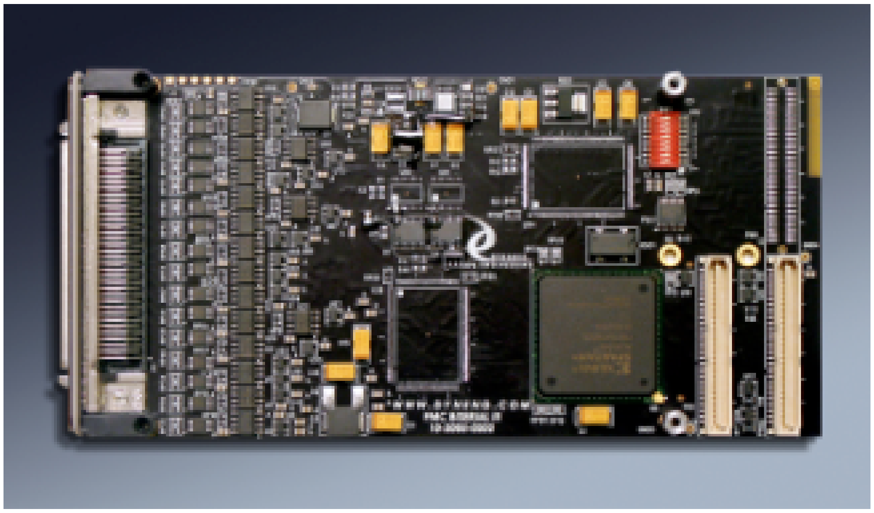
DYNAMIC ENGINEERING
150 DuBois St. Suite C, Santa Cruz, CA 95060
831-457-8891   Fax 831-457-4793
http://www.dyneng.com
sales@dyneng.com
Est. 1988

User Manual

PMC-BiSerial-III-NASA1

Digital Parallel Interface
PMC Module
LADEE – LLST (Tx & Rx) [LVDS]
NMS (Tx & Rx) [RS485]
UART (Tx & Rx) [RS485]
Uplink & Downlink Manchester IF [RS485]
Parallel Port [RS485]



Manual Revision A
Corresponding Fab: 10-2005-0204
FLASH 0x0C01

PMC-BiSerial-III-NASA1

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.

Revised 1/11/10

Table of Contents

PRODUCT DESCRIPTION 6

ADDRESS MAP 14

PROGRAMMING 16

Base Register Definitions 19

NASA1_BASE_BASE 19
NASA1_BASE_ID 20
NASA1_BASE_STATUS 21
NASA1_BASE_PARDIR 21
NASA1_BASE_PARDAT 22
NASA1_BASE_PARDATREGIN 22
NASA1_CHAN_CNTRL 23
NASA1_CHAN_STATUS 27
NASA1_CHAN_TX_FIFO_COUNT 33
NASA1_CHAN_RD_DMA_PNTR 33
NASA1_CHAN_RX_FIFO_COUNT 34
NASA1_CHAN_FIFO 34
NASA1_CHAN_TX_AMT_LVL 35
NASA1_CHAN_RX_AFL_LVL 35
NASA1_CHAN_TX_CNTRL 36
NASA1_CHAN_TX_WORD_COUNT 39
NASA1_CHAN_TX_FIFO_WORD_COUNT 40
NASA1_CHAN_TX_AMT_LVL_TOTAL 40
NASA1_CHAN_TX_READY_COUNT 41
NASA1_CHAN_RX_CNTRL 42
NASA1_CHAN_RX_WORD_COUNT NASA1_CHAN_RX_SYNC_PATTERN 46
NASA1_CHAN_RX_FIFO_WORD_COUNT 48
NASA1_CHAN_RX_AFL_LVL_TOTAL 48
NASA1_CHAN_EXT_FIFO_WORD_COUNT 49
NASA1_CHAN_TX_WORD_COUNT 49
NASA1_CHAN_RX_WORD_COUNT 50

LOOP-BACK 51

PMC MODULE LOGIC INTERFACE PIN ASSIGNMENT 52

PMC MODULE LOGIC INTERFACE PIN ASSIGNMENT 53

PMC MODULE REAR IO INTERFACE PIN ASSIGNMENT 54

APPLICATIONS GUIDE 55

Interfacing 55

Construction and Reliability 56

Thermal Considerations 56

Warranty and Repair 57

Service Policy 57

Out of Warranty Repairs 57

SPECIFICATIONS 58

ORDER INFORMATION 59

List of Figures

FIGURE 1 PMC-BISERIAL-III-NASA1 LLST – LADEE TIMING 8
FIGURE 2 PMC-BISERIAL-III-NASA1 NMS TIMING 8
FIGURE 3 PMC-BISERIAL-III-NASA1 BLOCK DIAGRAM 12
FIGURE 4 PMC-BISERIAL-III BASE ADDRESS MAP BASE 14
FIGURE 5 PMC-BISERIAL-III CHANNEL ADDRESS MAP 15
FIGURE 6 PMC-BISERIAL-III CONTROL BASE REGISTER BIT MAP 19
FIGURE 7 PMC-BISERIAL-III ID AND SWITCH BIT MAP 20
FIGURE 8 PMC-BISERIAL-III STATUS PORT BIT MAP 21
FIGURE 9 PMC-BISERIAL-III PARALLEL PORT DIRECTION BIT MAP 21
FIGURE 10 PMC-BISERIAL-III PARALLEL PORT DATA BIT MAP 22
FIGURE 11 PMC-BISERIAL-III PARALLEL PORT DATA BIT MAP 22
FIGURE 12 PMC-BISERIAL-III CHANNEL CONTROL REGISTER 23
FIGURE 13 PMC-BISERIAL-III CHANNEL STATUS PORT 27
FIGURE 14 PMC-BISERIAL-III WRITE DMA POINTER REGISTER 32
FIGURE 15 PMC-BISERIAL-III TRANSMIT DMA FIFO DATA COUNT PORT 33
FIGURE 16 PMC-BISERIAL-III READ DMA POINTER REGISTER 33
FIGURE 17 PMC-BISERIAL-III RECEIVE DMA FIFO DATA COUNT PORT 34
FIGURE 18 PMC-BISERIAL-III RX/TX FIFO PORT 34
FIGURE 19 PMC-BISERIAL-III TRANSMIT ALMOST EMPTY LEVEL REGISTER 35
FIGURE 20 PMC-BISERIAL-III RECEIVE ALMOST FULL LEVEL REGISTER 35
FIGURE 21 PMC-BISERIAL-III CHANNEL TRANSMITTER CONTROL REGISTER 36
FIGURE 22 PMC-BISERIAL-III TRANSMIT LW PER PACKET COUNT PORT 39
FIGURE 23 PMC-BISERIAL-III TARGET DATA COUNT PORT 40
FIGURE 24 PMC-BISERIAL-III TX AMT LEVEL CONTROL PORT 40
FIGURE 25 PMC-BISERIAL-III TX READY COUNT CONTROL PORT 41
FIGURE 26 PMC-BISERIAL-III CHANNEL RECEIVER CONTROL REGISTER 42
FIGURE 27 PMC-BISERIAL-III RX DATA SYNC PORT 46
FIGURE 28 PMC-BISERIAL-III RX SM DATA COUNT PORT 48
FIGURE 29 PMC-BISERIAL-III RX AFL LEVEL CONTROL PORT 48
FIGURE 30 PMC-BISERIAL-III RX EXTERNAL FIFO LEVEL PORT 49
FIGURE 31 PMC-BISERIAL-III TX TOTAL FIFO COUNT 49
FIGURE 32 PMC-BISERIAL-III RX TOTAL FIFO COUNT 50
FIGURE 33 PMC-BISERIAL-III PN1 INTERFACE 52
FIGURE 34 PMC-BISERIAL-III PN2 INTERFACE 53
FIGURE 35 PMC-BISERIAL-III FRONT PANEL INTERFACE 54

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 systems, and is adding more as new solutions are requested by our clients.

The NASA1 version of the PMC-BiSerial-III is the 19th implementation to date.

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.

The front and rear options are isolated to keep trace lengths direct and short. The signal path is matched length from the FPGA ball to the connector pin, and impedance controlled for RS-485 or LVDS systems.

PMC-BiSerial-III-NASA1 is a “clientized” version of the standard PMC-BiSerial-III board. “NASA1” is set to use the LVDS and RS-485 standards, has front panel IO, and supports multiple channels. The PLL is referenced to 40 MHz. and can be programmed with new .JED files using the driver. PLLA is used for the LADEE / LLST function and PLLB is used for the NMS function. PLLC is used as a master reference for the UART channel and PLLD is used for the Manchester Uplink and Downlink function. Local dividers allow the UART and Uplink/Downlink channels to change frequencies without reprogramming the PLL.

The driver is supplied with NASA1.jed which is a precompiled file used to program the PLL to the standard reference frequencies. Alternate frequencies can be generated with the Cypress Semiconductor PLL programming tool.

IO is mixed with both LVDS and RS485 transceivers utilized to support the different interfaces provided. The LVDS IO requires 3.3V. On this design the RS485 is implemented with 3.3V compatible devices.

The hardware design is hierarchical in nature and “channelized” in concept. The base level of the hardware includes the PCI interface and DMA arbitration, Parallel port, and board level status. 4 channels are implemented for the 4 major functions of the board. Each channel can request the PCI bus using DMA for it's TX and/or RX function. The arbitration mechanism interacts with the PCI bus and the channels to provide access to the PCI bus for Burst IN [to support TX] and Burst Out to support RX.

