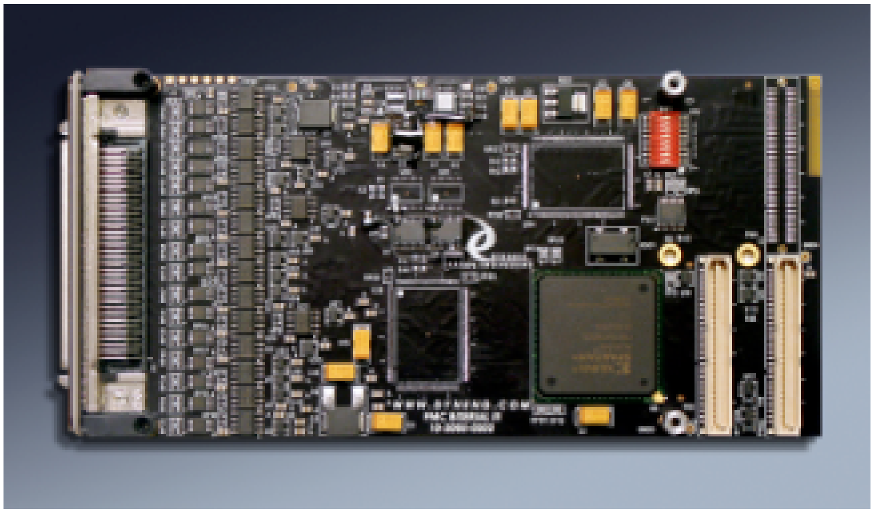
DYNAMIC ENGINEERING
150 DuBois St. Suite C, Santa Cruz, CA 95060
831-457-8891 Fax 831-457-4793
http://www.dyneng.com
sales@dyneng.com
Est. 1988

User Manual

PMC-BiSerial-III-ORB2

Digital Parallel Interface
PMC Module
34/32 programmable Differential IO
4 IO Protocols
LVDS and RS485



Revision B
Corresponding Hardware: Revision 3
10-2005-0204
FLASH 0903

PMC-BiSerial-III-ORB2
Digital Parallel Interface
PMC Module
Dynamic Engineering
150 DuBois St. Suite C, Santa Cruz CA 95060
831-457-8891 831-457-4793 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.

Manual Revision B. Revised 3/12/09

Table of Contents

List of Figures

Product Description

In embedded systems many of the interconnections are made with differential [RS-422/485 or LVDS] signals. Depending on the system architecture an IP or a PMC will be the right choice to make the connection. You have choices with carriers for cPCI, PCI, VME, PC/104p and other buses for both PMC and IP mezzanine modules.

Usually the choice is based on other system constraints as both the PMC and IP can provide the IO you require. Dynamic Engineering would be happy to assist in your decision regarding architecture and other trade-offs with the PMC / IP decision. Dynamic Engineering has carriers for IP and PMC modules for most architectures, and is adding more as new solutions are requested, and required by our clients.

The PMC compatible PMC-BiSerial-III has 34 independent differential IO available. The high density makes efficient use of PMC slot resources. The IO is available for system connection through the front panel [34], via the rear [Pn4] connector [32], or both. A high density 68 pin SCSI III front panel connector provides the front panel IO. The rear panel IO has a PIM and PIM Carrier available for rear panel wiring options.

PMC-BiSerial-III-ORB2 is a “clientized” version of the standard PMC-BiSerial-III board. “ORB2” is set to a combination of RS485 and LVDS standards, has front panel IO, and supports two channels each of 4 different standards. Each of the eight channels is programmable independently for frequency. The PLL is programmed for the base clock rate that is further divided by programmable down counters to provide the channel frequencies. COM1-4 operate in the range of 1-10 MHz. COM5,6 up to 2 MHz, and COM7 and COM8 use a separate PLL output and can operate up to 100 MHz. The PLL is referenced to 50 MHz. and can be programmed with new .JED files using the driver.

The channels are all half duplex and can be used in pairs for loop-back testing.
The HDEterm68 http://www.dyneng.com/HDEterm68.html
can be used as a breakout for the front panel IO. The HDEcabl68 provides a convenient cable. http://www.dyneng.com/HDEcabl68.html Custom cables can be manufactured to your requirements. Please contact Dynamic Engineering with your specifications.

Each COM is programmable to be input or output. Each channel has a separate DMA engine to support data transfers with the host. Each channel has separate memory using internal RAM for FIFO’s and in the case of COM7,8 external FIFO’s. The interrupts are maskable to allow polled operation as well. Terminations are automatically enabled based on the RX or TX function selected.

PLL output A is used as the reference to COM1-6. The divided version is used to run the TX state-machine. The PLL output is used to run the receiver. A clock detection circuit allows the receivers to operate without a continuous clock. With PLL A set to 140 MHz a 14:1 ratio on the receiver allows for a 10 MHz max rate on the Rx channels.

With several of the COM designs it is possible to have non 32 bit bounded data. A special AutoShift function is designed into the HW to automatically align the data in the "short" word to be on a 32 bit boundary.

All of the IO are routed through the FPGA to allow for custom applications that require hardware intervention or specific timing- for example an automatic address or data strobe to be generated. The initial model was register based [FLASH 0101]. Please contact Dynamic Engineering with your custom requirements. ORB2 is design number 8 for the PMC-BiSerial-III with a corresponding FLASH of 08xx.

The IO are buffered from the FPGA with differential transceivers. The transceivers can be populated with LVDS or RS-485 compatible devices. The power plane for the transceivers is isolated to allow selectable 3.3 or 5V references for the IO. The LVDS IO requires 3.3 and 40 MHz capable RS-485 requires 5V. When mixed LVDS and RS485 are used the reference is set to 3.3 and lower speed RS-485 parts are used that are compatible with the 3.3V.

The ORB2 COM 5,6 channels are limited to 2 MHz due to the in and out nature of those ports. The clock and gate are provided by the master while the target provides the data. The data is changed on the rising edge and sampled on the falling edge meaning that there is only 1/2 period to account for the delays on the Master and Target side of the interface. With the lower speed devices the path delays limits the round trip to a 2 MHz rate.

The other COM ports are single sided in that the data, clock and "strobe" are all driven from the same side so the delay through the gates is not much of an issue and the skew is low enough to allow the higher rate data transfer.

COM 5,6 can be implemented without the LVDS capability at a higher rate as the RS485 parts can be upgraded to the 40 Mhz variety instead of the ones currently in use.

The IO are matched from the connector edge to the ball on the FPGA. The differential side is routed with controlled impedance traces. "Trace and space".

Each of the transceivers have separate direction and termination controls to allow for Any configuration of in and out, half and full duplex designs.

Each of the IO have series terminations to allow the IO to be isolated or terminated. The isolation feature is used to allow rear or front panel implementations without "stub" issues for higher speed signals.

Each IO has pull-up and pull-down options to allow half duplex lines to be set to a "marking" state when no device is on the line. The P is is ganged and the M side is too. Each side can be set to gnd or vcc to allow a '1' or a '0' to be set on the lines. The resistors are in resistor packs and can be implemented with many values.

The terminations utilize analog switches to selectively parallel terminate the differential pair with approximately 100 ohms. It is recommended that the receiver side provide the termination.

The analog switches are protected with a DIODE on the input side of the power supply. The switches can back-feed voltage into the rest of the circuit when the PMC is powered down and the system connected to it is not. The DIODEs allow for more flexible operation and power sequencing.

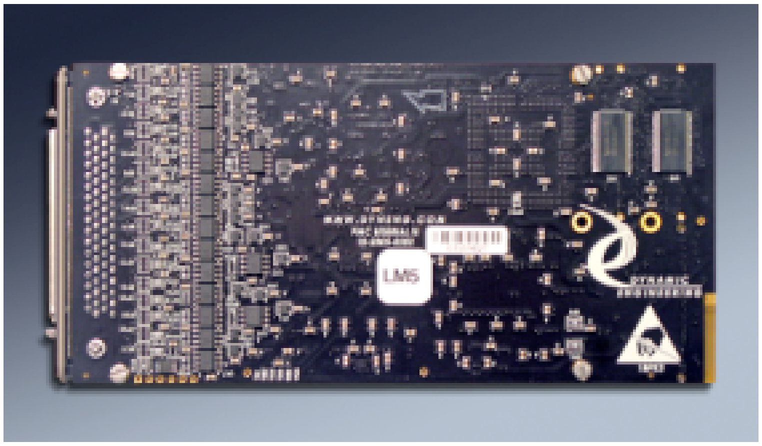


Figure 1 PMC-BISERIAL-III REAR VIEW

The registers are mapped as 32 bit words and support 32 bit access. Most registers are read-writeable. The Windows® compatible driver is available to provide the system level interface for this version of the Biserial III. Use standard C/C++ to control your hardware or use the Hardware manual to make your own software interface. The software manual is also available on-line. Linux is available by request.

PMC-BISERIAL-III is part of the PMC Module family of modular I/O components. The PMC-BISERIAL-III conforms to the PMC standard. 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-III can be used for multiple purposes with applications in telecommunications, control, sensors, IO, test; anywhere multiple independent or coordinated IO are useful.

PMC-BISERIAL-III features a Xilinx FPGA, and high speed differential devices. The FPGA contains the PCI interface and control required for the parallel interface.

The Xilinx design incorporates the “PCI Core” and additional modules for DMA in parallel with a direct register decoded programming model. The design model has a “base” level with the basic board level functions and “channels” which contain IO oriented functions. In the ORB2 design the COM functions are designed into channels and the PLL programming, switch, and other common or basic functions are in the base design.

From a software perspective the design can be treated as “Flat” or as a hierarchy. The Dynamic Engineering Windows® driver uses the hierarchical approach to allow for more consistent software with common bit maps and offsets. The user software can control the COM pairs with the same calls and use the channel number to distinguish. This makes for consistent and easier to implement user level software.

The hardware is designed with each of the channels on a common address map – each channel has the same memory allocated to it and as much as possible the offsets within each space are defined in the same way or similar way as that COM port requires. Again this make understanding each of the COM ports easier to accomplish and less likely to have errors.

The transceivers are initialized to the receive state.   Once a channel is defined via software to be a transmitter the IO are enabled and driven to the appropriate levels. Terminations are activated for the receive defined ports.

All the IO control and registers are instantiated within the FPGA, only the transceivers and termination switches are separate devices.   If desired, the IO lines can be specially programmed to create custom timing pulses etc.   Please contact Dynamic Engineering with your requirements.

