tx GAP time is best used when multiple packets of data are loaded and multiple packet descriptors are to be loaded with the Tx GAP time between them.

BIS6_NRZL_RXGAP

[$0038] Rx Gap Register

NRZL Rx Gap Register

DATA BIT	**DESCRIPTION**
31-24	“0”
23-0	Programmed Packet Timeout

FIGURE 34 NRZL RX GAP

The Rx Gap register is used to program the time to wait before declaring end of packet. The active edge of the received clock is used to capture data and to reset a timer. If the time gets to the programmed time “end of Packet” is declared. The bits shifted into position [if LSB first] and the remainder if any loaded [if any] into the Data FIFO.

The timer is using the PLLC reference – 100 MHz or the programmed rate if changed. If using a 5 MHz data stream the period of the clock received is 200 nS. The gap timer has an infinite wait for the 1st bit so it does not timeout before any data comes. Once the clocks are received the timer runs being reset with each bit. The time should be longer than the period of the expected data and shorter than the time between packets if multiple packets are expected. The time will also delay writing the descriptor allowing the host to process the stored data. Something like 2x the expected period allows some jitter on the received clock and not an excessive amount of added time before completing the packet.

Examples of using the timers are located in the reference SW available for this design.

Port I/O Line Mapping

ccPMC model is routed to Pn4

SDLC Port 0:

SDLC transmit data => I/O 0:	pin 1 +,	pin 3 –
SDLC receive data => I/O 1:	pin 2 +,	pin 4 –
SDLC transmit clock=> I/O 2:	pin 5 +,	pin 7 –
SDLC receive clock => I/O 3:	pin 6 +,	pin 8 –

SDLC Port 1:

SDLC transmit data => I/O 4:	pin 9 +,	pin 11 –
SDLC receive data => I/O 5:	pin 10 +,	pin 12 –
SDLC transmit clock=> I/O 6:	pin 13 +,	pin 15 –
SDLC receive clock => I/O 7:	pin 14 +,	pin 16 –

NRZ-L Port 2:

PCM transmit data => I/O 8:	pin 17 +,	pin 19 –
PCM receive data => I/O 9:	pin 18 +,	pin 20 –
PCM transmit clock => I/O 10:	pin 21 +,	pin 23 –
PCM receive clock => I/O 11:	pin 22 +,	pin 24 –

PCM Port3 3:

PCM transmit data => I/O 12:	pin 25 +,	pin 27 –
PCM receive data => I/O 13:	pin 26 +,	pin 28 –
PCM transmit clock => I/O 14:	pin 29 +,	pin 31 –
PCM receive clock => I/O 15:	pin 30 +,	pin 32 –

SDLC External Clock Test Resources

SDLC test clock 0 => I/O 30:	pin 61 +,	pin 63 –
SDLC test clock 1 => I/O 31:	pin 62 +,	pin 64 –

Interrupts

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

For example, the ccPMC-BiSerial-VI TX state machine(s) generates an interrupt request when a transmission is complete, and the TX int enable and Master interrupt enable bits are set. The transmission is considered complete when the last bit is output from the output shift register.

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

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

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

Loop-back

The Driver package has reference software, which includes an external loop-back test. The ORN1 version of ccPMC-BiSerial-VI utilizes Pn4 rear panel connector. Installed onto a test platform the IO are mapped to a SCSI connector enabling loop-back. The test requires an external cable with the following pins connected. Using HDEterm68 test fixture make the following connections (TP2 unless noted). The IO numbers match **Pn4** definitions later in the manual.
Note: TP1, 2 are both ordered as follows: 1, 35, 2, 36, 3, 37…32, 66, 33, 67, 34, 68.

SDLC SIGNAL	A	B	IO	A	B	IO
Port 0 Data TX to RX	1	35	0	2	36	1
Port 0 Clock TX to RX	3	37	2	4	38	3
Port 1 Data TX to RX	5	39	4	6	40	5
Port 1 Clock TX to RX	7	41	6	8	42	7

External TX Clock Distribution Network to Ports 0-1

SIGNAL	A	B	IO	A	B	IO
TX Test Clock 0	31	65	30	4	38	2,3
TX Test Clock 1	32	66	31	8	42	6,7

NRZ-L Signals	A	B	IO	A	B	IO
Port 2 Data Tx to Rx	9	43	8	10	44	9
Port 2 CLK Tx to Rx	11	45	10	12	46	11
Port 3 Data Tx to Rx	13	47	12	14	48	13
Port 3 CLK Tx to Rx	15	49	14	16	50	15

Loop-back is accomplished with HDEterm68 connected to PMC-UNIV-TEST.
The first HDEterm68 is connected as shown above.
A second test set-up with the lower 16 tied to the upper 16 [IO0-IO16 etc.] is used to test the parallel port.
A third HDEterm68 [open] is used along with the test clock to independently test the IO and terminations.
See the following links for the required devices:
https://www.dyneng.com/HDEterm68.html
https://www.dyneng.com/PMC-UNIV-TEST.html
https://www.dyneng.com/HDEcabl68.html

PMC PCI Pn1 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn1 Interface on ccPMC-BiSerial-VI. 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 35 PN1 INTERFACE

PMC PCI Pn2 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn2 Interface on ccPMC-BiSerial-VI. 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 36   PN2 INTERFACE

PMC IO Pn4 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn4 Interface on ccPMC-BiSerial-VI. See the User Manual for your carrier board for more information.