Each channel is assigned 20 LW addresses with R/W capability to each. The Base also has 20 LW addresses assigned. Not all addresses are used. Where possible similar functions are on the same relative address for each of the channels. The Windows® driver supports the channelized concept allowing independent threads to operate each of the channels, or as a combined controlling set of code. If you are writing your own driver a channelized approach can be used or a flattened approach where all of the addresses are considered part of the base. The HW is controlled in the same way in either case. The channelized approach lends itself to more modular SW and ease of porting functions in and out as requirements change.

Channel 0 implements the "LLST" interface. LLST is a receiver designed to capture data from LADEE. A LADEE compatible transmitter is also provided. The signals are CLK, DATA, FRMRDY_N, and DVAL_N. The channel is organized with independent transmitter and receiver to allow for full duplex operation and loop-back testing.   Two 8Kx32 FIFO's with DMA support are included in this channel.

The data length is controlled via register for transmission. The length is set with the number of LW's to send and how many bytes in the last LW. The 32 bit register can control very long lengths with byte control in this way. For the RX function the transmitter DVAL signal is used to determine the length of data received. The holding register is cleared between moving to the holding FIFO and loading the next byte to provide a zero filled last word written to FIFO for non-LW byte lengths. IO0-7 are utilized for the LLST interface.

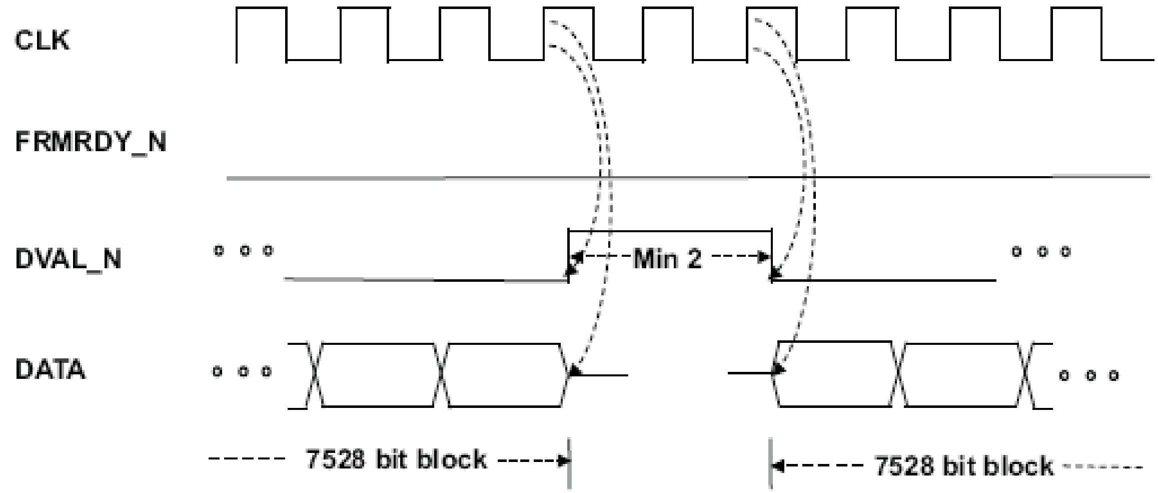


Figure 1 PMC-BISERIAL-III-NASA1 LLST – LADEE Timing

The basic interface timing is shown for the LLST. Data changes on the falling edge and is stable on the rising edge of the clock. The receiver asserts Frame Ready to allow the transmitter to start. The transmitter will complete the Frame even if the Frame Ready signal is deactivated. The LLST design asserts FrameReady, waits for DVAL and deasserts FrameReady. The bit count is programmable on a byte boundary.

Channel 1 implements the "NMS" interface for both transmit and receive functions. The signals involved with NMS are: FRMn, CLK, Data, and RDY. The signals are RS-485 compatible and operate with programmable clock rates below 10 MHz. PLLB is used to provide the TX rate. A local DCM is used to boost the TX programmed rate to sample the RX side. The clock is bursted, and the high speed sample rate clock allows the state-machine to operate continuously while the CLK signal is not continuous. Two 8Kx32 FIFO's with DMA support are included in this channel. IO8-15 are used for the NMS function.

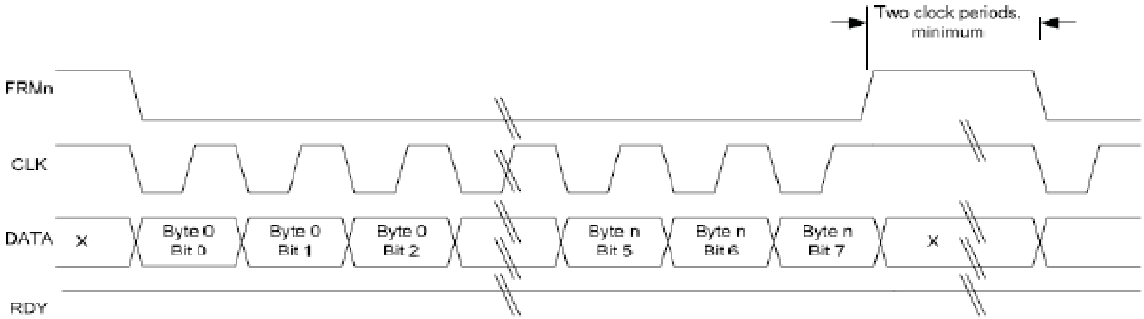


Figure 2 PMC-BISERIAL-III-NASA1 NMS Timing

RDY is asserted by the receive side of the interface whenever the transmitter has permission to send data. When RDY is de-asserted the Transmitter will stop transmitting data within 15 clocks and hold with FRMn asserted until RDY is reasserted. The NASA1 design uses byte boundaries to test the RDY signal to guarantee the hold within 15 clocks requirement and to keep the interface easy to manage. On the receive side the RDY signal is disabled if the FIFO goes almost full to allow for the transmitter delay in responding.

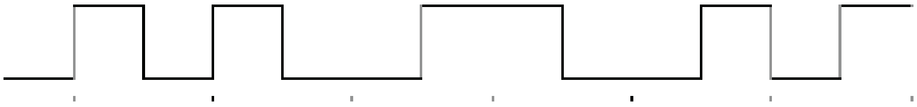
Data changes on the falling edge of the clock and is stable on the rising edge. FRMn is asserted with 1/2 period of set-up to the first clock and 1/2 period of hold for the last clock. The clock is bursted with the off state "high".

Channel 2 provides a UART transmitter and receiver. The reference oscillator is used to provide the base clock rate. A local programmable divider provides the x16 RX reference and a fixed ÷16 function provides the TX rate. The UART's are supported with 1Kx32 FIFO's and a selection of Byte and Packet modes. In Byte mode data is sent from the FIFO using only the bottom byte and is intended for standard low utilization "set and forget" operation. Packet mode uses 4 bytes per LW and a programmable length [TX] to send larger quantities of data. The count is programmable on a byte basis. Programmable Parity [on/off odd/even], number of stop bits and interrupts, plus the baud rate selection provide for most situations.

Standard UART operation with LSB first, start bit ['0'], programmable parity [off/on odd/even], programmable Baud rate, programmable 1 or 2 stop bits ['1'] and a marking state of '1'. The IO is RS-485 and use IO 16,17.

Channel 3 provides an UpLink [transmitter] and DownLink [receiver]. Both interfaces use Manchester encoding for communication and have programmable transfer rates. The UpLink and DownLink have separate 1Kx 32 FIFO's for storage. DMA and target accesses are supported. IO 18 , 19 are used and implemented with RS-485 devices.

PLLD is programmed with the upper reference frequency and a local divider used to generate the RX sampling rate. The TX transmission rate is a fixed division from the RX programmed rate. The TX rate is 2x the Bit rate and 1/4 of the RX rate. Program the RX rate to be 8x the expected bit rate. The Manchester encoding scheme used is Bi-Phase-L which defines a '1' as a 1-0 and a '0' as a 0-1 with the transition in the middle of the bit period.



The initial state of the bus is static and '0' in the case shown. The first bit period is a '1' with a 1-0 transition at mid period. The '1' is followed by a '1' and then a '0' with the bit transition going 0-1. Basically the first half period is the bit value and the second half period is the opposite. This guarantees at least 1/2 frequency bit transitions and a balanced line charge level. The hardware has software options to invert the data allowing Bi-Phase-L and Bi-Phase-H data types to be generated or received.

The Transmitter is held in Idle until enabled. In the Idle state the output will be '0'. Based on a software selection the transmitter can "Free Wheel" with a "5555" pattern which comes out as 00110011 and looks like a half rate clock or as a static value; either set to '0' or holding the second half period value from the previous bit.

The receiver can be programmed to expect an Idle pattern or a static value. With the idle pattern it is recommended to use the sync pattern to start capture and avoid a FIFO full of the Idle pattern.

In static mode the line can be in either '1' or '0' state and the hardware will compensate to properly synchronize on the first bit. The expected first bit is programmed into the sync register in position 31. For longer sync patterns, the boundary can be 8,16,24 or 32 bits with the msb used as the expected first bit to sync with.

An aside: Bit level synchronization is about properly decoding the Manchester code. Pattern synchronization is about filtering messages or Idle patterns.