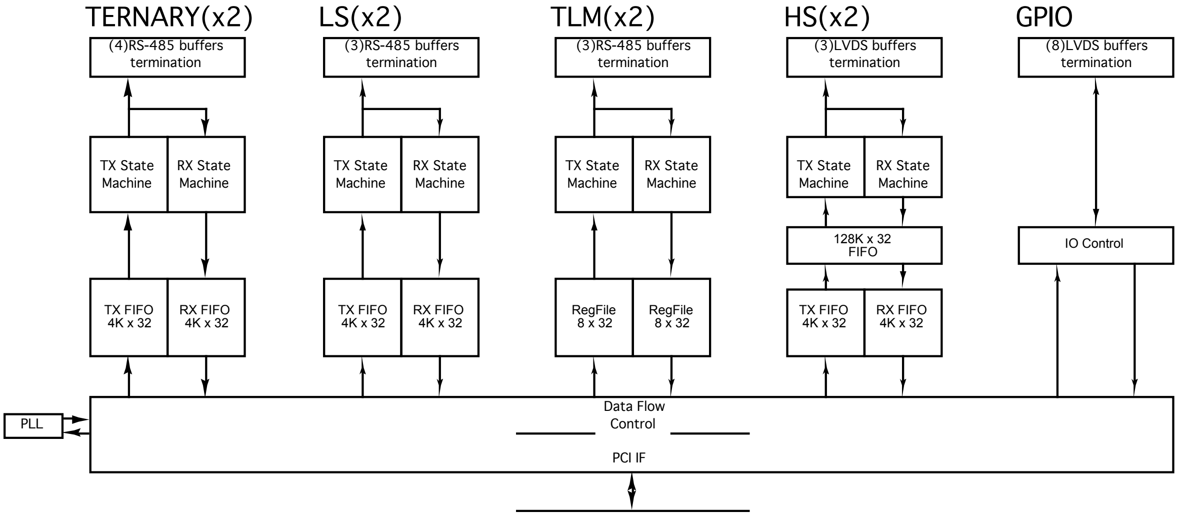


Figure 2 PMC-BISERIAL-III-ORB2 Block Diagram

PMC BiSerial III ORB2 features 8 COM ports with DMA support for all but COM 5,6. The internal block RAM is configured to provide FIFO's to support the COM ports. COM1-4 have 4K x 32 RX. COM7,8 have external 128K x32 plus internal 4K x 32 FIFO's to support the higher rate of the data on those channels.

The hardware will pull data from the FIFO memory and store into the system memory using DMA and vice-versa. The COM function will load the FIFO and DMA will unload. The DMA function operates at the PCI bus frequency. The COM frequency will determine the maximum load rate into the FIFO.

The DMA FIFO is 4Kx32 for COM1-4 leaving a lot of "rubber band" in the memory chain to support the function.

COM7 and COM8 use a 4K plus the 128K FIFO's to form the TX path or RX path. All of the FIFO ports operate at the PCI rate for COM7,8. The internal FIFO's are used to interact with the DMA process. The external holding FIFO is auto filled or auto read with the data moved to or from the "DMA FIFO". The Rx and Tx state-machines use rate matching logic to switch between the programmed IO rate and the FIFO rate.

The DMA programmable length is 32 bits => longer than most computer OS will allow in one segment of memory. The DMA is scatter gather capable for longer lengths than the OS max and for OS situations where the memory is not contiguous. With Windows® lengths of 4K are common while Linux can provide much larger spaces. Larger spaces are more efficient as there are fewer initialization reads and reduced overhead on the bus. A single interrupt can control the entire transfer. Head to tail operation can also be programmed with two memory spaces with two interrupts per loop.

The BiSerial naming convention is channel0-N. The COM ports are names 1-8 based on client requirements. Channel0 corresponds to COM1…Channel7 corresponds to COM8. Please note the mapping when using the Dynamic Driver for channel selection or referencing the channel registers defined later in this document.

A brief summary is provided here as a quick reference. The bit maps and programming details are in the next section.

A complete transmitter and a complete receiver are designed into each COM port. The ports are "half duplex" in the sense that they can operate as a Transmitter or a Receiver but not both at the same time. With two of each port type loop-back between the ports is possible. The manufacturing test procedure uses the loop-back capability to thoroughly test each unit.

COM1,2 support Ternary encoding/decoding. Each port has ONE, ZERO, CLK and S using RS-485 transceivers. The data is encoded with the ONE and ZERO lines where only 1 is active for a somewhat redundant capability. S is similar to a data strobe. Clock is used to sample the ONE, ZERO and S signals. The rate is programmable on each channel. The receiver samples the clock input and can automatically adapt to different clock rates. IO is tested up to 10 MHz.

DATA	ONE	ZERO	S
1	H	L	L
0	L	H	L
S-1	H	H	H
S-0	L	L	H

Please note that S-0 and S-1 are alternate encoding standards supported for the S signal.

Several of the operational parameters are programmable including length of data sent, length of delay between data sent, continuous or bursted clock, inversion on the CLK and S lines.

Rising CLK edge valid data, and active low S are the standard settings.

The Transmitter features the DMA path and a register based path. The DMA path can set to operate with "background" information to transmit, and the registers used for real time data. The register data is sent at the next available data transfer period as defined with the length and delay parameters.

Please see the register definitions for more information.

COM3,4 implement LS (Low Speed) ports. The signals supported are LS DATA, LS CLK and LS DATA VALID. Each signal is supported with an RS-485 transceiver.

The VALID signal standard is active high and uses the falling edge for stable data.

The length is programmable on a 32 bit basis.

Four modes of operation “PV”, VO, “SO” and “CO”. PV is packet based data using the VALID signal for length demarcation. VO is VALID based without using programmed packet lengths. SO is “Synchronization Only” mode and uses the CLK and Data lines to check against a programmable sync pattern. CO is “Clock Only” mode and captures data unaligned as long as it is enabled.

In Packet mode the valid signal is used to control capture and the programmed length is used to check for the correct length. Status can be appended to each message. If the message is shorter than expected the message is padded until the expected length is met. Status indicates “Word Short” in this case. If the message is longer than expected the message is truncated and the status indicates “Word Long”.

In non packetized modes the programmed length is ignored.

In Sync mode the synchronization pattern can be up to 64 bits long. The captured data is compared with 8 bit comparators. When the enabled bytes match sync is declared and data is captured with the next bit after the sync pattern. The bytes can be separately enabled to allow for different lengths of Sync detection. Disabled bytes act like “don’t care” which will allow for bus snooping and other configurations.

In Clock Only mode the software will enable the receiver. The receiver will initialize and then start to capture data based on the received clock. Reception will continue until disabled by software. Software will need to take care of bit alignment in this mode.

COM5,6 implement TLM (Telemetry) ports. In the TLM mode the ports act as Master or Target, As a Master the HW will transmit the GATE and Clock and read back data from a Target port. As a Target the HW will receive the Clock and GATE and supply data to the master.

In Type 1 the clock is masked and offset from the Gate signal providing 1 clock after the

Gate has been taken inactive. The data transmitted and received happens with the clock; the last bit is received after the GATE is taken inactive. Data is stable on the rising edge. Gate is active high.

In Type 2 the clock is free running and the Gate matches the data. The falling edge has stable data. Gate is active low.

The hardware is designed with the Gate and Clock inversions programmed separately to allow for all four combinations. The specific timing of the Gate is also programmable to match Type 1 or Type 2.

COM7,8 implement HS (High Speed) ports. COM7 and COM8 use the same basic protocol as COM3 and COM4. The clock must be free running for the receiver to operate. The intended frequency is 85 MHz. External FIFO's are used in conjunction with the internal FIFO's to create much larger buffers. The frequency is controlled with the PLLB output.

The TX side has an added feature of using a programmable count of clocks to suppress. When enabled N counts are masked out while the data continues to be sent, then the clock is restored. The control bit auto clears.

The PLL is used to provide a reference to COM ports 1-6 and a direct control over COM7,8. PLLA and PLLB are programmed by creating a .JED file using the Cypress CyberClocks software. The PLL in use is the CY22393. The reference clock is 50.000 MHz. PLLA should be set to 140 MHz.

PLLB is user selectable and has a design range of 50-85 MHz. Lower frequencies may also work. No real reason the lower frequencies would not work, but not tested below 50. Higher frequencies may work under some conditions. The FPGA timing requirements are set with an upper limit at 100. The design has not been tested above 85 Mhz. Higher frequencies may be obtainable by reducing some of the clocking options – inversion and masking to provide more margin.

Address Map

Function	Offset
// PMC BiSerial III ORB2 definitions	
#define ORB2_BASE_BASE	0x0000 // 0 ORB2Base Base control register
#define ORB2_BASE_PLL_WRITE	0x0000 // 0 ORB2Base Base control register
#define ORB2_BASE_PLL_READ	0x0000 // 0 ORB2Base base control register
#define ORB2_BASE_USER_SWITCH	0x0004 // 1 ORB2Base User switch read port DIP switch read
#define ORB2_BASE_XILINX_REV	0x0004 // 1 ORB2Base Xilinx revision read port
#define ORB2_BASE_XILINX_DES	0x0004 // 1 ORB2Base Xilinx design read port
#define ORB2_BASE_STATUS	0x0008 // 2 ORB2Base status Register offset
#define ORB2_GPIO_TERM	0x0040 //16 Termination Programming for GPIO Port
#define ORB2_GPIO_DIR	0x0044 //17 Direction Programming for GPIO Port
#define ORB2_GPIO_DATAREG	0x0048 //18 IO Data Register for GPIO Port
#define ORB2_GPIO_DATAIN	0x004C //19 RX Data read from GPIO Port

Figure 3 PMC-BISERIAL-III Internal Address Map Base Functions

The address map provided is for the local decoding performed within PMC-BiSerial-III. The addresses are all offsets from a base address. The carrier board that the PMC is installed into provides the base address. Dynamic Engineering prefers a long-word oriented approach because it is more consistent across platforms.

The map is presented with the #define style to allow cutting and pasting into many compilers "include" files.

The host system will search the PCI bus to find the assets installed during power-on initialization. The VendorId = 0x10EE and the CardId = 0x0038 for the PMC-BiSerial-III-ORB2.

The ORB2 design has 9 sections. The BASE contains the common elements of the design, while the 8 Channels have the IO specific interfaces. The BASE starts at the card offset. There are 20 register addresses assigned to the BASE 0-19. Each channel also has 20 register addresses assigned.

