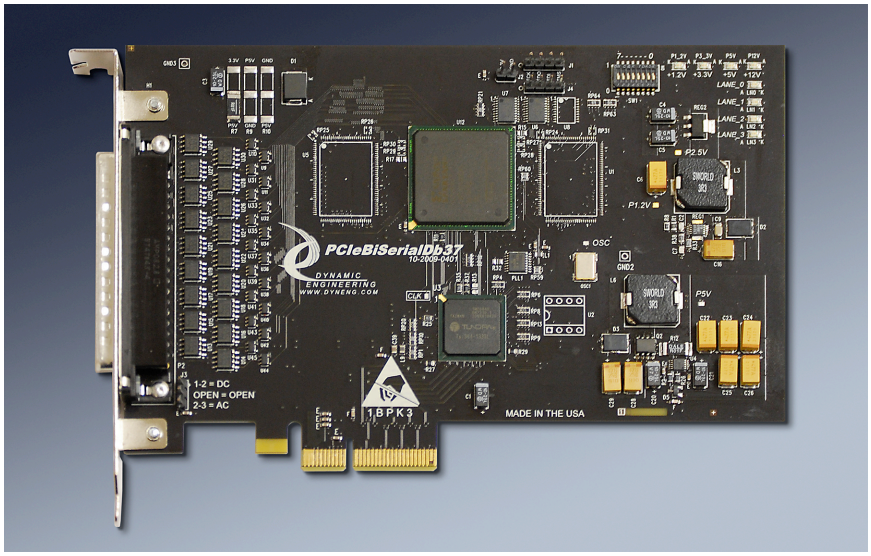
DYNAMIC ENGINEERING
150 DuBois St. Suite C, Santa Cruz, CA 95060
831-457-8891 Fax 831-457-4793
http://www.dyneng.com
sales@dyneng.com
Est. 1988

User Manual

PCIeBiSerialDb37-L3COM1

Half Duplex Byte Wide Data Port
Reference clock, Flow Control, Strobe
PCIe 4 lane Module
LVDS



Revision A6
Corresponding Hardware: Revision 3
10-2009-0403
FLASH 0106

PCIeBiSerialDb37L3Com1

Digital Parallel Interface
PCIe Module
Dynamic Engineering
150 DuBois St. Suite C, Santa Cruz CA 95060
831-457-8891 831-457-4793 FAX

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 PCIe motherboards and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage.

Revised 6/3/17

Table of Contents

List of Figures

Product Description

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

The BiSerial family currently includes IP, PMC, PCI-104, and PCIe versions, each with multiple "clientized" design implementations.

Usually the choice of format is based on other system constraints. Dynamic Engineering is happy to assist in your decision regarding architecture and other trade-offs with the form factor decision. Dynamic Engineering has carriers for for most systems, and is adding more as new solutions are requested by our clients.

PCIe compatible PCIeBiSerialDb37 has 18 independent differential IO available. The DB-37 connector is mounted through the bezel to carry the signals. Each of the IO has independent direction and termination controls. Each of the IO is matched length and routed with 100 Ω differential impedance.

The IO's 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 can operate with 3.3V or 5V [different devices]. Mixed IO types are implemented with the 3.3V versions of the devices.

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

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

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

PcieBiSerialDb37L3Com1 is a "clientized" version of the standard PCIeBiSerialDb37 board. "L3COM1" is set to use the LVDS standard, and supports a single half duplex channel. The transmitter and receiver are designed to interface with a byte wide data stream using a free running clock plus data enable and ready/busy flow control.

The receive side auto-bauds to the incoming rate by using the received clock and enable to load data bytes into a FIFO. Data is removed from the FIFO using an internal reference clock.

The Transmitter is supported with a combination of 4Kx32 FIFO and 8Kx32 FIFO for a total of 12Kx32. Using DMA transfers the transmit side can provide a continuous flow of data at the output or bursted depending on the programmed byte wide data speed and the system DMA capabilities.

The Receiver has a 4K x8 first stage FIFO followed by two 128Kx32 FIFO's and a 1Kx32 FIFO. The 1Kx32 FIFO serves as the DMA source FIFO, the two 128K x32 FIFO's the bulk storage for the receiver.

With a 40 MHz input stream, the receiver FIFO chain can store 1,056,760 bytes of data for ~26 mS of storage. With streaming DMA to a storage file, the system can adsorb the data for very large transfers with no loss of data. With 26 mS of storage the system can "be out to lunch" for a while without loss of data. The DMA engine can push unlimited size DMA transfers – the PCIe side will stop a particular transfer and the HW will automatically request another until the entire programmed transfer is complete.

The Receiver and Transmitter are separately supported with scatter gather capable DMA engines.