Base is the collection of non-channelized assets on the card including the parallel port, I^2C interface for the PLL, switch read-back, and board level status. The PLL is programmed via the base register using several bits to create a SW clock, Data etc. to control the PLL. The Dynamic Drivers have utilities to load .JED files [output from PLL configuration utility] to the PLL. Utilities also are available to read the contents back and check against the .JED reference.

The parallel port is designed to allow the user to "do whatever" with the otherwise unused bits – IO 33 – 20. Three registers allow the user to set the direction [in or out] for each bit independently, set the output bit values, and to read back from the direction, dataout definition and the IO lines directly. The terminations are automatically set to be enabled when in receive configuration and off when in transmit configuration. The port defaults to the read configuration. RS-485 transceivers are used on this port.

The Transmit and Receive sides of each channel can be used in pairs for loop-back testing. HDEterm68 http://www.dyneng.com/HDEterm68.html can be used as a breakout for the IO. The HDEcabl68 provides a convenient cable. http://www.dyneng.com/HDEcabl68.html Custom cables can be manufactured to your requirements. The loop-back IO definitions are toward the end of this manual. Please contact Dynamic Engineering with your specifications.

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 [FLASH 0101]. Please contact Dynamic Engineering with your custom requirements. NASA1 is design number "C" for the PMC-BiSerial-III with a corresponding FLASH of 0Cxx.

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 [16 Mbps rated] RS-485 parts are used that are compatible with the 3.3V.

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

Each of the IO has 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.

For the NASA1 design the direction and termination options are pre-programmed within the FPGA to match the IO definition.

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 DIODE's allow for more flexible operation and power sequencing.

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.

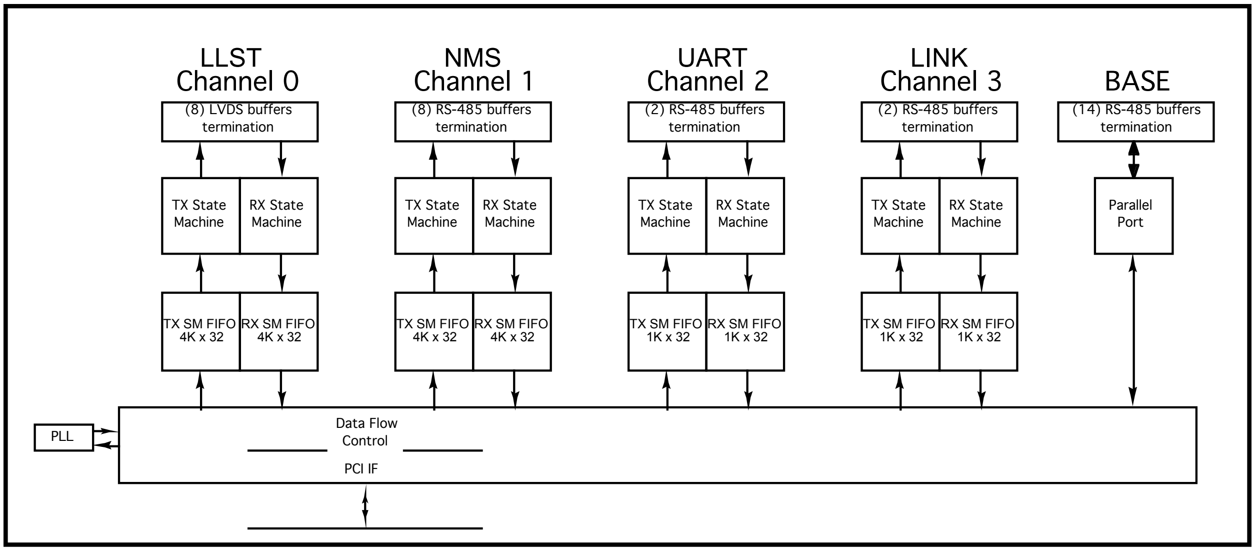


Figure 3 PMC-BISERIAL-III-NASA1 Block Diagram

PMC BiSerial III NASA1 features four channels each with Transmit and Receive capabilities. Each channel has separate DMA support. The internal block RAM is configured to provide FIFO's to support the DMA and IO transfer process.

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

The DMA programmable length is 4 Gbytes => 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. A single interrupt can control the entire transfer. Head to tail operation can be programmed with two memory spaces and two interrupts per loop.

Each channel is independent with complete DMA interfaces for each channel. The separate DMA engines are tied together through a local arbitration unit which can handle 16 channels. The NASA1 design uses 8 channels [4 TX and 4 RX]. The arbitration unit takes care of negotiation for the PCI bus and passing control to the channel that needs service. By having complete DMA engines, each can be programmed for the entire transfer that channel is going to perform. The hardware can automatically control when the transfer is implemented and then alert the SW once the transfer has been implemented.

The channelized approach is superior to single DMA systems since the CPU side has only a few interrupts to deal with and a lower "Real Time" component since the data flow is largely self controlled. In a single DMA system with multiple functions, the CPU will have to service an interrupt each time a function requires a transfer and then program the controller for that transfer. Since multiple functions are competing for the same resources the transfer sizes will necessarily be shorter leading to more interrupts and more overhead.

Address Map

```
Function                            Offset
// PMC BiSerial III NASA1 definitions
#define NASA1_BASE_BASE             0x0000 // 0 NASA1Base Base control register
#define NASA1_BASE_PLL_WRITE        0x0000 // 0 NASA1Base Base control register
#define NASA1_BASE_PLL_READ         0x0000 // 0 NASA1Base base control register
#define NASA1_BASE_USER_SWITCH 0x0004 // 1 NASA1Base User switch read port  DIP switch read
#define NASA1_BASE_XILINX_REV       0x0004 // 1 NASA1Base Xilinx revision read port
#define NASA1_BASE_XILINX_DES       0x0004 // 1 NASA1Base Xilinx design read port

#define NASA1_BASE_STATUS           0x0008 // 2 NASA1Base status Register offset
#define NASA1_BASE_PARDIR           0x0010 // 4 NASA1 Base Parallel Port Direction Register
#define NASA1_BASE_PARDAT           0x0014 // 5 NASA1 Base Parallel Port Parallel out, IO in
#define NASA1_BASE_PARDATREGIN 0x0018 // 6 NASA1 Base Parallel Port Register Data read-back
```

Figure 4 PMC-BISERIAL-III Base Address Map Base

The address map provided is for the local decoding performed within PMC-BiSerial-III. 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.

There is an additional Address Map for the Channels. For the channel address maps when using in a flat system be sure to add the channel offset to the offset shown in the table. Please note the definitions in the register bit map section for each address as there may be some differences between channel functions in the bit definitions.

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

VendorId = 0x10EE and the CardId = 0x0040 for the PMC-BiSerial-III-NASA1.

The NASA1 design has 4 channels implemented. The BASE contains the common elements of the design, while the Channels have the IO specific interfaces. The BASE starts at the card offset. Channel 0 starts at register 20.

Section	Register Address Range (starting Hex address)	Port name
Base	0-19 (0x0000)	PLL, Switch, Status
Channel 0	20-39 (0x0050)	NASA1 LLST Tx & Rx
Channel 1	40-59 (0x00A0)	NASA1 NMS Tx & Rx
Channel 2	60-79 (0x00F0)	NASA1 UART Tx & Rx
Channel 3	80-99 (0x0140)	NASA1 Uplink/Downlink Tx & Rx

```
Function                                       Offset from Channel Base Address
// PMC BiSerial III NASA1 Channel definitions
#define NASA1_CHAN_CNTRL                    0x00000000  //0 General control register

#define NASA1_CHAN_STATUS                   0x00000004  //1 Interrupt status port
#define NASA1_CHAN_INT_CLEAR                0x00000004  //1 Interrupt clear port

#define NASA1_CHAN_WR_DMA_PNTR              0x00000008  //2 Write DMA physical PCI address  reg
#define NASA1_CHAN_TX_FIFO_COUNT            0x00000008  //2 TX FIFO count read port

#define NASA1_CHAN_RD_DMA_PNTR              0x0000000C  //3 Read DMA physical PCI address reg
#define NASA1_CHAN_RX_FIFO_COUNT            0x0000000C  //3 RX FIFO count port including pipeline

#define NASA1_CHAN_FIFO                     0x00000010  //4 FIFO single word access RW
#define NASA1_CHAN_TX_AMT_LVL               0x00000014  //5 TX AMT level register RW, DMA FIFO
#define NASA1_CHAN_RX_AFL_LVL               0x00000018  //6 RX AFull level reg RW, DMA FIFO

#define NASA1_CHAN_TX                       0x0000001C  //7 TX control register
#define NASA1_CHAN_TX_WORD_COUNT            0x00000020  //8 TX count
#define NASA1_CHAN_TX_FIFO_WORD_COUNT 0x00000024  //9 TX StateMachine FIFO count TX
#define NASA1_CHAN_TX_AMT_LVL_TOTAL         0x00000028  //10 TX AMT level control for interrupt

#define NASA1_CHAN_RX                       0x00000034  //13 RX control register
#define NASA1_CHAN_RX_SYNC_PATTERN          0x00000038  //14 RX Sync Pattern for DownLink
#define NASA1_CHAN_RX_FIFO_WORD_COUNT 0x0000003C  //15 RX StateMachine FIFO count  RX
#define NASA1_CHAN_RX_AFL_LVL_TOTAL         0x00000040  //16 RX AFull level control for interrupt

#define NASA1_CHAN_EXT_FIFO_WORD_COUNT 0x00000044  //17 Spare for NASA1
#define NASA1_CHAN_TX_WORD_COUNT            0x00000048  //18 TX total count. Spare for NASA1
#define NASA1_CHAN_RX_WORD_COUNT            0x0000004C  //19 RX total count Spare for NASA1
```

Figure 5 PMC-BISERIAL-III Channel Address Map

Programming

Programming the PMC-BISERIAL-III-NASA1 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-NASA1 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-NASA1. The PLL will need to be programmed to use the NASA1 function. The Cypress CyberClocks software can be used to create new .JED files if desired. PLLA – PLLD will need to be programmed based on the desired LLST, NMS IO rates and the reference rates for the UART and Manchester function.