Section	Register Address Range	COM name
	(starting Hex address)	
Base	0-19 (0x0000)	no COM port, GPIO, PLL, Switch, Status
Channel 0	20-39 (0x0050)	COM1 – Ternary Port
Channel 1	40-59 (0x00A0)	COM2 – Ternary Port
Channel 2	60-79 (0x00F0)	COM3 – LS Port
Channel 3	80-99 (0x0140)	COM4 – LS Port
Channel 4	100-119 (0x0190)	COM5 – Telemetry Port
Channel 5	120-139 (0x01E0)	COM6 – Telemetry Port
Channel 6	140-159 (0x0230)	COM7 – HS Port
Channel 7	160-179 (0x0280)	COM8 – HS Port

```
Function                                     Offset from Channel Base Address
// PMC BiSerial III ORB2 Channel definitions
#define ORB2_CHAN_CNTRL               0x0000  //0 ORB2Chan General control register
#define ORB2_CHAN_STATUS              0x0004  //1 ORB2Chan Interrupt status port
#define ORB2_CHAN_INT_CLEAR           0x0004  //1 ORB2Chan Interrupt clear port

#define ORB2_CHAN_WR_DMA_PNTR         0x0008  //2 ORB2Chan Write DMA dpr[1] physical PCI add
#define ORB2_CHAN_TX_FIFO_COUNT       0x0008  //2 ORB2Chan Tx FIFO count read port
#define ORB2_CHAN_RD_DMA_PNTR         0x000C  //3 ORB2Chan Read DMA dpr physical PCI add
#define ORB2_CHAN_RX_FIFO_COUNT       0x000C  //3 ORB2Chan Rx FIFO count including pipeline
#define ORB2_CHAN_FIFO                0x0010  //4 ORB2Chan FIFO single word access RW
#define ORB2_CHAN_TX_AMT_LVL          0x0014  //5 ORB2Chan Rx almost empty level register RW
#define ORB2_CHAN_RX_AFL_LVL          0x0018  //6 ORB2Chan Rx almost full  level register RW
#define ORB2_CHAN_RX_SYNC_PAT         0x001C //7 ORB2Chan RX Sync Pattern for COM1,2
#define ORB2_CHAN_RX_CNTL             0x0020  //8 ORB2Chan Rx Control R/W
#define ORB2_CHAN_TX_CNTL             0x0024  //9 ORB2Chan Tx Control R/W
#define ORB2_CHAN_RX_LEN_ERR          0x0028  //10
//ORB2Chan Length Error Counter - write = load, read = count or preload value

General Purpose Registers used for specific purposes on each COM channel.  All are RW.
#define ORB2_CHAN_TX_REG0             0x002C  //11 ORB2Chan Tx Alternate Register Path Reg 0
#define ORB2_CHAN_TX_REG1             0x0030  //12 ORB2Chan Tx Alternate Register Path Reg 1
#define ORB2_CHAN_TX_REG2             0x0034  //13 ORB2Chan Tx Alternate Register Path Reg 2
#define ORB2_CHAN_TX_REG3             0x0038  //14 ORB2Chan Tx Alternate Register Path Reg 3
#define ORB2_CHAN_TX_REG4             0x003C  //15 ORB2Chan Tx Alternate Register Path Reg 4
#define ORB2_CHAN_TX_REG5             0x0040  //16 ORB2Chan Tx Alternate Register Path Reg 5
#define ORB2_CHAN_TX_REG6             0x0044  //17 ORB2Chan Tx Alternate Register Path Reg 6
#define ORB2_CHAN_TX_REG7             0x0048  //18 ORB2Chan Tx Alternate Register Path Reg 7
#define ORB2_CHAN_TX_CNT              0x004C  //19 ORB2Chan Tx BitCount Delay Count Reg
```

Figure 4 PMC-BISERIAL-III Channel Address Map

[1] **DPR = Descriptor Pointer and is the physical location of the descriptor in host memory**

Programming

Programming the PMC-BISERIAL-III-ORB2 requires only the ability to read and write data in the host's PMC space.

Once the initialization process has occurred, and the system has assigned addresses to the PMC-BiSerial-III-ORB2 card the software will need to determine what the address space is for the PCI interface [BAR0]. The offsets in the address tables are relative to the system assigned BAR0 base address.

The next step is to initialize the PMC-BiSerial-III-ORB2. The PLL will need to be programmed to use any of the COM ports. The Cypress CyberClocks software can be used to create new .JED files if desired. The PLLA should be set to 140 MHz. PLLB can be programmed based on your requirements for COM7,8.

The driver comes with several .JED files prepared. pmcbis3_freqafreqb.jed is the naming convention with the freqa being the PLLA frequency, and freqb the PLLB frequency. For example pmcbis3_140a50b.jed will cause PLLA = 140 MHz. and PLLB = 50 MHz. The driver has a utility to load the PLL and read back. The reference application software has an example of the use of PLL programming.

Each COM port has a separate set of registers to initialize. Each set is independent of the others. For the COM Ports 1-6 PLLA is divided locally to a reference frequency for the COM port. PLLA is used directly to control the receiver. If the COM port is used as a receiver the TX control can be left in the reset state. If the COM port is used as a transmitter the clock rate will need to be selected.

The IO direction and termination is automatic and determined by the state of the transmitter control register. The start signal will cause the port to initialize transmission. If the “DMA” start bit is enabled the IO direction will be set to transmit, and stay in that mode until software intervenes. If the “Register” start option is used the drivers will only be enabled during the transmission. When both are selected the drivers follow the DMA definition.

The other control bits will select how the data is transmitted – clock orientation, strobe sense, all at once or in packets, delays between packets and so forth. In most cases the bit/word count, delay count [same register] and tx control registers will be all that is required.

For receiving the received clock is treated as a signal and sampled with PLLA. When the programmed edge is found the data is captured into the shift register. When sufficient bits are captured the data is loaded into memory. If the receiver is in a mode where the data has an expected length, status can be inserted indicating if the message was of the proper length. Messages that are incorrect are modified to match the expected size and status appended [programmable on/off] with the short or long indication for “not enough” or “too much”.

COM 7 and COM 8 use PLLB to control the transmit frequency. The clock is used directly rather than divided down. Other than the PLLB clock reference the control registers are similar to the other ports. The receiver requires a free running clock. The transmitter has a mask bit to allow bursted clocks on the output side.

For Windows™ and Linux systems the Dynamic Drivers[2] can be used. The driver will take care of finding the hardware and provide an easy to use mechanism to program the hardware. The Driver comes with reference software showing how to use the card and reference frequency files to allow the user to duplicate the test set-up used in manufacturing at Dynamic Engineering. Using simple, known to work routines is a good way to get acquainted with new hardware.

To use the ORB2 specific functions the Channel Control, and PLL interface plus DMA will need to be programmed. To use DMA, memory space from the system should be allocated and the link list stored into memory. The location of the link list is written to the ORB2 to start the DMA. Please refer to the Burst IN and Burst Out register discussions.

DMA should be set-up before starting the COM port function. For transmission this will result in the FIFO being full or close to it when the transfer is started. For reception it means that the FIFO is under HW control and the delay from starting reception to starting DMA won’t cause an overflow condition.

DMA can be programmed with a specific length. The length can be as long as you want within standard memory limitations. At the end of the DMA transfer the Host will receive an interrupt. The receiver can be stopped and the FIFO reset to clear out any extra data captured. For on-the-fly processing multiple shorter DMA segments can be programmed; at the interrupt restart DMA to point at the alternate segment to allow processing on the previous one. This technique is sometimes referred to as “ping-pong”.

[2] Currently only Windows® is supported. Please contact Dynamic Engineering for Linux.

Base Register Definitions

ORB2_BASE_BASE

[$00 parallel-io Control Register Port read/write]

DATA BIT	DESCRIPTION
31-21	spare
20	bit 19 read-back of pll_dat register bit
19	pll_dat [write to PLL, read-back from PLL]
18	pll_s2
17	pll_sclk
16	pll_en
15-0	spare

Figure 5 PMC-BISERIAL-III Control Base Register Bit Map

This is the base control register for the PMC BiSerial III ORB2. The features common to all channels are controlled from this port. Unused bits are reserved for additional new features. Unused bits should be programmed '0' to allow for future commonality.

pll_en: When this bit is set to a one, the signals used to program and read the PLL are enabled.

pll_sclk/pll_dat : These signals are used to program the PLL over the I^2C serial interface. Sclk is always an output whereas Sdata is bi-directional. This register is where the Sdata output value is specified or read-back.

pll_s2: This is an additional control line to the PLL that can be used to select additional pre-programmed frequencies. Set to '0' for most applications.

The PLL is programmed with the output file generated by the Cypress PLL programming tool. [CY3672 R3.01 Programming Kit or CyberClocks R3.20.00 Cypress may update the revision from time to time.] The .JED file is used by the Dynamic Driver to program the PLL. Programming the PLL is fairly involved and beyond the scope of this manual. For clients writing their own drivers it is suggested to get the Engineering Kit for this board including software, and to use the translation and programming files ported to your environment. This procedure will save you a lot of time. For those who want to do it themselves the Cypress PLL in use is the 22393. The output file from the Cypress tool can be passed directly to the Dynamic Driver [Linux or Windows] and used to program the PLL without user intervention.

The reference frequency for the PLL is 50 MHz.

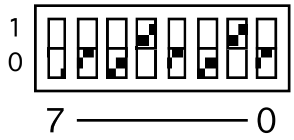
ORB2_BASE_ID

[$04 Switch and Design number port read only]

DATA BIT	DESCRIPTION
31-24	spare
23-8	Design ID and Revision
7-0	DIP switch

Figure 6 PMC-BISERIAL-III ID and Switch Bit Map

The DIP Switch is labeled for bit number and ‘1’ ‘0’ in the silk screen. The DIP Switch can be read from this port and used to determine which PMC BiSerial III physical card matches each PCI address assigned in a system with multiple cards installed. The DIPswitch can also be used for other purposes – software revision etc. The switch shown would read back 0x12.



The Design ID and Revision are defined by a 16 bit field allowing for 256 designs and 256 revisions of each. The ORB2 design is 0x09 the current revision is 0x01.

The PCI revision is updated in HW to match the design revision. The board ID will be updated for major changes to allow drivers to differentiate between revisions and applications.

ORB2_BASE_STATUS

[$08 Board level Status Port read only]

DATA BIT	DESCRIPTION
31-12	spare
11	reserved PLLC locked
10	reserved PLLD locked
9	PLLB Locked
8	PLLA Locked
7	Channel 7 interrupt
6	Channel 6 Interrupt
5	Channel 5 Interrupt
4	Channel 4 Interrupt
3	Channel 3 Interrupt
2	Channel 2 Interrupt
1	Channel 1 Interrupt
0	Channel 0 Interrupt

Figure 7 PMC-BISERIAL-III Status Port Bit Map

Channel 7-0 Interrupt – These are the local masked interrupt status from each channel. Each channel can have different interrupt sources. DMA Write or DMA Read or IntForce or TX/RX request are typical sources.

Interrupt Status – Set if the PCI interrupt is asserted. These bits can be checked to determine if this card is causing an interrupt to the system. If set the other bits can be checked to see which feature(s) of the board need to be serviced. Secondary reads to the Channel status registers will determine the exact type of interrupt. Software can implement any priority scheme desired when dealing with multiple interrupts.

PLL Locked is status indicating that the local DCM is properly tracking the PLL port. The DCM is automatically reset when it comes out of lock to allow it to retrack and acquire. The oscillator and PLL are stable elements and will stay in lock once the initial lock is achieved. When changing frequencies it is recommended to check for lock before using the affected HW.