IO_0+	IO_1+	1	2
IO_0-	IO_1-	3	4
IO_2+	IO_3+	5	6
IO_2-	IO_3-	7	8
IO_4+	IO_5+	9	10
IO_4-	IO_5-	11	12
IO_6+	IO_7+	13	14
IO_6-	IO_7-	15	16
IO_8+	IO_9+	17	18
IO_8-	IO_9-	19	20
IO_10+	IO_11+	21	22
IO_10-	IO_11-	23	24
IO_12+	IO_13+	25	26
IO_12-	IO_13-	27	28
IO_14+	IO_15+	29	30
IO_14-	IO_15-	31	32
IO_16+	IO_17+	33	34
IO_16-	IO_17-	35	36
IO_18+	IO_19+	37	38
IO_18-	IO_19-	39	40
IO_20+	IO_21+	41	42
IO_20-	IO_21-	43	44
IO_22+	IO_23+	45	46
IO_22-	IO_23-	47	48
IO_24+	IO_25+	49	50
IO_24-	IO_25-	51	52
IO_26+	IO_27+	53	54
IO_26-	IO_27-	55	56
IO_28+	IO_29+	57	58
IO_28-	IO_29-	59	60
IO_30+	IO_31+	61	62
IO_30-	IO_31-	63	64

FIGURE 37 PN4 INTERFACE

Applications Guide

Interfacing

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

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

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

Power all system power supplies from one switch. Connecting external voltage to ccPMC-BiSerial-VI 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. ccPMC-BiSerial-VI does have transorbs for ESD and signal excursion protection.

Keep cables short. Flat cables, even with alternate ground lines, are not suitable for long distances. 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 ccPMC-BiSerial-VI 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 (HDEterm68). The terminal block can mount on standard DIN rails.

https://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. The transorb protection is meant to provide protection against transient events.

Construction and Reliability

ccPMC Modules are conceived and engineered for rugged industrial environments. The ccPMC-BiSerial-VI is constructed out of 0.062 inch thick High Temp FR4 material. The PC Boards are ROHS compliant. Dynamic Engineering has selected gold immersion processing to provide superior performance, and reliability (not subject to tin whisker issues).

Through hole and surface mounting of components are used.

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

The ccPMC is secured against the carrier with multiple screws attached to the 2 stand-offs and thermal interface locations. The screws provide significant protection against shock, vibration, and incomplete insertion.

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

Thermal Considerations

The ccPMC-BiSerial-VI 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. In a conduction cooled environment the thermal attachment points become very important.

ccPMC-BiSerial-VI has an internal floating thermal plane attached to the thermal rib locations. In addition, full plane ground and power planes will help keep the PCB at an even temperature – avoiding hot spots. It is up to the system to provide the thermal interface to provide the path for the thermal flux to move from ccPMC-BiSerial-VI into the cooling system of the chassis.

The devices utilized are rated at Industrial or better. It is recommended to provide adequate margin to allow for the thermal rise through the chassis to the PCB itself.

Warranty and Repair

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

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

Service Policy

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

Out of Warranty Repairs

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

For Service Contact:

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

Specifications

Host Interface:	(PMC) PCI Mezzanine Card - 32 bit, 33 MHz
Interface:	2 Full Duplex SDLC serial interfaces. 16-bit word size, LSB first. 2 NRZ-L full duplex interfaces. Programmable LSB/MSB first, bit count, clock sense, data sense. Parallel Port can be selected on a bit-by-bit basis.
TX Data rates generated:	40 MHz oscillator used to generate 48 MHz I/O clock sampling frequency (PLL clock A) and internal/external SDLC transmit clock test frequencies (PLL clock B). PLLC programmed to 100 MHz default. User changeable for other unique frequencies
RX Data rates accepted:	SDLC rates 1-3 MHz accepted. NRZ-L programmable via Tx Rate register and PLLC value.
Software Interface:	Control Registers, Status Ports, Dual Port RAM, Driver Available
Initialization:	Hardware reset forces all registers to 0.
Access Modes:	LW boundary Space (see memory map)
Wait States:	1 for all addresses
Interrupt:	SDLC: TX and Rx interrupts at end of message SDLC: TX interrupt at end of frame transmission SDLC: RX interrupt when abort received NRZL: TX and RX Packet complete interrupts Software interrupt
DMA:	Not implemented at this time
Onboard Options:	All Options are Software Programmable
Interface Options:	ccPMC depends on carrier options.
Dimensions:	Standard Single ccPMC Module.
Construction:	High temp. FR4 Multi-Layer Printed Circuit, Through Hole and Surface Mount Components.
Temperature Coefficient:	2.17 W/^{O}C for uniform heat across PMC
Power:	Max. **TBD** mA @ 5V
Temperature range	Industrial Temperature components standard (-40 + 85)

Order Information

ccPMC-BiSerial-VI-ORN1	ccPMC Module with 2 SDLC & 2 NRZ-L ports, 32 bit parallel port (overlaps with serial channels) RS-485 I/O. 32-bit data interface
Eng Kit–ccPMC-BiSerial-VI	HDEterm68 - 68 position screw terminal adapter https://www. dyneng. com/HDEterm68. html HDEcabl68 - 68 I/O twisted pair cable https://www. dyneng. com/HDEcabl68. html PMC-UNIV-Test – passive vertical adapter for PCI position with PMC installed. PCI test points, SCSI support for Pn4. Option for engineering unit or standard. https://www.dyneng.com/PMC-UNIV-TEST.html

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