The driver comes with NASA1.jed which has the default rates [MHz] of 40 PLLA, 8.25 PLLB, 14.7456 PLLC, and 8 PLLD.

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. The reference application software also includes XLATE.c which converts the .JED file from the CyberClocks tool to an array that can be programmed into the PLL.

The IO direction and termination are hardwired in this design. The ports are unidirectional and initialization is simplified with this approach.

The control bits will select how the data is transmitted – Byte ordering, size of transfer etc.

For Windows™ and Linux systems the Dynamic Drivers[1] 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 NASA1 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

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

the NASA1 to start the DMA. Please refer to the Burst IN and Burst Out register discussions.

DMA should be set-up before starting the channel 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.

The Ready Count register can be programmed to make the HW wait until there is at least the Ready Count number of positions filled in the FIFO's before starting. When larger non-stop transfers are desired an amount close to capacity is recommended.

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".

Other DMA notes: When DMA is operating in the receive direction => writing to system memory, the PCI control signals can be used to indicate how much data is being transferred. Frame and IRDY remain asserted until the end of the requested length unless the Host asserts a STOP. In most cases a 256 word transfer is accomplished per DMA transfer with the hardware automatically generating new requests to move the data from the FIFO to the host memory. Most Bridge devices have FIFO buffers deep enough to allow for the full size DMA transfer to be accommodated resulting in a pipelined large word count transfer taking place.

In some cases the Bridge may need to be reprogrammed to do larger transfers to facilitate obtaining full bandwidth on the bus.

When operating as a transmitter the Hardware does a read from system memory and uses the same control signals to accomplish the transfer. The PMC BiSerial III NASA1 can handle back-to-back max size transfers. In some cases the Bridge may need to be reprogrammed to do larger transfers to facilitate obtaining full bandwidth on the bus.

Since the request from the NASA1 must pass through the bridges to the host prior to the host sourcing data, and since there is a delay in passing through the bridge, the transfer size is reduced to the Bridge prefetch length.

In addition while the host PCI controller can adsorb data at full length repeatedly, the same controller will frequently assert STOP after 16 words drastically shortening the

DMA transfer. Adjusting the default for the PCI controller may also be necessary to reach proper operational data flow. The NASA1 hardware has no control over the bridge or host PCI controller.

By moving the PMC BiSerial III NASA1 to the first PCI bus – attached to the PCI controller all bridges are bypassed allowing 16 word transfers in the Transmit direction and 256 in the Receive direction. With a burst size of 16 the overall throughput is better than 25% which is enough to sustain 1 channel at 8.25 MLW/sec.

This issue is computer dependent. Your results may be different for length of DMA achieved without making changes to the bridges between the PMC device and the system memory. BIOS can program the Bridges to operate and some may have FLASH attached to allow user "autoloading of parameters" into the configuration space control registers.

We are working on an autoloading utility for the bridges in the path and will soon offer that with our hardware products to solve the performance issue. We are also researching the PC Chip-set issue with the PCI master bridge to see what we can do to correct that as well.

Base Register Definitions

NASA1_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 6 PMC-BISERIAL-III Control Base Register Bit Map

This is the base control register for the PMC BiSerial III NASA1. 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 40 MHz.

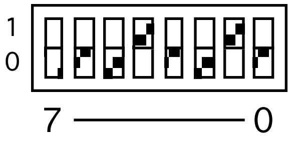
NASA1_BASE_ID

[$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 7 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 NASA1 design is 0x0C 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.

NASA1_BASE_STATUS

[$08 Board level Status Port read only]

DATA BIT	DESCRIPTION
31-12	Set to ‘0’
11-8	Reserved for PLL status
7-4	set to ‘0’
3	Masked Channel 3 Interrupt
2	Masked Channel 2 Interrupt
1	Masked Channel 1 Interrupt
0	Masked Channel 0 Interrupt

Figure 8 PMC-BISERIAL-III Status Port Bit Map

Channel Interrupt – The local masked interrupt status from the channel. Each channel can have different interrupt sources. DMA Write or DMA Read or IntForce or TX/RX request are typical sources. Polling can be accomplished using the channel status register and leaving the channel interrupt disabled.

NASA1_BASE_PARDIR

[$10 Board level Status Port R/W]

DATA BIT	DESCRIPTION
31-14	Set to ‘0’
13-0	Set to Transmit on corresponding Parallel Port Bit

Figure 9 PMC-BISERIAL-III Parallel Port Direction Bit Map

The Parallel Port provides 14 undefined IO which can be treated as independent bits or as a group. Each bit within the PARDIR register selects the Transmit or Receive orientation of the transceiver attached to that port. Bit zero corresponds to bit (0) of the parallel port and so forth. Please note that bit (0) is IO20 at the connector level. The hardware remaps the register zero aligned definitions to the IO aligned to bit 20.

The default is received. When in receive the termination is applied to the IO. When in transmit the termination is disabled.

NASA1_BASE_PARDAT

[$14 Parallel Port Data Register and IO R/W]

DATA BIT	DESCRIPTION
31-14	Set to '0'
13-0	Data Register for output definition

Figure 10 PMC-BISERIAL-III Parallel Port Data Bit Map

Each bit within the PARDAT register determines the state of any lines set to transmit. Lines set to receive are not affected by this register. Reading back from this register provides the IO side of the port. The register values may not match the IO values. For example if 1/2 of the bus is set to receive and 1/2 to transmit the receive side may not match the transmit register definition.

NASA1_BASE_PARDATREGIN

[$18 Parallel Port Data Register read-back read only]

DATA BIT	DESCRIPTION
31-14	Set to '0'
13-0	Data Register read-back

Figure 11 PMC-BISERIAL-III Parallel Port Data Bit Map

This read only port allows read-back of the Data output register. To do a read-modify-write cycle on the parallel data, read this port, then write back to the PARDAT port.

Channel Register Definitions

Channel Bit Maps

The NASA1 design has 4 channels. 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.

NASA1_CHAN_CNTRL

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

Channel Control Register

Data Bit	Description
31-24	spare
23-16	UART Bit Rate
15	Direction TX/RX – reserved
14	ExtFifoMux RX/TX -- reserved
13	ExtFifoEn -- reserved
12	ExtFifoLoad -- reserved
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	RX FIFO Reset
0	TX FIFO Reset

Figure 12 PMC-BISERIAL-III channel Control Register

There are differences in some of the bit definitions based on channel.

FIFO TX/RX Reset: When set to a one, the transmit and/or receive FIFO’s 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.

UART Bit Rate: is used for the UART channel only. The 12 bit field allows the channel software to divide the PLL reference to a user specified frequency. Set PLLC to rates which can be divided to the desired frequencies. Use the local divisor to select the operational rate. Use the formula /2N. The divider uses the programmed rate to terminate and restart counting. N = the 1/2 period rate. This results in a square wave output for the state-machines to work with.

The UART channel uses a 16x clock for the receive side. The Transmit side is automatically divided down from the Rx reference rate selected. Program the RX reference rate = 16x the desired Tx rate. For example if PLLC is set to the default of 14.7456 MHz, then setting the divisor to 0x0001 will create a reference of 7.3728 for the Rx and provide data at 460.8 KHz on the Tx side.

TX	RX	Divisor
460.8 K	7.3728 M	1
230.4 K	3.6864 M	2
115.2 K	1.8432 M	4
57.6 K	921.6 K	8
38.4 K	614.4 K	12
19.2 K	307.2 K	24
9600	153.6 K	48

Please note that the receiver for the UART function is designed to be adaptable to a range of frequencies. The TX side is programmed to the desired frequency [exactly]. What you program is what you get. The receive side can live with some variation – jitter or baseline frequency. The larger the offset the lower the jitter than can be handled.

The UART uses the data transitions received to re-center the sampling point which will allow for a range of received frequencies. The interface is data dependent for the adaptability. In the worst case with no parity and only the start bit transition [0x00 for data] the 16x clock would need to be off far enough to get the wrong value. Depending on the valid time of the data within the bit period the clock rate can cause a shift of close to 2 periods per bit and still get the correct data. 7-15% variance can be handled by static data, and more range where transitions help the design to auto-sync to the received pattern.

Not implemented on NASA1 – reserved for external FIFO designs.

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

ExtFifoLoad: The external FIFO's have a LOAD pin which is tied directly to this register bit. The FIFO's in use are IDT72V36110. This is a dual purpose pin. During Master Reset, the state of the LD input determines the default offset values for the PAE and PAF flags selected. After Master Reset, this pin enables writing to and reading from the offset registers.