ORB2_GPIO_TERM

[$40 Termination Register bits 15-0 read – write]

DATA BIT	DESCRIPTION
15-8	Spare
7-0	GpioTerm7-0

Figure 8 PMC-BISERIAL-III GPIO Termination Bit Map

The terminations for the parallel port are controlled with this port. When reset this port is cleared 0x00. Setting each bit enables the termination for the corresponding IO bit. LVDS requires termination, preferably at the receiver.

ORB2_GPIO_DIR

[$44 Direction Register bits 15-0 read – write]

DATA BIT	DESCRIPTION
15-8	Spare
7-0	GpioDir7-0

Figure 9 PMC-BISERIAL-III GPIO Direction Bit Map

The Direction for the parallel port bits are controlled with this port. When reset this port is cleared 0x00. Setting each bit enables the corresponding IO bit to be a transmitter. Clearing each bit disables the transmit function. The IO can be read independent of the transmit or receive condition on each bit. LVDS is a point-to-point electrical specification. Care should be taken not to enable transmitters onto lines driven by other sources until those sources are disabled.

ORB2_GPIO_DATAREG

[$48 Data IO Port read/write]

DATA BIT	DESCRIPTION
15-8	Spare
7-0	GpioDataReg7-0

Figure 10 PMC-BISERIAL-III GPIO Data Register Bit Map

The Data Register is used to capture the data to be transmitted. Each bit corresponds to an IO bit. Each bit can be set or cleared independently. Each bit is independent of the Direction definition. For example writing 0xff to the data register with the direction register set to 0xaa will result in the bits 7,5,3,1 being driven and 6,4,2,0 not being driven. A read from the DataReg will always return what was written to it. A read from the IO port will return the combination of what was driven locally and what is controlled externally.

ORB2_GPIO_DATAIN

[$4C Data IO Port read/write]

DATA BIT	DESCRIPTION
15-8	Spare
7-0	GpioDataIn7-0

Figure 11 PMC-BISERIAL-III GPIO Data IN Bit Map

The state of the GPIO IO side is read from this port. The port will return whatever bits are defined as outputs in the direction register combined with the external sources on the alternate bits.

Channel Bit Maps

The ORB2 design has 8 channels, 4 types with 2 of each type. The basic control signals are the same for the channel base, channel status, FIFO and DMA interfaces. The following descriptions will be in the form of a common feature description for each address and then differences if any for each channel.

Notes:
The offsets shown are relative to the channel base address not the card base address.

ORB2_CHAN_CNTRL

[0x0] Channel Control Register (read/write)

Channel Control Register

Data Bit	Description
31-13	spare
12	FIFO External Reset [HS channels only]
11-9	Spare
8	OutUrgent
7	InUrgent
6	Read DMA Interrupt Enable
5	Write DMA Interrupt Enable
4	Force Interrupt
3	Channel Interrupt Enable
2	Bypass
1	FIFO RX Reset
0	FIFO TX Reset

Figure 12 PMC-BISERIAL-III channel Control Register

FIFO TX/RX Reset: When set to a one, the transmit and/or receive FIFOs will be reset. When these bits are zero, normal FIFO operation is enabled. In addition the TX and RX State Machine is also reset.

Write/Read DMA Interrupt Enable: These two bits, when set to one, enable the interrupts for DMA writes and reads respectively.

Channel Interrupt Enable: When this bit is set to a one, all enabled interrupts (except the DMA interrupts) will be gated through to the PCI interface level of the design; when this bit is a zero, the interrupts can be used for status without interrupting the host. The channel interrupt enable is for the channel level interrupt sources only.

Force Interrupt: When this bit is set to a one, a system interrupt will occur provided the Channel Interrupt and master interrupt enables are set. This is useful for interrupt testing.

InUrgent / OutUrgent when set causes the DMA request to have higher priority under certain circumstances. Basically when the TX FIFO is almost empty and InUrgent is set the TX DMA will have higher priority than it would otherwise get. Similarly if the RX FIFO is almost full and OutUrgent is set the read DMA will have higher priority. The purpose is to allow software some control over how DMA requests are processed and to allow for a higher rate channel to have a higher priority over other lower rate channels.

ByPass when set allows the FIFO to be used in a loop-back mode internal to the device. A separate state-machine is enabled when ByPass is set and the TX and RX are not enabled. The state-machine checks the TX and RX FIFO's and when not empty on the TX side and not Full on the RX side moves data between them. Writing to the TX FIFO allows reading back from the RX side. An example of this is included in the Driver reference software.

FIFO External Reset: When cleared to a zero, the External FIFOs will be reset. When set the External FIFO is enabled. The HS channels have external 128Kx32 FIFO's attached. Please note that the state of the Load pin in the transmit control register affects how the part comes out of reset – what the default Almost Full and Almost Empty offsets are.

ORB2_CHAN_STATUS

[0x4] Channel Status Read/Clear Latch Write Port

Channel Status Register

Data Bit	Description
31	Interrupt Status
30	Local Interrupt
29	spare
28	Direction
27	FifoFullExternal
26	FifoAlmostFullExternal
25	FifoAlmostEmptyExternal
24	FifoEmptyExternal/VerboseSmIdle
23	BurstInIdle
22	BurstOutIdle
21	TxSmIdle
20	RxSmIdle
19	FIFO Underrun error
18	FIFO Overrun error
17	VerboseFifoOverFlow
16	spare
15	Read DMA Interrupt Occurred
14	Write DMA Interrupt Occurred
13	Read DMA Error Occurred
12	Write DMA Error Occurred
11	RxFifoInt
10	TxFifoInt
9	RxInt
8	TxInt
7	spare
6	RX FIFO Full
5	RX FIFO Almost Full
4	RX FIFO Empty
3	Spare
2	TX FIFO Full
1	TX FIFO Almost Empty
0	TX FIFO Empty

Figure 13 PMC-BiSerial-III Channel STATUS PORT

ORB2 FIFO: Two 4K x 32 FIFO's are used to create the internal Rx and Tx memory for COM1-4,7 and 8. The status for the TX FIFO and RX FIFO refer to these FIFO's. The status is active high. 0x13 would correspond to empty RX and empty TX internal FIFO's.

Please note with the Receive side status; the status reflects the state of the FIFO and does not take the 4 deep pipeline into account. For example the FIFO may be empty and there may be valid data within the pipeline. The data count with the combined FIFO and pipeline value and can also be used for read size control. [see later in register descriptions]

RX FIFO Empty: When a one is read, the FIFO contains no data; when a zero is read, there is at least one data word in the FIFO.

RX FIFO Almost Full: When a one is read, the number of data words in the data FIFO is greater than the value written to the corresponding RX_AFL_LVL register; when a zero is read, the FIFO level is less than that value.

RX FIFO Full: When a one is read, the receive data FIFO is full; when a zero is read, there is room for at least one more data-word in the FIFO. If the FIFO is full when time to write received data to the FIFO an overflow error is declared. [COM1-4 only]

TX FIFO Empty: When a one is read, the FIFO contains no data; when a zero is read, there is at least one data word in the FIFO. If the FIFO is empty when time to read transmitted data from the FIFO an underflow error is may be declared. [COM1-4 only]

TX FIFO Almost Empty: When a one is read, the number of data words in the data FIFO is less than or equal to the value written to the corresponding TX_AMT_LVL register; when a zero is read, the FIFO level is more than that value.

TX FIFO Full: When a one is read, the receive data FIFO is full; when a zero is read, there is room for at least one more data-word in the FIFO.

FifoXXExternal refer to the external 128K x 32 FIFO's attached to the COM7 and 8 ports. Please note that COM7 and 8 also have internal FIFO's.

FifoEmptyExternal when set indicates that the external FIFO is empty. The internal FIFO may not be empty yet. When cleared there is at least 1 data location within the external FIFO.

FifoAlmostEmptyExternal when set indicates that the external FIFO is almost empty. FifoAlmostFullExternal when set indicates that the external FIFO is almost full.

The level is programmable with two mechanisms. When the FIFO is taken out of reset the load signal is tested. If load is low the FIFO defaults to 127 from empty and 127 from full for the Almost Empty and Almost Full test levels respectively. When load is high coming out of reset the default becomes 1023.

If load is low during reset the parallel [data] port on the FIFO can be used to reprogram the Almost full and Almost empty levels. Please refer to the IDT72V3610 data sheet for more information regarding programming the levels.

FifoFullExternal when set indicates that the external FIFO is full. If the External FIFO is full when time to write received data to the FIFO an overflow error is declared. [COM7,8 only]

FIFO Overflow Error Occurred: When a one is read, an error has been detected. This will occur if FIFO is full when the loader function tries to write to it. A zero indicates that no error has occurred. This bit is latched and can be cleared by writing back to the Status register with a one in the appropriate bit position.

VerboseFifoOverFlow Error Occurred: When a one is read, an error has been detected. This will occur if FIFO is full when the Verbose Mode loader function tries to write to it. A zero indicates that no error has occurred. This bit is latched and can be cleared by writing back to the Status register with a one in the appropriate bit position.

Write/Read DMA Error Occurred: When a one is read, a write or read DMA error has been detected. This will occur if there is a target or master abort or if the direction bit in the next pointer of one of the chaining descriptors is incorrect. A zero indicates that no write or read DMA error has occurred. These bits are latched and can be cleared by writing back to the Status register with a one in the appropriate bit position.

Write/Read DMA Interrupt Occurred: When a one is read, a write/read DMA interrupt is latched. This indicates that the scatter-gather list for the current write or read DMA has completed, but the associated interrupt has yet to be processed. A zero indicates that no write or read DMA interrupt is pending.

Channel Interrupt Active: When a one is read, it indicates that a system interrupt is potentially asserted caused by an enabled channel interrupt condition. A zero indicates that no system interrupt is pending from an enabled channel interrupt condition. The Board level master interrupt enable will also need to be asserted to allow the active channel interrupt to become an interrupt request.

RX_IDLE is set when the state-machine is in the idle state. When lower clock rates are used it may take a while to clean-up and return to the idle state. If SW has cleared the start bit to terminate the reception; SW can use the IDLE bit to determine when the HW has completed its task and returned.

TX IDLE is set when the state-machine is in the idle state. When lower clock rates are used it may take a while to clean-up and return to the idle state. If SW has cleared the start bit to terminate the transmission; SW can use the IDLE bit to determine when the HW has completed its task and returned.

BO and BI Idle are Burst Out and Burst In IDLE state status for the Receive and Transmit DMA actions. The bits will be 1 when in the IDLE state and 0 when processing a DMA. A new DMA should not be launched until the State machine is back in the IDLE state. Please note that the direction implied in the name has to do with the DMA direction – Burst data into the card for TX and burst data out of the card for Receive.