With ExtFifoLoad low during reset the default offset is 127. With ExtFifoLd high during reset the default offset is 1023. The offset is from the end of the FIFO. For example the PAE flag would be set to be 127, with the level of the FIFO 127 and less the PAE flag would be asserted. With 128K – 127 and more the PAF flag will be asserted.

In addition if the flags are to be reprogrammed using the parallel load feature the load pin should be low during reset Setting low allows writes to the FIFO to load new values into the FIFO. Taking Load high again puts the FIFO back into Data mode.

The FIFO's are configured with 32 in and 32 out. To parallel load the offset registers the PAE value is written first and the PAF value written second. Both values must be written to properly load. The values can be read back with the load pin low.

The data for the flags will flow through the TX DMA FIFO to the external FIFO. It is important to clear the FIFO's via reset prior to programming the offset registers to reset the pointer for the registers and to make sure the data written through the DMA FIFO is what is loaded into the External FIFO. Please also see ExtFifoMux control. The mux needs to be set to TX.

ExtFifoMux: bit controls if data is moved from the TX DMA FIFO or from the RX side into the External FIFO. '0' selects the TX path and '1' selects the RX path.

Direction: is used to select TX or RX operation for the IO. The direction bit is independent of the state-machines. Setting to 1 selects transmit operation and clearing to 0 selects RX operation for the transceivers assigned to the channel plus enables the terminations in the case of RX.

NASA1_CHAN_STATUS

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

Channel Status Register

Data Bit	Description
31	Interrupt Status
30	Local Int
29	spare
28	Direction – unused NASA1
27	Ext FIFO Full – unused NASA1
26	Ext FIFO AFull – unused NASA1
25	Ext FIFO AMT – unused NASA1
24	Ext FIFO MT – unused NASA1
23	Burst In Idle [write]
22	Burst Out Idle [read]
21	TX Idle State – channel dependent
20	RX Idle State – channel dependent
19	TX UnderFlow Err Lat / ParErr
18	RX OverFlow Err Lat
17	RX Complete / FrameErr / ManErr
16	TX Complete
15	Read DMA Interrupt Occurred
14	Write DMA Interrupt Occurred
13	Read DMA Error Occurred
12	Write DMA Error Occurred
11	RX AFull Int Lat
10	TX AMT Int Lat
9	RX AFull Int Lvl
8	TX AMT Int Lvl
7	spare
6	RX DMA FIFO Full
5	RX DMA FIFO Almost Full
4	RX DMA FIFO Empty
3	Spare
2	Tx DMA FIFO Full
1	Tx DMA FIFO Almost Empty
0	Tx DMA FIFO Empty

Figure 13 PMC-BiSerial-III Channel STATUS PORT

NASA1 FIFO: Two FIFO’s are used to create the internal Transmit and Receive DMA memory for each channel. The FIFO’s are tied to the PCI bus to enable burst operations for DMA. The status for the Transmit FIFO and Receive FIFO refer to these FIFO’s. The status is active high. 0x13 would correspond to empty Transmitter and empty Receiver.

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.

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.

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.

TX AMT Int Lvl: is set when the combined FIFO contents count is below the Almost Empty level total programmed into the reference register.

RX AFull Int Lvl: is set when the combined FIFO contents count is above the Almost Full Level Total programmed into the reference register. This value uses the word count including the FIFO and Pipeline.

TX AMT Int Lat: is set when the TX DMA FIFO contents count has been above the almost empty level and has transitioned below that level. This bit is a sticky bit and is held until cleared by writing back to this address with this bit set.

RX AFull Int Lat: is set when the RX DMA FIFO contents count has been below the almost full level and has transitioned above that level. This bit is a sticky bit and is held until cleared by writing back to this address with this bit set. This value uses the word count including both the FIFO and pipeline.

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.

TX Complete: This bit is set at the completion of the defined transmission. This is a sticky bit and can be cleared by writing back with this bit position set. The interrupt can be used as a timer based on the image size or to help with flow control when using larger DMA transfers with multiple transmissions. Tx complete is set when the programmed data length is reached. This bit has meaning when the packet modes are used.

RX Complete: This bit is set at the completion of the reception. This is a sticky bit and can be cleared by writing back with this bit position set. The interrupt can be used as a timer based on the image size or to help with flow control when using larger DMA transfers with multiple receptions. For LLST the end of frame is determined by DVAL being deasserted. For NMS the Frame signal is used to determine the end of the reception. RX complete is has alternate definitions for the UART and Manchester channels.

ManErr: This bit is set when the Manchester decoder has synchronized with a data stream and then a non Manchester valid character is found. In static mode the error bit will be set at the end of the reception since the static condition following valid data is not a valid Manchester condition, and there is no defined trailer to mask. This bit will have meaning for the Free Wheel mode where the idle pattern fills the void between receptions and is made up of valid Manchester characters.

FrameErr: For the UART channel this bit is set when a Frame Error is detected.   A framing error is an error where an expected Marking state is not received based on the programmed variables and the detection of the start bit. This can also indicate an error in the programmed number of Stop Bits. This is a sticky bit and requires a write back with this bit set to clear. This bit should be cleared after the UART is set-up.

ParityErr: For the UART channel this bit is set when a Parity error is detected. This is a sticky bit and requires a write back with this bit set to clear. This bit is ignored when Parity is disabled. This bit should be cleared after the UART is set-up. Parity can be Even or Odd or Disabled. When enabled the expected parity programming is tested against the received parity bit and an error detected if they do not match. If you are receiving parity errors first check that the programmed definition matches the transmitted data.

RX Over Flow: This bit is set if the RX SM FIFO is full when it is time to write more data. This is a sticky bit and requires a write back with this bit set to clear. In systems where the RX has a hold off mechanism if this bit is set the transmitter is not stopping within the specified number of clocks. [LLST, NMS] For non handshake interfaces it means the CPU is not keeping up with the data received.

TX Under Flow: This bit is set if the TX SM FIFO is empty when it is time to read more data. This is a sticky bit and requires a write back with this bit set to clear. This bit should not be set during operation. If it is check the ready size which will hold off the transmitter until sufficient data is stored. Alternatively for larger transfers there may be a system level issue preventing DMA from operating at the proper bandwidth. Please see additional comments in the DMA section.

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 data transfer; 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 transfer; 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 Transmit and burst data out of the card for Receive.

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.

Spare bits for NASA1 – No external FIFO implemented

EXT FIFO MT: is set when the external FIFO is empty, when 0 at least 1 data is stored into the external FIFO.

EXT FIFO AMT: is set when the external FIFO is almost empty, when 0 more than the Almost Empty threshold is stored into the external FIFO. The threshold is programmable. Please refer to the channel control definition for more programming information.

EXT FIFO AFull: is set when the external FIFO is Almost Full, when 0 less than the Almost Full threshold is stored into the external FIFO. The threshold is programmable. Please refer to the channel control definition for more programming information.

EXT FIFO Full: is set when the external FIFO is full, when 0 at least 1 data position is available for new data to be stored into the external FIFO.

NASA1_CHAN_WR_DMA_PNTR
[0x08] 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.

NASA1_CHAN_TX_FIFO_COUNT

[0x08] TX [Target] FIFO data count (read only)

TX FIFO Data Count Port

Data Bit	Description
31-16	Spare
15-0	TX Data Words Stored

Figure 15 PMC-BiSerial-III Transmit DMA FIFO data count Port

This read-only register port reports the number of 32-bit data words in the Transmit DMA FIFO. Unused bits within the count word are set to ‘0’. The spare bits are not driven and should be masked.

NASA1_CHAN_RD_DMA_PNTR

[0x0C] 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 16 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.

NASA1_CHAN_RX_FIFO_COUNT

[0x0C] RX [Master] FIFO data count (read only)

RX FIFO Data Count Port

Data Bit	Description
31-16	Spare
15-0	RX Data Words Stored

Figure 17 PMC-BiSerial-III Receive DMA FIFO data count Port

This read-only register port reports the number of 32-bit data words in the Receive FIFO plus pipeline. The maximum count is the FIFO size plus 4. The upper unused data bits within the word are set to ‘0’ by HW. The spare bits are not driven and should be masked.

NASA1_CHAN_FIFO

[0x10] Write TX/Read RX FIFO Port

RX [Master] and TX [Target] FIFO Port

Data Bit	Description
31-0	FIFO data word

Figure 18 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 Transmit FIFO, Reading unloads the Receive FIFO.

NASA1_CHAN_TX_AMT_LVL

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

Target Almost-Full Level Register

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

Figure 19 PMC-BiSerial-III Transmit ALMOST EMPTY LEVEL register

This read/write port accesses the almost-empty level register for the DMA FIFO. When the number of data words in the transmit DMA 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.

NASA1_CHAN_RX_AFL_LVL

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

Master Almost-Full Level Register

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

Figure 20 PMC-BiSerial-III Receive ALMOST FULL LEVEL register

This read/write port accesses the almost-full level register. When the number of data words in the receive DMA FIFO + pipeline is 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. The level includes the pipeline for an additional 4 locations.