VerboseSmIdle is set when the Verbose state-machine is in the idle state. When lower clock rates are used it may take a while to clean-up and return to the idle state. If SW has cleared the Sync or Data start bits to terminate the reception; SW can use the IDLE bit to determine when the HW has completed its task and returned.

Local Interrupt is the masked combined interrupt status for the channel not including DMA. The status is before the master interrupt enable for the channel.

Interrupt Status is the combined Local Interrupt with DMA and the master interrupt enable. If this bit is set this channel has a pending interrupt request.

Direction when set indicates that the channel is acting as a transmitter or master. The channel may not be active. The transceivers are energized in the transmit direction.

TxFifoInt when set indicates that the TX FIFO went Almost Empty. The bit is captured and stays set until cleared by writing with the bit position set. The Almost Empty level is programmable. Useful when using interrupts and reactive DMA transfers – when not sure how much data to write, program the level to cause an interrupt with sufficient time to refill and wait for the interrupt. The FIFO will be able to handle the Full minus Almost empty level as a DMA transfer or single word writes. The current exact count can be read from the Tx FIFO count register.

RxFifoInt when set indicates that the RX FIFO went Almost Full. The bit is captured and stays set until cleared by writing with the bit position set. The Almost Full level is programmable. Useful when using interrupts and reactive DMA transfers – when not sure how much data to read, program the level to cause an interrupt with sufficient time to empty without overflowing, and wait for the interrupt. The FIFO will be source the Almost Full amount as a DMA transfer or single word reads. The current exact count can be read from the Rx FIFO count register.

The TX and RX interrupts are based on State Machine activity and can be used to let the processor know when certain events are completed. In most cases the DMA interrupts will be used to control the process. For COM5,6 [register based] the interrupts indicate when the Master or Target has completed.

ORB2_CHAN_WR_DMA_PNTR

[0x8] Write DMA Pointer (write only)

BurstIn DMA Pointer Address Register

Data Bit	Description
31-2	First Chaining Descriptor Physical Address
1	direction [0]
0	end of chain

Figure 14 PMC-BiSerial-III Write DMA pointer register

This write-only port is used to initiate a scatter-gather write [TX] DMA. When the address of the first chaining descriptor is written to this port, the DMA engine reads three successive long words beginning at that address. Essentially this data acts like a chaining descriptor value pointing to the next value in the chain.

The first is the address of the first memory block of the DMA buffer containing the data to read into the device, the second is the length in bytes of that block, and the third is the address of the next chaining descriptor in the list of buffer memory blocks. This process is continued until the end-of-chain bit in one of the next pointer values read indicates that it is the last chaining descriptor in the list.

All three values are on LW boundaries and are LW in size. Addresses for successive parameters are incremented. The addresses are physical addresses the HW will use on the PCI bus to access the Host memory for the next descriptor or to read the data to be transmitted. In most OS you will need to convert from virtual to physical. The length parameter is a number of bytes, and must be on a LW divisible number of bytes.

Status for the DMA activity can be found in the channel control register and channel status register.

Notes:

1. Writing a zero to this port will abort a write DMA in progress.
2. End of chain should not be set for the address written to the DMA Pointer Address Register. End of chain should be set when the descriptor follows the last length parameter.
3. The Direction should be set to ‘0’ for Burst In DMA in all chaining descriptor locations.

ORB2_CHAN_RD_DMA_PNTR

[0xC] Read DMA Pointer (write only)

BurstIn DMA Pointer Address Register

Data Bit	Description
31-2	First Chaining Descriptor Physical Address
1	direction [1]
0	end of chain

Figure 15 PMC-BiSerial-III Read DMA pointer register

This write-only port is used to initiate a scatter-gather read [RX] DMA. When the address of the first chaining descriptor is written to this port, the DMA engine reads three successive long words beginning at that address. Essentially this data acts like a chaining descriptor value pointing to the next value in the chain.

The first is the address of the first memory block of the DMA buffer to write data from the device to, the second is the length in bytes of that block, and the third is the address of the next chaining descriptor in the list of buffer memory blocks. This process is continued until the end-of-chain bit in one of the next pointer values read indicates that it is the last chaining descriptor in the list.

All three values are on LW boundaries and are LW in size. Addresses for successive parameters are incremented. The addresses are physical addresses the HW will use on the PCI bus to access the Host memory for the next descriptor or to read the data to be transmitted. In most OS you will need to convert from virtual to physical. The length parameter is a number of bytes, and must be on a LW divisible number of bytes.

Status for the DMA activity can be found in the channel control register and channel status register.

Notes:

1. Writing a zero to this port will abort a write DMA in progress.
2. End of chain should not be set for the address written to the DMA Pointer Address Register. End of chain should be set when the descriptor follows the last length parameter.
3. The Direction should be set to ‘1’ for Burst Out DMA in all chaining descriptor locations.

ORB2_CHAN_FIFO

[0x10] Write TX/Read RX FIFO Port

RX and TX FIFO Port

Data Bit	Description
31-0	FIFO data word

Figure 16 PMC-BiSerial-III RX/TX FIFO Port

This port is used to make single-word accesses from the FIFO. Data read from this port will no longer be available for DMA transfers. Writing to the port loads the TX FIFO, Reading unloads the RX FIFO.

ORB2_CHAN_RX_AFL_LVL

[0x18] RX almost-full (read/write)

RX Almost-Full Level Register

Data Bit	Description
31-16	Spare
15-0	RX FIFO Almost-Full Level

Figure 17 PMC-BiSerial-III RX ALMOST FULL LEVEL register

This read/write port accesses the almost-full level register. When the number of data words in the receive data FIFO is equal or greater than this value, the almost-full status bit will be set. The register is R/W for 16 bits. The mask is valid for a size matching the depth of the FIFO. 4k x32 + 4 is the RX FIFO size for a 13 bit valid count range [12-0]. Recommend to set to 16 off boundary for load function [0x10]. The level includes the pipeline for an additional 4 locations [0xFFF + 4 = 0x1003].

ORB2_CHAN_TX_AMT_LVL

[0x14] TX almost-empty level (read/write)

RX Almost-Full Level Register

Data Bit	Description
31-16	Spare
15-0	TX FIFO Almost-Empty Level

Figure 18 PMC-BiSerial-III RX ALMOST FULL LEVEL register

This read/write port accesses the almost-empty level register. When the number of data words in the transmit data FIFO is less than than this value, the almost-empty status bit will be set. The register is R/W for 16 bits. The mask is valid for a size matching the depth of the FIFO. 4k x32 is the TX FIFO for a 12 bit valid count range [11-0]. Recommend setting to 16 off the end count [0xFF0] for each FIFO to support loader function.

ORB2_CHAN_RX_FIFO_COUNT

[0xC] RX FIFO data count (read only)

RX FIFO Data Count Port

Data Bit	Description
31-14	Spare
13-0	RX Data Words Stored

Figure 19 PMC-BiSerial-III RX FIFO data count Port

This read-only register port reports the number of 32-bit data words in the receive FIFO. The channel status register contains the combined pipeline and FIFO count. The size depends on the FIFO size. This design has 4095 locations possible plus the pipeline for a total of [0x1003]. Please note than COM7 and COM8 use the internal FIFO for this size, Data in the external FIFO is not included.

ORB2_CHAN_TX_FIFO_COUNT

[0x8] TX FIFO data count (read only)

TX FIFO Data Count Port

Data Bit	Description
31-14	Spare
13-0	TX Data Words Stored

Figure 20 PMC-BiSerial-III RX FIFO data count Port

This read-only register port reports the number of 32-bit data words in the transmit FIFO.   This design has 4095 locations possible [0xFFF]. Please note than COM7 and COM8 use the internal FIFO for this size, Data in the external FIFO is not included.

ORB2_CHAN_RX_CNTL

[0x20 Receive Control Register read -write]

DATA BIT	DESCRIPTION
31	RxSyncSearch
30-24	spare
24-20	VerboseSyncEn COM12
23	RxSyncEn7 include bits 63-56 in sync check
22	RxSyncEn6 include bits 55-48 in sync check
21	RxSyncEn5 include bits 47-40 in sync check
20	RxSyncEn4 include bits 39-32 in sync check
19	RxSyncEn3 include bits 31-24 in sync check
18	RxSyncEn2 include bits 23-16 in sync check / RxStartVerboseData
17	RxSyncEn1 include bits 15-8 in sync check / RxStartVerboseSync
16	RxSyncEn0 include bits 7-0 in sync check / RegFilePointerClr
15-14	Set to 0
13-12	RxMode
11	Set to 0
10	RxFifoPathEn (HS) / TlmType(TLM)
9	RxDataValidPolarity
8	RxDataFill
7	RxClkPolarity
6	RxDataOrder
5	RxSMode(Tern) RxValidMode (LS/HS)
4	OverFlowInterruptEn
3	RxFifoInterruptEn
2	RxInterruptEn
1	RxStatusDisable
0	RxStart

Figure 21 PMC-BISERIAL-III Receive Control Bit Map

Please note: not all bits are used in all COM interfaces.

RxSyncSearch is used in COM3,4,7,8 [LS and HS ports] only. When set the receiver will terminate the current reception if one is active and begin looking for the synchronization pattern. Please note that the RxStart bit must be set for RxSyncSearch to have an affect.

RxSyncSearch is self clearing – when the receive state machine finds a valid synchronization pattern the state-machine clears the request bit. The cleared status can be used to determine that Sync has been found.

RxSyncEn0-7 are used to enable on a byte basis which bits are tested against the incoming data to find the sync pattern. These bits also only have meaning in the

COM3,4,7,8 ports. Each bit corresponds to a byte in the sync pattern. The max length is 64 bits. When set the byte is included in the search, when cleared the byte becomes a “don’t care”. Shorter lengths can be used by enabling the initial bytes and disabling the later bytes so the pattern is captured before passing through the full 64 bit comparator. Longer tests with don’t cares in certain fields can also be accomplished for snooper modes where an address range may be tested etc.

The data is received LSB first. The data immediately after sync is captured is the start of the first 32 bit word. Once sync is obtained only the clock is used to validate the received data.

The shift register is designed with 31-0 in the data path. DataIn is received into Bit 31 of the shift register and shifted down to bit 0. A counter tracks the data and when 32 are received loads to a holding register while the next word is being captured. The holding register data is then moved to the FIFO and is used to account for the different clock rates. Bit 0 is used to load into a second 32 bit shift register in place to support the sync check. Bits 63-32 come from this second shift register. The bits in the sync detection match the bits in the shift registers as defined in this paragraph.

A somewhat unusual mapping is the result of the bit to test against and the order they are received. 32 bit and 64 bit boundaries track directly.

RxSyncEn3 is the first byte received and would be the only enable activated when an 8 bit sync pattern is desired. RxSyncEn2, 1, 0 would follow for the 16, 24 and 32 bit cases. RxSyncEn7, 6, 5, 4 would be used for the 5, 6, 7, and 8 byte cases respectively. The byte enables would be additive. [use 0 and 1 for 16 bits etc.]

Two of the general purpose registers are used to store the expected sync pattern. Please refer to the ORB2_CHAN_TX_REG section for more details.

RxMode is a two bit field used to select which mode of operation the receiver is in. Applies to LS and HS COM Ports only.
00 = Valid and Packet control, 01 = Valid only, 10 = Sync Only, 11 = clock only.

Valid refers to the DataValidIn signal which acts like a data strobe.
Packet refers to having an expected length and uses hardware to check against the programmed expected length. [ORB2_CHAN_TX_CNT]
Sync mode is described in the RxSyncEn section.
Clock Only mode captures data when enabled and continues until stopped. No implied boundary detection or alignment.

RxFifoPathEn is used in the HS [COM 7 & 8] ports only. When set the path from the external memory to the RX FIFO is enabled. A state-machine checks the levels on the external FIFO and the internal FIFO and when data is available and there is room, moves the data from the external FIFO to the internal FIFO.   The enable should be set for normal operation and FIFO loop-back testing.

TlmType is used in the TLM mode to determine the type of data expected. When cleared type 1 is decoded, and when set type II is decoded. The TLM interface is unusual in that for Type 1 data the last bit is uncovered by the Valid signal and for Type II an extra bit is covered by the valid signal. Also in Type 1 the CLK is active high for valid data and inverted for Type II. The clock is bursted for Type 1 and free running for type II. The other control bits are used to control the clocking and valid orientation to allow for non [specification] defined test cases etc.

RxDataValidPolarity when set causes the HW to invert the received Data Valid signal.
Ternary 0 = standard = S is active low during transmission 1 = inverted
LS / HS 0 = active high Valid, 1 = inverted valid signal expected
TLM [TargetGatePolarity] when 0 Gate is high in Type 1 and low in Type II. When 1 the Gate is inverted.

For all modes this bit is normally programmed low.

RxDataFill When using packetized modes in LS, HS, Ternary if the message received is too short, the missing bits are filled with ‘1’ when RxDataFill is set or ‘0’ when RxDataFill is cleared.

RxClkPolarity
Ternary 0 = use rising edge clock, 1 = use falling edge clock. 0 = standard operation
LS/HS 0 = use falling edge, 1 = use rising edge clock. 0 = standard operation
TLM 0 = use falling edge, 1 = use rising edge. Type 1 uses falling edge for Target [Rx], Type II uses rising edge for Target. Please note that this mode does require changing the Clock Polarity depending on the Type being used.

RxDataOrder   When set the data order is reversed. Data is received LSB first and loaded into the FIFO on LW boundaries. The bits are reversed when this control is enabled. 31=>0 and 0=>31.

RxValidMode When set selects the decoder to use Valid as a condition and 0 selects ignore [clock only decoding]. Set for VO and VP modes. Clear for SO and CO modes.

RxSMode When set selects Type S-1 and when cleared selects type S-0.

OverFlowInterruptEn when set enables the FIFO OverFlow Status to cause an interrupt to the host. When cleared the status can be used but the interrupt will not be active.

RxFifoInterruptEn When set enables the Almost Full FIFO Status to cause an interrupt to the host. When cleared the FIFO counts and FIFO status can still be used.

RxInterruptEn When set and an Rx State Machine interrupt function is defined an interrupt is caused to the host.
Ternary: none defined, expected that DMA and FIFO status are used.
TLM : Set when GateIn activity is detected and completed. Use as signal to update registers before next Target request is sent.
LS/HS : none defined, expected that DMA and FIFO status are used.

RxStatusDisable When set and in a packetized mode the status can be inserted or suppressed. The status when enabled adds a 32 bit word to each message. The status is on a LW boundary. Bit 0 is the “Short” status and bit 1 is the “Long” status. If the message is shorter than expected, the message is padded to the proper size and the status added has the short bit set. If the message is longer than expected the message is truncated and the long bit set. If the message is the correct length no bits are set [0x00]. If the status is disabled the packettes will be on LW boundaries and back to back.

RxStart When set the receiver is enabled to operate. The receiver will continue to operate with the defined characteristics until the RxStart bit is cleared.

Since the ports are effectively half duplex it is possible to create events by switching from off to TX – when the transceivers are enabled going from undefined to the off state may look like a transition. For loop-back testing the Transmitter should be enabled before the receiver to avoid unwanted data from being captured.

Verbose Mode Bits: Verbose mode was added to the COM1 & COM2 ports to allow the received data to include the S and Error bits.   This mode is normally used for Bit Error testing etc. Verbose mode uses the same resources as the standard modes and should not be started at the same time as the other modes. When switching use the IDLE status to insure that all RX state-machine functions are idle before starting a new mode.

Verbose mode has two synchronization methods. RxStartVerboseData and RxStartVerboseSync are used to begin data capture. The store function will continue until SW aborts by clearing the start bit.

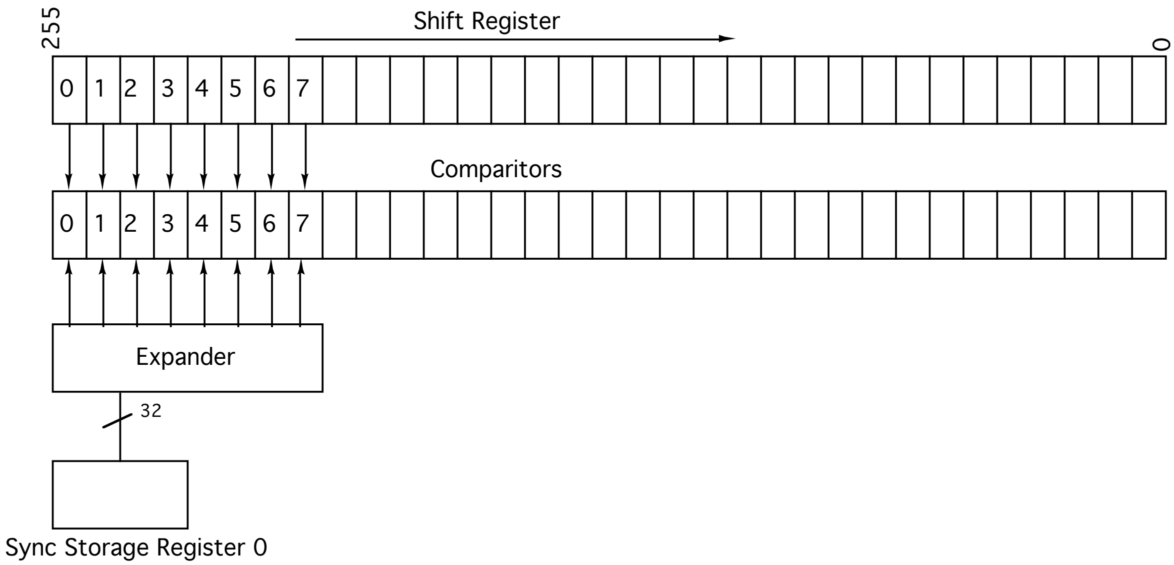
RxStartVerboseData when set causes the HW to wait for the first valid data bit to be received. Once received all data is captured. The initial bit is both a trigger and a stored bit.

RxStartVerboseSync waits for the received data to match a programmed synchronization pattern. The sync pattern is not stored. All received events after the sync pattern are stored.

In verbose mode the data is stored along with S Tones and Error conditions. The added information means that 2 bits are required per received encoded pattern. The clock is sampled and on the active edge the state of the S, ONE and ZERO lines are determined. The follow chart shows the encoding done to capture the state for storage into memory.

S	ONE	ZERO	Encoded	Definition
0	0	1	00	legal ‘0’
0	1	0	01	legal ‘1’
1	0	0	10	legal S Tone
1	0	1	11	Error S Tone
1	1	0	11	Error S Tone
1	1	1	11	Error S Tone
0	1	1	11	Error Data
0	0	0	11	Error Data

Design Hints:
1) that the expansion means that more data is loaded into the receive FIFO than is received, and since S Tones and Errors are loaded a lot more data could be loaded.
2) Since the data is packed S TONES the received patterns will not be realigned to LW boundaries.
3) Since data mode starts with data it will tend to have data to start the received information. Since Sync mode starts with the sync pattern, and can be set to be the size of the packet being send many times the stored data will start with S TONES.



Verbose Sync mode requires a sync pattern. The pattern is stored into the Sync Register File for COM1/2 – address offset 7 for the channel. The register file has 8 registers of which the first 4 align with the sync pattern to be checked for. The last 4 are currently spares and can be used for future expansion. The actual number of bits used in the sync pattern is programmable via VerboseSyncEn.

The sync pattern is applied after the data has been expanded with the two bit conversion table shown above. The sync pattern is stored in standard form and automatically expanded to be checked in the two bit format. Since each stored bit in the sync pattern registers can only represent data the sync pattern will only trigger on data.

The Enables are encoded with the 5 bit word. 0-1F correspond to 1-32 bytes to match in the expanded data – this provides nibble level selection for the stored sync pattern. Setting to 0 uses the 1st byte of the expanded data, setting to 1 uses bytes 0 and 1.

The bytes are numbered as shown in the block diagram. The data is treated as LSB first and flows through a shift register 255->0. Byte 0 uses 255-247 and byte 32 covers 7-0. For a given sync pattern length it is best to refer to the diagram and see which bytes cover the sync pattern and which bits within the sync pattern registers are involved.

For example with a 32 bit sync pattern 64 bits of expanded data are used. The data will flow from 255 toward 0 and when the 64 bits in the shift register match the expanded 64 bits the sync will match. This means that the first 8 expanded bytes are used or the first

4 sync pattern bytes are used. SyncReg0 should be programmed with the expected pattern.

For lengths not on a LW boundary the pattern will be MSB aligned since the data flows from MSB to LSB.

ORB2_CHAN_RX_SYNC_PAT

[0x1C from channel base Verbose Mode Sync Pattern Storage]

DATA BIT	DESCRIPTION
31-0	SyncPattern stored as a register file.

The 8 registers are used for storing the Sync Pattern used in the Verbose mode on COM1 and COM2. The registers are read-writeable. Set and release CLR control in RX control register to reset pointers prior to use. Please see the description above.

The address acts as a register file. Writing modifies the current register and advances the pointer to the next register in a loop. The clear control forces the pointer to the 0 position. Reading back also occurs in a loop. The clear bit also resets the read pointer.