NASA1_CHAN_TX_CNTRL

[0x1C] Channel Transmitter Control Register (read/write)

Channel Control Register

Data Bit	Description
23-16	channel clock rate Uplink
11	TxClkSel - UART
10	TxDataPhase - UpLink
10	TxStop2 - UART
9	TxStaticSelect – UpLink
9	TxParityOdd - UART
8	TxSFwMode - UpLink
8	TxParityEn - UART
7	TxByteMode – UART, UpLink
6	TxAeLvlIntEn
5	TxByteOrder
4	TxUnFlEn
3	TxAeIntEn
2	TxIntEn
1	TxStartClr – UART, Manchester
0	TxEn

Figure 21 PMC-BISERIAL-III Channel Transmitter Control Register

Each channel has a TX control register at the same offset. Where common features exist the same bit is used to control them. The bits with channel definitions are specific to those channels. The driver has separate structures for the UART and UpLink to make this easier to deal with.

TxEn when set causes the Transmit State Machine to begin operation. The transmitter will wait until there is sufficient data in the FIFO pipeline to begin operation. This is a programmable number and can be adjusted based on transfer size and scheme of operation. For continuous transmission it is recommended to set this value to several transfers to allow the pipeline to be almost full before starting.

TxStartClr when set enables the local Tx State-machine to auto-clear the Start bit to prevent starting the next transfer until software explicitly enables the Tx function with TxEn. When ‘0’ TxEn is unaffected by the state-machine auto-clear.

TxIntEn when set enables an interrupt to be generated for each image transmitted. The status and interrupt are set each time an image is completed.

TxAeIntEn when set enables an interrupt to be generated when the FIFO level has been more than Almost Empty and becomes Almost Empty. The status is a sticky bit and retains the information until cleared. The Status and interrupt is reasserted each time the transition occurs.

TxUnFlEn when set enables the interrupt from an underflow condition. The underflow status bit is set when the transmitter is ready to read the next pixel pair and the FIFO is empty. The status is a sticky bit and held until cleared. The underflow status is masked with TxUnFlEn to create the interrupt. Clear by clearing this bit or clearing the status [preferred].

TxByteOrder when cleared uses the PCI standard for word ordering and when set reverses the word order. When cleared the data on D7-0 are used first and the data on D31-24 are used last. When dealing with LSB first [UART] this bit is usually cleared. When dealing with MSB first this bit is usually set.

TxAeLvlIntEn when set enables the interrupt based on the Almost Empty condition. When the interrupt occurs a programmable amount of data is left in the FIFO. The interrupt can be used to request a new DMA for a new image to be transferred or for other purposes. Larger DMA transfers spanning multiple images are recommended for continuous operation.

TxByteMode – UART, UpLink When ‘1’ the Byte mode is used instead of Packet mode. The first byte of each word [only] is sent allowing the FIFO to be used like a register. Mainly intended for the UART channel. When the ByteOrder is ‘0’ only D7-0 are sent from each LW written. The FIFO still acts like a buffer, but the length programming is not required and the data is transferred as soon as it becomes valid in the FIFO.

When cleared to ‘0’ packet mode is used. The data length to send is programmed, and the data is sent with the minimum delay. This mode is normally used for longer data transfers.

TxClkSel – UART when set ‘1’ the Tx reference clock is substituted onto the TX lines to allow for ocilloscope checking of the programmed frequency. The PLL and dividers are checked. Normally not set for standard operation.

TxDataPhase – UpLink Set ‘1’ to invert the output data stream. Bi-Phase-L encoding is considered standard where the first 1/2 bit period is 1 and the second half bit period is 0 for a One and vice-versa for a zero. The encoding can be reversed by switching this bit.

TxStop2 – UART When set ‘1’ the UART is programmed to use at least 2 stop bits between characters. When cleared ‘0’ the UART is programmed to use at least 1 stop bit between characters. The state-machine will run with minimized gaps when the Packet mode is utilized. When in Byte mode the delay between characters will likely be much longer and will still be at least as long as programmed.

TxStaticSelect – UpLink When programmed to ‘1’ the StaticSelect bit causes the Transmitter to hold the last bit transmitted if in the Static Mode. When cleared ‘0’ the static mode value is ‘0’ with the standard DataPhase setting.

TxParityOdd – UART When set ‘1’ and Parity is enabled the Parity added will be odd. When cleared and Parity is enabled the parity added will be even. Odd parity is defined as being set when the number of data bits set is even to make the total count odd. Similarly Even parity is set to make the total of databits plus the parity bit an even count.

TxSFwMode – UpLink When set ‘1’ the StaticMode is selected. In StaticMode the output is held at 0 or the last bit sent [0 to begin with if nothing has been sent] until a new Manchester output is transmitted. The lines are again held once the transmission is complete. When cleared ‘0’ FreeWheel mode is used where the TX path is always active. When data is not being sent, “5555” is sent. 5555 decodes to a 00110011 pattern and looks like a 1/2 rate clock.

FreeWheel mode has the advantage of allowing the receiver to autosync to the pattern to be ready for the data when it comes, and to avoid any charge build-up on the lines from being held in a static state. The StaticMode has the advantage of not needing a filtering mechanism to keep the Idle pattern from filling the receiver memory.

Program this feature to match your system requirements.

Uplink Clock Rate: is used to control the transmit rate. The 12 bit field allows the channel software to divide the PLL reference to a user specified frequency. Set PLLD to rates which can be divided to the desired frequencies.   PLLD is used for both the Transmitter and Receiver. Both are programmed separately. Use the local divisor to select the operational rate. Use the formula /2N. The divider uses the programmed rate to terminate and restart counting. N = the 1/2 period rate. This results in a squarewave output for the state-machines to work with.

The Manchester channel (3) uses the divisor to generate the TX 2x clock. Set the 2X clock based on the TX desired rate X2. 2X is used to allow for both 1/2 periods per data bit period to be generated. The factor is 2N for the divisor. With an 8 MHz reference [PLLD], a divisor of 16 will produce a TX rate of 125 KHz.

The Manchester Uplink / Downlink rates can be programmed as follows for the standard frequencies.

TX	Divisor
125 K	4
100 K	5
50 K	10
25 K	20
10 K	50
5 K	100
2500	200

NASA1_CHAN_TX_WORD_COUNT

[0x20] TX “Packet” size

TX LW per Packet Definition

Data Bit	Description
31-30	Number of Bytes to send in last LW
29-0	Number of LW to transmit per Packet

Figure 22 PMC-BiSerial-III Transmit LW Count Port

This read-write register port holds the number of long words to transmit per Packet. The upper bits represent the number of bytes to send from the last LW. 00 = 1 byte, 01 = 2 bytes, 10 = 3 bytes, 11 = 4 bytes from the last LW. 0x80000010 would send 16 LW of data with 3 bytes used out of the 16th LW. The ByteOrder bit will determine from which end of each LW the data is sent including the last.

NASA1_CHAN_TX_FIFO_WORD_COUNT

[0x24] TX State Machine FIFO data count (read only)

TX State Machine FIFO Level

Data Bit	Description
31-16	Spare
15-0	TX Data Count in SM FIFO

Figure 23 PMC-BiSerial-III TARGET DATA COUNT Port

Unused for NASA1

The 1Kx32 FIFO between the External FIFO and the State-Machine is used for rate matching and local data storage for the TX state machine. The amount data stored in the FIFO can be read from this port. The clock reference to the FIFO is not the PCI clock. Reading this port while the level is changing can result in incorrect values. The values are correct when static. For example if the system is pre-filled before the TX side is enabled.

NASA1_CHAN_TX_AMT_LVL_TOTAL

[0x28] TX Almost Empty Level Control Register

TX Interrupt Level

Data Bit	Description
31-0	Number of words in FIFO for AMT level check

Figure 24 PMC-BiSerial-III TX AMT Level Control Port

This register defines a comparison against the TX FIFO path. The TX counts are added together and compared with the value programmed into this register. When the value of the count is less than the value of the register the AMT level interrupt can be asserted. The control register has the enable for the interrupt. The status register has the AMT Level Interrupt status.

Since NASA1 currently does not utilize the external FIFO's this register is somewhat overlapping with the AMT level programmed for the FIFO [5].

NASA1_CHAN_TX_READY_COUNT

[0x2C] TX Ready Count Control Register

TX Ready Count

Data Bit	Description
31-0	Number of words in FIFO for TX start

Figure 25 PMC-BiSerial-III TX Ready Count Control Port

This register defines a comparison against the TX FIFO path. The TX DMA FIFO Plus the External FIFO plus the TX SM FIFO counts are added together and compared with the value programmed into this register. When the value of the count is less than the value of the register the start of transmission is held off. The purpose is to allow the DMA to pre-fill the FIFO chain prior to starting transmission and to allow the software to perform one large DMA transfer instead of doing a prefill, start TX and then an additional DMA. The level should be set to match the size of the data to send up to the total FIFO size. For 1 transfer set to a single transfer -1 size, for two set to two etc.

The hold off can be used in conjunction with the AMT level interrupt to refill whenever “almost empty” and wait to transfer until at least 1 packet is loaded.   Set the Almost Empty to match the packet size in this case.

Alternatively multiple packets can be sent with 1 DMA using the hardware to parse out the separate packets as programmed. For large transfers this is the preferred method.

NASA1_CHAN_RX_CNTRL

[0x34] Channel Receiver Control Register (read/write)

Channel Control Register

Data Bit	Description
31	RxReSync – DownLink
30-24	Spare
23-16	Downlink Clock Rate
15-12	RxSyncSel - DownLink
11	Spare
10	RxDataPhase - DownLink
10	RxStop2 - UART
9	RxParityOdd - UART
8	RxSFwMode - DownLink
8	RxParityEn - UART
7	RxByteMode – UART, DownLink
6	RxAflLvlIntEn
5	RxByteOrder
4	RxOvFlEn
3	RxAfIntEn
2	RxIntEn
1	Spare
0	RxEn

Figure 26 PMC-BISERIAL-III Channel Receiver Control Register

Each channel has an RX control register at the same offset. Where common features exist the same bit is used to control them. The bits with channel definitions are specific to those channels. The driver has separate structures for the UART and DownLink to make this easier to deal with.