ORB2_CHAN_TX_CNTL

[0x24 Transmit Control Register read -write]

DATA BIT	DESCRIPTION
31-24	Spare
23-16	Divisor [COM1-6]
11	FifoLoad
10	TxDataPolarity [set to 0]
9	TxFifoMuxControl
8	TxDataValidPolarity
7	TxClkMaskEn
6	TxClkPolarity
5	TxMode
4	TxDataOrder
3	TxFifoInterruptEn
2	TxInterruptEn
1	TxStartDMA (Tern,LS,HS) MasterIO (TLM)
0	TxStartReg (Tern) TxClkSuppress(HS)

Figure 22 PMC-BISERIAL-III Transmit Control Bit Map

Divisor[11-0] are the clock divisor select bits. The clock source is divided by a 12-bit counter. The output frequency is {reference / [2(n+1)]}, n≥1. The counter divides by N+1 due to counting from 0 to n before rolling over. The output is then divided by 2 to produce a square wave output.

The desired frequency of 1 MHz. Is achieved by selecting PLLA reference to 140 MHz, and a factor of 69. 2(N+1) = 140 => N = 69. Remember to convert to Hex before writing to the hardware.

FifoLoad is used to control the mode of the external FIFO's in COM7&8. With FifoLoad set low when reset is applied the offset will default to 127. With the FifoLoad set high the offset will be 1023 and the method of programming the offsets will be serial. If 127 or 1023 are used directly the FifoLoad pin can be set accordingly. If a user value is desired the FifoLoad pin will need to be low when Reset is applied. While FifoLoad is low the writes to the data port on the FIFO will program the PAE and PAF settings. Reads from the FIFO will return the programmed values. Take FifoLoad high after reset for normal operation.

TxDataPolarity [set to 0] The data polarity can be inverted. Normally leave set to '0'. Applies to COM7&8 only.

TxFifoMuxControl controls the path to load the external FIFO. Data can be moved from the TX internal FIFO to the external FIFO or from the RX state-machine to the External FIFO. When Low the TX path is selected, when high the RX path is selected. COM7&8

only.

TxDataValidPolarity when set causes the HW to invert the received Data Valid signal.
Ternary 0 = standard = S is active low during transmission 1 = inverted
LS / HS 0 = active high Valid, 1 = inverted valid signal expected
TLM [TargetGatePolarity] when 0 Gate is high in Type 1 and low in Type II. When 1 the Gate is inverted.

TxClkMaskEn when set the clock will be masked based on when data is active. Usually used with Packetized data. The clock will be bursted to provide one clock per data bit sent. The active edge is programmable with the TxClkPolarity. Some receive modes require an active clock [COM7&8].

TxClkPolarity
Ternary 0 = rising edge valid data, 1 = falling edge valid clock. 0 = standard operation
LS/HS 0 = falling edge valid data, 1 = rising edge valid data. 0 = standard operation
TLM 0 = falling edge data valid, 1 = rising edge data valid. Type 1 uses falling edge for Target [Rx], Type II uses rising edge for Target. Please note that this mode does require changing the Clock Polarity depending on the Type being used.

TxMode When set the transmitter uses the packetized programming and when cleared the transmitter uses the FIFO state to control transfer size. Applies to modes where packetized transmission is a part of the definition. (COM1,2,3,4,7,8)

TxDataOrder will use the natural order of the data from the FIFO or reverse the bits. Natural order when cleared and reversed when set. All modes use LSB first transmission. Natural order is D0 first D31 last. D0 = D0 on PCI bus.

TxFifoInterruptEn When set enables the Almost Empty FIFO Status to cause an interrupt to the host. When cleared the FIFO counts and FIFO status can still be used.

TxInterruptEn When set and a Tx State Machine interrupt function is defined an interrupt is caused to the host.
Ternary: When Register Data is enabled an interrupt is generated at the completion of the transmission.
TLM: Set when GateOut activity is detected and completed. Use as signal to update registers before next Target request is sent.
LS/HS: At the end of transmission in one of the non packetized modes.

TxStartDMA When set enables the state machine to start transmission based on programmed parameters, and the availability of Data in the Tx FIFO. The FIFO Almost Empty status is used for start-up. The Almost Empty level can be programmed to be the same or larger than the packet size when multiple 32 bit words are required per packet to avoid under-run situations.

MasterIO is used in TLM mode to enable the master transmitters. In the other modes the IO is enabled based on the transmission being enabled. With TLM due to the transmitted clock and valid and received data it is possible for the receiver to misinterprete the change from undriven to driven as a clock or enable. MasterIO should be enabled whenever this card is the master to avoid data line hazards.

TxStartReg: In Ternary mode there are two data paths. TxStartDMA controls the FIFO path and TxStartReg controls the Register based path. The register based path has higher priority than the FIFO based path. When the TxStartReg signal is asserted the Tx State Machine will begin sending register data at the next data boundary. For example, if the TxStartDMA has already been asserted and a packet is underway, the packet will be allow to complete and when it is time to start the next packet the register data will be used. Once the register data is being processed the Tx State Machine clears the TxStartReg. At the end of the register data the transmitter will return to using FIFO data if the DMA path is enabled.

In TLM the register start corresponds to a Master cycle starting. The programmed clocks are sent and data is received from the Target. The bit is cleared at the end of the acquisition.

TxClkSuppress: In the HS mode [COM7&8] there is a requirement to suppress the clock during transmission for a programmed number of clocks. Setting this bit will trigger the clock suppression mechanism. The bit is self clearing. Please note that the clocks are not synchronized to anything, just suppressed when this bit is enabled.

ORB2_CHAN_RX_LEN_ERR

[0x28 Receive Length Error Counter read - preload]

DATA BIT	DESCRIPTION
31-0	Length Error Count

Figure 23 PMC-BISERIAL-III Receive Length Error

In the packetized modes when errors occur the length counter is advanced. The count is advanced independent of the Status Disable state. Both “long” and “short” errors are counted.

The counter is read-writeable with a write causing a preload of the counter. Normally the preload feature is used to clear the counter. In some situations a negative count may be used to account for some start-up issues with a particular system. Positive counts can be used to track errors across multiple sessions of use. The count can be read at any time.

ORB2_CHAN_TX_REG0-7

[0x2C,30,34,38,3C,40,44,48 General Purpose Registers to support COM Functions]

DATA BIT	DESCRIPTION
31-0	Data

Figure 24 PMC-BISERIAL-III TX REG0-7

The 8 registers are used for different purposes for the Tern, TLM, LS and HS ports. The registers are read-writeable.

Ternary : the registers are used to store the secondary transmission path data. Register 0 is sent first, Register 7 is sent last. Bit 0 is sent first, 31 last from each word. The total bits sent are controlled with the ORB2_CHAN_TX_CNT register.

TLM: uses the register space for register files to support the master and target capabilities. There are 8 registers in each “file”. There are two files implemented per TLM port. The master file is configured to allow RW from the host and writing from the master SM. The target is configured to allow RW from the host and reading from the Target SM.

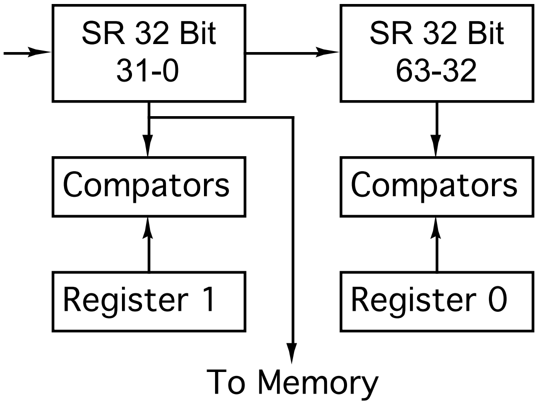
Register address 0 corresponds to the R/W port for the host to the Master Register File.
Register address 1 corresponds to the address pointer clear for writes to the Master Register File.
Register address 2 corresponds to the R/W port for the host to the Target Register File.
Register address 3 corresponds to the address pointer clear for writes to the Target register file.

The R/W capability is to allow for self test. In normal operation the Master register file is read by the host and written by the state-machine. The Target register file is written by the host and read by the state machine.

The Master State-machine can clear the address pointer to always load from address 0.
The Target State machine can read direct without needing to clear any pointers.
The Host can clear the pointer since a previous access may have left the register pointer within the register file not pointing to the base register.

The registers can be updated at any time. The transmitted data [Target] or received data [Master] may change while being accessed if the host accesses while the bus is active. The Master side is controlled by the host. Collisions are easy to avoid on this side. The interrupt can be used to know when a Target interaction has occurred and based on the system parameters and relatively low duty cycle of these messages have plenty of time to update prior to the next Master data request.

LS and **HS** use Register 0 and 1 to store the RX synchronization pattern. Register 0 stores the "upper" and Register 0 stores the "lower" bits.



Data is received into the Shift Registers. The data is based on 32 bit words. The first shift register captures the data and potentially sends it to be stored in memory. This is true in all modes. In the sync mode the data is also shifted through a secondary shift register to create a 64 bit wide word. The values stored into the TX Register 0 and TX Register 1 are used to set the sync pattern to check against. As each clock is received the new pattern held in the 64 bit shift register is compared against the stored value. If the values match the sync pattern has been found. The next bit into the lower shift register after sync has been found is the first bit to be stored. As described in the receiver controller section the sync pattern length to be used can be set by enabling different bytes to be tested.

Comparator definitions

Data Bits	Register Bits	RxSyncEn
7-0	Reg1 7-0	4
15-8	Reg1 15-8	5
23-16	Reg1 23-16	6
31-24	Reg1 31-24	7
39-32	Reg0 7-0	0
47-40	Reg0 15-8	1
55-48	Reg0 23-16	2
63-56	Reg0 31-24	3

Data is shifted from 31 toward 0 and from 63 toward 32 due being LSB first. With a 64 bit sync pattern everything is straight forward as the bits will line up. For shorter Sync patterns some care must be exercised in determining which bytes to use for comparison.

For an 8 bit pattern the first 8 bits of the first shift register should be used since the next bit after the 8 will be captured as data. Therefore Register 1 upper byte should be loaded with the pattern and RxSyncEn7 used to enable the pattern checking. For a 16 bit check the upper 16 bits of Register 1 would be used and the enables 7 and 6.

Start with the MS Byte of register 1 and work down to the LS Byte then continue with the MS Byte of register 0 and work down to the LS byte. Start with enable 7 and add the next lower one for each byte added.

Leaving enables not enabled has the effect of the HW always declaring a match for that byte effectively making it a “don’t care”. The don’t care feature may be useful in some situations where a partial sync pattern is desired; capturing data intended for multiple addresses etc.

Register offset 2 is used for storing the bit count to suppress. 3-0 are the valid locations allowing up to 16 clocks to be suppressed. The count is used to control the clock mask based on the Clock Suppression bit. HS mode only.

ORB2_CHAN_TX_CNT

[0x4C Length and Delay Register]

DATA BIT	DESCRIPTION
31-16	Delay
15-0	Length

Figure 25 PMC-BISERIAL-III Length & Delay Counts

In Packetized or single shot modes [TLM] the length is the transmitted length or expected received length of the data. In the Ternary and TLM modes this is in bits. In LS and HS the count is in 32 bit words. The count is +1, a length of 0x10 provides a transmitted or expected length of 0x11. The required range is 8 bits and the offset allows a range of 1-256 [bits or words].

In Packetized modes the delay between packets is also programmable. The delay is in terms of clocks in all modes. The minimum programmed delay is 3. The delay needs to be sufficiently long for the hardware to process status and potentially incorrect lengths at the end of received messages. At higher clock rates more delay is required than at lower clock rates.

Loop-back

The Engineering kit includes reference software, utilizing external loop-back tests.

The test set-up included PCIBPMC, ORB2, SCSI cable, and HDEterm68 to provide the loop-back. The Pin numbers are for the interconnections on the HDEterm68. The IO names can be used to accommodate a different set-up.

Signal	From	To	Signal
COM1 S+	pin 1	pin 5	COM2 S+
COM1 S-	pin 35	pin 39	COM2 S-
COM1 ONE+	pin 2	pin 6	COM2 ONE+
COM1 ONE-	pin 36	pin 40	COM2 ONE-
COM1 ZERO+	pin 3	pin 7	COM2 ZERO+
COM1 ZERO-	pin 37	pin 41	COM2 ZERO-
COM1 CLK+	pin 4	pin 8	COM2 CLK+
COM1 CLK-	pin 38	pin 42	COM2 CLK-
COM3 DATA+	pin 9	pin 12	COM4 DATA+
COM3 DATA-	pin 43	pin 46	COM4 DATA-
COM3 CLK+	pin 10	pin 13	COM4 CLK+
COM3 CLK-	pin 44	pin 47	COM4 CLK-
COM3 VALID+	pin 11	pin 14	COM4 VALID+
COM3 VALID-	pin 45	pin 48	COM4 VALID-
COM5 DATA+	pin 15	pin 18	COM6 DATA+
COM5 DATA-	pin 49	pin 52	COM6 DATA-
COM5 CLK+	pin 17	pin 20	COM6 CLK+
COM5 CLK-	pin 51	pin 54	COM6 CLK-
COM5 GATE+	pin 16	pin 19	COM6 GATE+
COM5 GATE-	pin 50	pin 53	COM6 GATE-
COM7 DATA+	pin 21	pin 24	COM8 DATA+
COM7 DATA-	pin 55	pin 58	COM8 DATA-
COM7 CLK+	pin 23	pin 26	COM8 CLK+
COM7 CLK-	pin 57	pin 60	COM8 CLK-
COM7 VALID+	pin 22	pin 25	COM8 VALID+
COM7 VALID-	pin 56	pin 59	COM8 VALID-
PAR0+	pin 27	pin 31	PAR4+
PAR0-	pin 61	pin 65	PAR4-
PAR1+	pin 28	pin 32	PAR5+
PAR1-	pin 62	pin 66	PAR5-
PAR2+	pin 29	pin 33	PAR6+
PAR2-	pin 63	pin 67	PAR6-
PAR3+	pin 30	pin 34	PAR7+
PAR3-	pin 64	pin 68	PAR7-

PMC Module Logic Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn1 Interface on the PMC-BiSerial-III.   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		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 26 PMC-BISERIAL-III Pn1 Interface

PMC Module Logic Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn2 Interface on the PMC-BiSerial-III.   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		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 27 PMC-BISERIAL-III Pn2 Interface

PMC Module Front Panel IO Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module IO Interface on the PMC-BiSerial-III. Installed for –FP and –FRP models. Also see the User Manual for your carrier board for more information. Standard ORB2 is –FP [no Pn4]

IO_0p (COM1 S+)	IO_0m (COM1 S-)	1	35
IO_1p (COM1 ONE+)	IO_1m (COM1 ONE-)	2	36
IO_2p (COM1 ZERO+)	IO_2m (COM1 ZERO-)	3	37
IO_3p (COM1 CLK+)	IO_3m (COM1 CLK-)	4	38
IO_4p (COM2 S+)	IO_4m (COM2 S-)	5	39
IO_5p (COM2 ONE+)	IO_5m (COM2 ONE-)	6	40
IO_6p (COM2 ZERO+)	IO_6m (COM2 ZERO-)	7	41
IO_7p (COM2 CLK+)	IO_7m (COM2 CLK-)	8	42
IO_8p (COM3 DATA+)	IO_8m (COM3 DATA-)	9	43
IO_9p (COM3 CLK+)	IO_9m (COM3 CLK-)	10	44
IO_10p (COM3 VALID+)	IO_10m (COM3 VALID-)	11	45
IO_11p (COM4 DATA+)	IO_11m (COM4 DATA-)	12	46
IO_12p (COM4 CLK+)	IO_12m (COM4 CLK-)	13	47
IO_13p (COM4 VALID+)	IO_13m (COM4 VALID-)	14	48
IO_14p (COM5 DATA+)	IO_14m (COM5 DATA-)	15	49
IO_15p (COM5 CLK+)	IO_15m (COM5 CLK-)	16	50
IO_16p (COM5 GATE+)	IO_16m (COM5 GATE-)	17	51
IO_17p (COM6 DATA+)	IO_17m (COM6 DATA-)	18	52
IO_18p (COM6 CLK+)	IO_18m (COM6 CLK-)	19	53
IO_19p (COM6 GATE+)	IO_19m (COM6 GATE-)	20	54
IO_20p (COM7 DATA+)	IO_20m (COM7 DATA-)	21	55
IO_21p (COM7 VALID+)	IO_21m (COM7 VALID-)	22	56
IO_22p (COM7 CLK+)	IO_22m (COM7 CLK-)	23	57
IO_23p (COM8 DATA+)	IO_23m (COM8 DATA-)	24	58
IO_24p (COM8 VALID+)	IO_24m (COM8 VALID-)	25	59
IO_25p (COM8 CLK+)	IO_25m (COM8 CLK-)	26	60
IO_26p (PAR0+)	IO_26m (PAR0-)	27	61
IO_27p (PAR1+)	IO_27m (PAR1-)	28	62
IO_28p (PAR2+)	IO_28m (PAR2-)	29	63
IO_29p (PAR3+)	IO_29m (PAR3-)	30	64
IO_30p (PAR4+)	IO_30m (PAR4-)	31	65
IO_31p (PAR5+)	IO_31m (PAR5-)	32	66
IO_32p (PAR6+)	IO_32m (PAR6-)	33	67
IO_33p (PAR7+)	IO_33m (PAR7-)	34	68

Figure 28 PMC-BISERIAL-III FRONT PANEL Interface

Applications Guide

Interfacing

The pin-out tables are displayed with the pins in the same relative order as the actual connectors. 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. Differential interface devices provide some immunity from, and allow operation when part of the circuit is powered on and part is not. It is better to avoid the issue of going past the safe operating areas by powering the equipment together and by having a good ground reference.

Keep cables short. Flat cables, even with alternate ground lines, are not suitable for long distances. In addition series resistors are used and can be specified to be something other than the 0 ohm standard value. The connector is pinned out for a standard SCSI II/III cable to be used. It is suggested that this standard cable be used for most of the cable run or an equivalent with proper twisted pairs and shielding.

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 particular device's rated voltages.

Construction and Reliability

PMC Modules were conceived and engineered for rugged industrial environments. The PMC-BiSerial-III is constructed out of 0.062 inch thick high temperature ROHS compliant material.

The traces are matched length from the FPGA ball to the IO pin. The options for front panel and rear panel are isolated with series resistor packs to eliminate bus stubs when one of the connectors is not in use.

Surface mounted components are used.

The PMC Module connectors are keyed and shrouded with Gold plated pins on both plugs and receptacles. They are rated at 1 Amp per pin, 100 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable.

The PMC is secured against the carrier with the connectors and front panel. If more security against vibration is required the stand-offs can be secured against the carrier.

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

Thermal Considerations

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

Warranty and Repair

Please refer to the warranty page on our website for the current warranty offered and options. http://www.dyneng.com/warranty.html

Service Policy

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

Out of Warranty Repairs

Out of warranty repairs will be billed on a material and labor basis. The current minimum repair charge is $125. 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 St. Suite C
Santa Cruz, CA 95060
831-457-8891
831-457-4793 fax
support@dyneng.com

Specifications

Logic Interface:	PMC Logic Interface [PCI] 32/33
Digital Parallel IO:	Multiple COM standards supported including Ternary, LS, HS, TLM.
CLK rates supported:	PLLA is programmed to 140 Mhz and divided locally for COM1-6. PLLB is used directly for COM7&8. PLL reference is 50 Mhz. COM7&8 [HS] are tested to 85 Mhz. COM1-4 [Ternary, LS] are tested 1-10 Mhz. COM 5&6 [TLM] are tested at 1 and 2 Mhz.
Software Interface:	Control Registers, IO registers, IO Read-Back registers, FIFO. R/W, 32 bit boundaries.
Initialization:	Programming procedure documented in this manual
Access Modes:	LW to registers, read-write to most registers
Access Time:	Frame to TRDY 121 nS [4 PCI clocks] or burst mode DMA – 1 word per PCI clock transferred.
Interrupt:	Each COM port has independently programmable interrupt sources, DMA interrupts included.
Onboard Options:	All Options are Software Programmable
Interface Options:	68 Pin SCSI III connector at front bezel. Rear IO by special request.
Dimensions:	Standard Single PMC Module.
Construction:	Multi-Layer Printed Circuit, Through Hole and Surface Mount Components.
Temperature Coefficient:	2.17 W/^{O}C for uniform heat across PMC
Power:	TBD mA @ 5V

Order Information

standard temperature range -40-85°C
PMC-BiSerial-III-ORB2 PMC Module with 34 IO channels divided among 4 “COM” types including Ternary, LS, HS, and TLM
http://www.dyneng.com/pmc_biserial_III.html

Order Options:

Pick One
–FP for front panel IO only [default if no selection made]
-RP for rear panel IO PN4 only
-FRP for both IO connections
Shown for reference. ORB2 selection determines [-FP]

Pick any combination to go with IO
-CC to add conformal coating
-CT to reduce to commercial Temp [0 - 70]
-TS to add thumbscrew option – standard is latch block at SCSI connector

Related:
PCIBPMC: PCI to PMC adapter to allow installation of PMC-BiSerial-III into a PCI system.
http://www.dyneng.com/pciBpmc.html

PCIeBPMCX1: PCIe to PMC adapter to allow installation of PMC-BiSerial-III into a PCIe system.
http://www.dyneng.com/pciebpmcx1.html

HDEterm68: 68 position terminal block with two SCSI II/III connectors. PMC-BiSerial-III compatible.
http://www.dyneng.com/HDEterm68.html

HDEcabl68: SCSI II/III cable compatible with FPIO on PMC Parallel IO.
http://www.dyneng.com/HDEcabl68.html

PMC BiSerial III Eng Kit : HDEterm68-MP, HDEcabl68, Windows Driver software, reference schematics. Recommended for first time purchases.
http://www.dyneng.com/pmc_parallel_TTL.html