RxEn when set causes the Receiver State Machine to begin operation. The Receive state machine will begin checking for data when enabled. The data will begin capture based on the protocol and be loaded with the received clock until stopped by software.

For LLST the received clock is used to run the receiver interface. If the clock is not running the status and general operation of the “front end” of the LLST receiver will be suspect.

For NMS the received clock is sampled. If the clock is not present or stops the hardware will wait for the clock to continue. Pausing the clock can in effect pause the transfer without changing the data transferred.

At the end of the programmed packet size the hardware will look for the next start sequence. For LLST and NMS this is a HW "handshake". For the DownLink it can be the start of data or a synchronization pattern buried within the data. For the UART it is the Start bit being asserted once the Stop from the previous character has been detected.

RxIntEn when set causes an interrupt for each packet captured. This interrupt can be used to read each image sequentially as they are received. Alternatively the programmable level interrupt can be used based on data within the FIFO. For continuous operation larger multi-image DMA transfers are recommended. The LLST and NMS interfaces have defined HW handshaking allowing for the packet complete status.

The UART and DownLink do not have corresponding end of packet capabilities. The corresponding status bit is re-used for error status with these two channels.

RxAfIntEn when set enables an interrupt to be generated when the FIFO level has been less than Almost Full and becomes Almost Full. The status is a sticky bit and retains the information until cleared. The Status and interrupt is reasserted each time the transition occurs.

RxOvFlEn when set causes an interrupt when the Rx FIFO overflows. The FIFO is considered to be in an OverFLow condition if it is full when it is time to write the next data.

RxByteOrder when cleared causes the state machine to load the lower [D7-0] data with the first received data and upper [D31-24] with the 4th data received… When set the loading is reversed.

RxAflLvlIntEn when set enables the level based Almost Full Interrupt. This interrupt is asserted independent of transitions whenever the RX FIFO is almost full. Reading the data will reduce the level and remove the interrupt. The level can also be masked by clearing this control bit. Whenever the interrupt is asserted a DMA transfer of Almost Full size can be requested.

RxByteMode – UART, DownLink When '1' the Byte mode is used instead of Packet mode. The first byte of each word [only] is loaded allowing the FIFO to be used like a register. Mainly intended for the UART channel. When the ByteOrder is '0' only D7-0 are sent from each LW written. The FIFO still acts like a buffer and the data is available as soon as each character is received.

When cleared to ‘0’ packet mode is used. Data is packed into LW and loaded to the FIFO. The holding register is cleared after each LW is loaded to pre-fill to 0x00 before loading the next sequence of data. When SW has received “enough” and clears the start bit the remaining stored data is loaded to the FIFO with any unloaded bytes set to 0x00.

RxDataPhase – DownLink Set ‘1’ to invert the output data stream. Bi-Phase-L encoding is considered standard where the first 1/2 bit period is 1 and the second half bit period is 0 for a One and vice-versa for a zero. The expected encoding can be reversed by switching this bit.

RxStop2 – UART When set ‘1’ the UART is programmed to expect at least 2 stop bits between characters. When cleared ‘0’ the UART is programmed to expect at least 1 stop bit between characters. The Framing check will utilize this setting to check for proper operation.

RxParityOdd – UART When set ‘1’ and Parity is enabled the Parity will be tested for odd. When cleared and Parity is enabled the parity will be tested for even. Odd parity is defined as being set when the number of data bits set is even to make the total count odd. Similarly Even parity is set to make the total of data bits plus the parity bit an even count.

RxSFwMode – DownLink When set ‘1’ the StaticMode is selected. In StaticMode the expected input is held at 0 or the last bit sent [0 to begin with if nothing has been sent] until a new Manchester output is transmitted. The lines are again held once the transmission is complete. When cleared ‘0’ FreeWheel mode is expected where the TX path is always active. When data is not being sent, “5555” is sent.

Based on this selection the receiver HW selects between using the FreeWheel pattern to auto-synchronize to the received data pattern or to wait on the first bit received for synchronization. If the first bit is used – static mode – the bit expected is pulled from the synchronization pattern register using a mux to select the MSB based on the bytes enabled for use. The Receiver samples the data lines and determines if a ‘1’ or ‘0’ is the static value and then uses the knowledge of the first bit encoding to properly sync up with the initial transitions of the data.

Program this feature to match your system requirements.

RxSyncSel – DownLink – the 4 bits in this field correspond to the 4 bytes in the sync pattern register. D31-24 D23-16 D5-8 D7-0 correspond to the bits. When set to “0000” and in Static mode D31 is used for determining the expected first bit – no sync pattern just the sync bit.

The sync pattern is matched against the received data. With an 8 bit sync pattern D7-0 would be the first bits tested and normally be the ones used for matching with “0001”. Please note that the decoder uses all 32 bits for all comparisons, and when the corresponding byte is set to “0” for the sync is treated as a don’t care. This means that the second byte could be used instead with a don’t care for the first byte or the first and last etc. This may be useful for using a partial sync pattern in broadcast snooping situations.

Be sure to align the MSB of the expected sync pattern with the position the MSB will be in when received – for 8 bits only the bottom byte, for 32 bits all 4.

Please note that the sync pattern will allow the hardware to skip over idle patterns until the sync pattern is found. Since the HW has no way of knowing when the data transitions back to an Idle pattern the data will continue to be captured. The expected length can be programmed into the Almost Full definition. The receiver can be stopped or set to resync to stop further data capture until the next sync pattern comes. The FIFO can be emptied and the useful data retained. The FIFO word count can be used for this purpose.

Downlink Clock Rate: is used for the Manchester Downlink channel only. The 12 bit field allows the channel software to divide the PLL reference to a user specified frequency. Set PLLD to a rate which can be divided to the desired frequencies for both Rx and Tx Manchester functions. Use the formula /2N. The divider uses the programmed rate to terminate and restart counting. N = the 1/2 period rate. This results in a square wave output for the state-machines to work with.

The Manchester channel (3) uses the divisor to generate the RX 8x clock. With an 8 MHz reference [PLLD], a divisor of 4 will produce a 1 MHz Rx reference rate used to accept a bit rate of 125 Khz.

The Manchester Downlink rates can be programmed as follows for the standard frequencies.

Bit Rate	RX Rate	Divisor
125 K	1 M	4
100 K	800 K	5
50 K	400 K	10
25 K	200 K	20
10 K	80 K	50
5 K	40K	100
2500	20 K	200

Please note that the receiver for the Manchester function is designed to be adaptable to a range of frequencies. With the Manchester interface the reference should be within .75-1.25 of the programmed frequency [frequency ± jitter].

RxReSync – DownLink When ‘1’ the DownLink channel will reacquire the sync pattern as described by the Sync Select bits and the Sync Pattern register. The bit is auto cleared when Sync is acquired. It is not required to set this bit to acquire the sync pattern the initial time. If it is set the clearing mechanism can provide status that the sync pattern has been detected.

NASA1_CHAN_RX_WORD_COUNT
NASA1_CHAN_RX_SYNC_PATTERN

[0x38] RX LW Count of received Packets

RX Data Count Port

Data Bit	Description
31-0	RX LW expected per packet received
31-0	RX Sync Pattern - DownLink

Figure 27 PMC-BiSerial-III RX Data Sync Port

Unused NASA1 – register in place for LLST, NMS, UART for R/W only

This read-write register port holds the number of LW expected to be received per packet. Please note the total is Bytes42 to account for packing into LW for storage and DMA transfer. When started the RX hardware will wait for synchronization and then load data based on the received clock until the LW Count is received. The Packet Complete status is set and the HW begins looking for the sync pattern again.

For the **DownLink** channel this port is used to store the Synchronization pattern used. The LW can hold 4 bytes which correspond to the sync enable field in the Rx Control register. When set to ‘1’ the corresponding byte is enabled to be tested against the incoming data. The Sync pattern is primarily for filtering out the Idle pattern in a FreeWheel system or to filter out messages in a static system.

When no bytes are selected D31 is used to select the expected first bit in a static system without a sync pattern. When bytes are selected the MSB of the MS byte selected is used for the first bit detection in static mode.

In FreeWheel mode the Manchester decoder auto-synchronizes to the received pattern. The 00110011 continuous pattern has a combination of valid and illegal characters depending on how the bit periods are detected. The Manchester decoder starts with the first received data and if an illegal character is then detected will require to the other phase and be in sync. “00” and “11” are illegal, but “01” and “10” are legal.

NASA1_CHAN_RX_FIFO_WORD_COUNT

[0x3C] RX State Machine FIFO data count (read only)

RX State Machine FIFO Level

Data Bit	Description
31-16	Spare
15-0	TX Data Count in SM FIFO

Figure 28 PMC-BiSerial-III RX SM DATA COUNT Port

UNUSED in NASA1

The 1Kx32 FIFO between the External FIFO and the State-Machine is used for rate matching and local data storage for the RX state machine. The amount data stored in the FIFO can be read from this port. The clock reference to the FIFO is not the PCI clock. Reading this port while the level is changing can result in incorrect values. The values are correct when static. For example if the system is disabled and the remaining data is to be read out, the total data register can be read or the individual FIFO counts can be read and used to determine the final transfer size.

NASA1_CHAN_RX_AFL_LVL_TOTAL

[0x40] RX Almost Full Level Control Register

RX Interrupt Level

Data Bit	Description
31-0	Number of words in FIFO for AFL level check

Figure 29 PMC-BiSerial-III RX AFL Level Control Port

This register defines a comparison against the RX FIFO path. The RX DMA FIFO Plus the External FIFO plus the RX SM FIFO counts are added together and compared with the value programmed into this register. When the value of the count is greater than the value of the register the AFL level interrupt can be asserted. The control register has the enable for the interrupt. The status register has the AFL Level Interrupt status.

For NASA1 the external FIFO and SM FIFO are not required. The count reverts to the DMA FIFO plus pipeline.

NASA1_CHAN_EXT_FIFO_WORD_COUNT

[0x44] External FIFO Level Read Only register

External FIFO Level

Data Bit	Description
31-0	Number of words in FIFO

Figure 30 PMC-BiSerial-III RX External FIFO Level Port

Unused NASA1

This read only register has the count for the amount of data stored into the external FIFO. The reference clock for the external FIFO is not the same as the PCI clock. The count is re-clocked onto the PCI clock for this port. The values can jump due to the re-clocking. When static the value will be true.

NASA1_CHAN_TX_WORD_COUNT

[0x48] TX total FIFO Count

Total FIFO Data TX

Data Bit	Description
31-0	Combined TX Path FIFO Count

Figure 31 PMC-BiSerial-III TX Total FIFO Count

Unused NASA1

This read only register provides the total data count held in the TX FIFO path. Please note that the external FIFO overlaps with the RX path. Reading this port in RX mode will return the overlapping portion of the data and vice-versa.

NASA1_CHAN_RX_WORD_COUNT

[0x4C] RX Total FIFO Count

Total FIFO Data RX

Data Bit	Description
31-0	Combined RX Path FIFO Count

Figure 32 PMC-BiSerial-III RX Total FIFO Count

Unused NASA1

This read only register provides the total data count held in the RX FIFO path. Please note that the external FIFO overlaps with the TX path. Reading this port in TX mode will return the overlapping portion of the data and vice-versa.

Loop-back

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

The test set-up included PCIBPMCX1, NASA1, 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
LlstFrmRdyTx+	pin 1	pin 5	LlstFrmRdyRx+
LlstFrmRdyTx-	pin 35	pin 39	LlstFrmRdyRx-
LlstClkTx+	pin 2	pin 6	LlstClkRx+
LlstClkTx-	pin 36	pin 40	LlstClkRx-
LlstDValnTx+	pin 3	pin 7	LlstDValnRx+
LlstDValnTx-	pin 37	pin 41	LlstDValnRx-
LlstDataTx+	pin 4	pin 8	LlstDataRx+
LlstDataTx-	pin 38	pin 42	LlstDataRx-
NmsFrmnTx+	pin 9	pin 13	NmsFrmnRx+
NmsFrmnTx-	pin 43	pin 47	NmsFrmnRx-
NmsClkTx+	pin 10	pin 14	NmsClkRx+
NmsClkTx-	pin 44	pin 48	NmsClkRx-
NmsDataTx+	pin 11	pin 15	NmsDataRx+
NmsDataTx-	pin 45	pin 49	NmsDataRx-
NmsRdyTx+	pin 12	pin 16	NmsRdyRx+
NmsRdyTx-	pin 46	pin 50	NmsRdyRx-
UartTx+	pin 17	pin 18	UartRx+
UartTx-	pin 51	pin 52	UartRx-
UpLink+	pin 19	pin 20	DownLink+
UpLink-	pin 53	pin 54	DownLink-
Par0+	pin 21	pin 28	Par7+
Par0-	pin 55	pin 62	Par7-
Par1+	pin 22	pin 29	Par8+
Par1-	pin 56	pin 63	Par8-
Par2+	pin 23	pin 30	Par9+
Par2-	pin 57	pin 64	Par9-
Par3+	pin 24	pin 31	Par10+
Par3-	pin 58	pin 65	Par10-
Par4+	pin 25	pin 32	Par11+
Par4-	pin 59	pin 66	Par11-
Par5+	pin 26	pin 33	Par12+
Par5-	pin 60	pin 67	Par12-
Par6+	pin 27	pin 34	Par13+
Par6-	pin 61	pin 68	Par13-

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 33 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 34 PMC-BISERIAL-III Pn2 Interface

PMC Module Rear IO Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module IO Interface on the PMC-BiSerial-III. Installed for –RP models. Also see the User Manual for your carrier board for more information. Standard NASA1 is –FP [SCSI II]

IO_0p (Ch0Pixel0+)	IO_0m (Ch0Pixel0-)	1	35
IO_1p (Ch0Pixel1+)	IO_1m (Ch0Pixel1-)	2	36
IO_2p (Ch0Pixel2+)	IO_2m (Ch0Pixel2-)	3	37
IO_3p (Ch0Pixel3+)	IO_3m (Ch0Pixel3-)	4	38
IO_4p (Ch0Pixel4+)	IO_4m (Ch0Pixel4-)	5	39
IO_5p (Ch0Pixel5+)	IO_5m (Ch0Pixel5-)	6	40
IO_6p (Ch0Pixel6+)	IO_6m (Ch0Pixel6-)	7	41
IO_7p (Ch0Pixel7+)	IO_7m (Ch0Pixel7-)	8	42
IO_8p (Ch0Pixel8+)	IO_8m (Ch0Pixel8-)	9	43
IO_9p (Ch0Pixel9+)	IO_9m (Ch0Pixel9-)	10	44
IO_10p (Ch0Pixel10+)	IO_10m (Ch0Pixel10-)	11	45
IO_11p (Ch0Pixel11+)	IO_11m (Ch0Pixel11-)	12	46
IO_12p (Ch0BadBit+)	IO_12m (Ch0BadBit-)	13	47
IO_13p (Ch0HREF+)	IO_13m (Ch0HREF-)	14	48
IO_14p (Ch0VREF+)	IO_14m (Ch0VREF-)	15	49
IO_15p (Ch0CLK+)	IO_15m (Ch0CLK-)	16	50
IO_16p (Ch1Pixel0+)	IO_16m (Ch1Pixel0-)	17	51
IO_17p (Ch1Pixel1+)	IO_17m (Ch1Pixel1-)	18	52
IO_18p (Ch1Pixel2+)	IO_18m (Ch1Pixel2-)	19	53
IO_19p (Ch1Pixel3+)	IO_19m (Ch1Pixel3-)	20	54
IO_20p (Par0+)	IO_20m (Par0-)	21	55
IO_21p (Par1+)	IO_21m (Par1-)	22	56
IO_22p (Par2+)	IO_22m (Par2-)	23	57
IO_23p (Par3+)	IO_23m (Par3-)	24	58
IO_24p (Par4+)	IO_24m (Par4-)	25	59
IO_25p (Par5+)	IO_25m (Par5-)	26	60
IO_26p (Par6+)	IO_26m (Par6-)	27	61
IO_27p (Par7+)	IO_27m (Par7-)	28	62
IO_28p (Par8+)	IO_28m (Par8-)	29	63
IO_29p (Par9+)	IO_29m (Par9-)	30	64
IO_30p (Par10+)	IO_30m (Par10-)	31	65
IO_31p (Par11+)	IO_31m (Par11-)	32	66
IO_32p (Par12+)	IO_32m (Par12-)	33	67
IO_33p (Par13+)	IO_33m (Par13-)	34	68

Figure 35 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:	LVDS IO for LLST [LADEE], 485 IO for NMS. UART, UpLink, DownLink, and Parallel Port.
CLK rates supported:	PLLA is set to the LLST transmit rate, PLLB is set to the NMS transmit rate, PLLC is set to the UART reference rate, PLLD is set to the Manchester reference rate. The UART and Manchester channels use local dividers to provide multiple IO rates without reprogramming the PLL.
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 120 nS [4 PCI clocks] or burst mode DMA – 1 word per PCI clock transferred.
Interrupt:	Each port has independently programmable interrupt sources, DMA interrupts included.
Onboard Options:	All Options are Software Programmable
Interface Options:	SCSI III connector at front bezel standard. Rear IO via Pn4 standard 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 Industrial
PMC-BiSerial-III-NASA1 PMC Module with 4 channels plus Parallel Port. LLST, NMS, UART, and Manchester encoding/decoding. Full Duplex operation. Independent rates.
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. NASA1 selection determines [-FP]

Pick any combination to go with IO
-CC to add conformal coating
-ET to change to industrial Temp [-40 - +85C]
-COM to change to commercial temp parts [0-70] NASA1 defines this option as standard
-TS to add thumbscrew option – standard is latch block at SCSI connector

Related:
PCIBPMCX1: PCI to PMC adapter to allow installation of PMC-BiSerial-III into a PCI system with differential, matched length, impedance controlled Pn4 IO.
http://www.dyneng.com/pcibpmcx1.html

PCIeBPMCX1: PCIe to PMC adapter to install the 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. Differentially routed break-out for SCSI cabled systems.
http://www.dyneng.com/HDEterm68.html

HDEcabl68: SCSI II/III cable compatible with NASA1 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_biserial_III.html