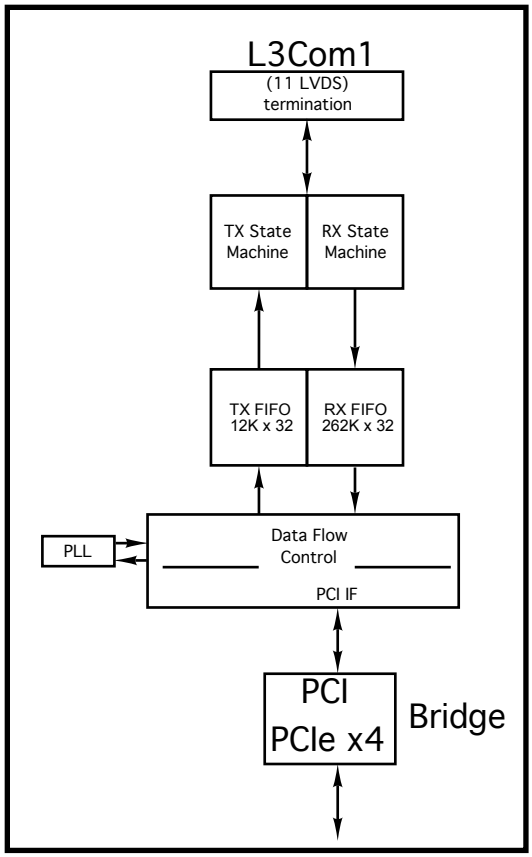


Figure 1 PcieBiSerialDb37L3Com1 Block Diagram Primary Function

The L3COM1 supports transmission and reception of byte wide data. The Data transfer is controlled with a free running clock. The receiver uses the received clock to load data into a small FIFO. A high speed reference clock is used to read the data and assemble into 32 bit words. The data is then loaded into the storage pipeline using the PCI clock.

The data is received on the rising edge of the clock. The Data Enable is used to select which bytes are loaded. When the receiver is Almost full the hand shake control can be used to hold off the transmitter. In addition the hold off is asserted when the receiver is not enabled.

The transmitter provides approximately 50% duty cycle and changes the data, and Data Enable on the falling edge of the clock. The transmitter can use the hold off to pause transmission using two modes. One mode stops at the completion of the packet and one at the current data. If the gap between packets is used to determine packets the system should use the stop at the end feature. If alternate means are used to determine boundaries the stop during transmission feature can be used as a more traditional flow control.

The PLL is used to create the Tx clock reference. The control is via SW. The PLL is referenced to 50 MHz. and can be programmed with new .JED files using the driver. The IO reference is programed to 2x the desired output rate. For example 76 MHz will produce 38 MHz at the IO.

In packet mode, the hardware waits until there is a definition of the packet byte count, and data in the transmit FIFO matches or exceeds the start-up value before transmission begins. Once started if the data FIFO is empty when the transmitter is ready to read the next data set and the packet is not complete, an error for underflow is flagged. The error can cause an interrupt if desired.

Both the transmitter and receiver allow for byte reversal. The data is stored as 32 bit words into the transmit FIFO from the system or the receive FIFO from the interface. The data is used with little Endian conventions as the default – 0,1,2,3 for the byte order where 0 = D7-0 [data on AD7-0] first and D31-24 last. The bits are sent LSB first so D0 is first on the line and D31 is last if all 4 bytes are to be sent. Similarly the receiver loads 0,1,2,3 so the first bit in goes into D0 and the last into D31 for each long word. When the bytes are reversed the order becomes 3,2,1,0. For systems using Windows Little Endian is consistent with the driver and memory mapping. With Linux and some RISC based systems the reversal may be necessary.

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.

In the “L3COM1” design the Termination and Direction controls are set in the VHDL for the IO. The received signals are terminated and the transmitted signals are not. The terminations automatically switch based on how the IO is configured.

All of the IO is routed through the FPGA to allow for custom applications. Larger external and internal FIFO’s and Dual Ported memories are implemented for this version by FPGA selection and adding the 128K x 32 FIFO’s to the board.

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

PcieBiserialDb37 can be used for multiple purposes with applications in telecommunications, control, sensors, IO, test; anywhere multiple independent or coordinated IO are useful.

PcieBiserialDb37 features a Xilinx FPGA, and high speed differential devices. The FPGA contains the PCI interface and control required for the IO interface.

The Xilinx design incorporates the “PCI Core” and additional modules for DMA in parallel with a direct register decoded programming model. The design model has a “base” level with the basic board level functions and “channels” which contain IO oriented functions. In the L3COM1 design the IO functions are designed into the channel and the PLL programming, switch, and other common or basic functions are in the base design.

From a software perspective the design can be treated as “Flat” or as a hierarchy. The Dynamic Engineering Windows® driver uses the hierarchical approach to allow for more consistent software with common bit maps and offsets. This implementation has only one channel. The channel function was kept to allow for future expansion with more than 1 IO interface or a secondary function in added channels. The user software can control the Channels with the same calls and use the channel number to distinguish. This makes for consistent and easier to implement user level software.

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

The transceivers are initialized to the receive state. Once a channel is defined via software to be a transmitter the IO are enabled and driven to the appropriate levels. Terminations are activated for ports defined to be receivers. The transmit clock is generated as soon as the transmit function is selected. The clock is free running and may be required for proper receiver operation.

PcieBiserialDb37 is part of the PCIe Module family of modular I/O components. The PcieBiserialDb37 conforms to the PCIe standard. This guarantees compatibility with your PCIe system. The base is 4 lane operation. The design can handle 1-4 lanes being available. LED’s are provided to show the active PCIe lanes.

Designs implemented on PC104p, PMC, IP and PCIe versions of the BiSerial family can in large part be ported between platforms. If you see what you need in one version and prefer it on another please contact Dynamic Engineering about porting the design. In most cases it will require a recompile of the VHDL and not much more. We do a lot of “just like but different “ adaptations for our clients. Please contact us to help you with a successful special adaptation of off- the-shelf hardware.

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

programmed with two memory spaces with two interrupts per loop.

The hardware is organized with the IO function in channel 0 and the card level functions in the “base”. The driver provides the ability to find the hardware and to allocate resources to use the base and channel functions.

The basic use of the interface is to facilitate data transfer between the host and the remote target.

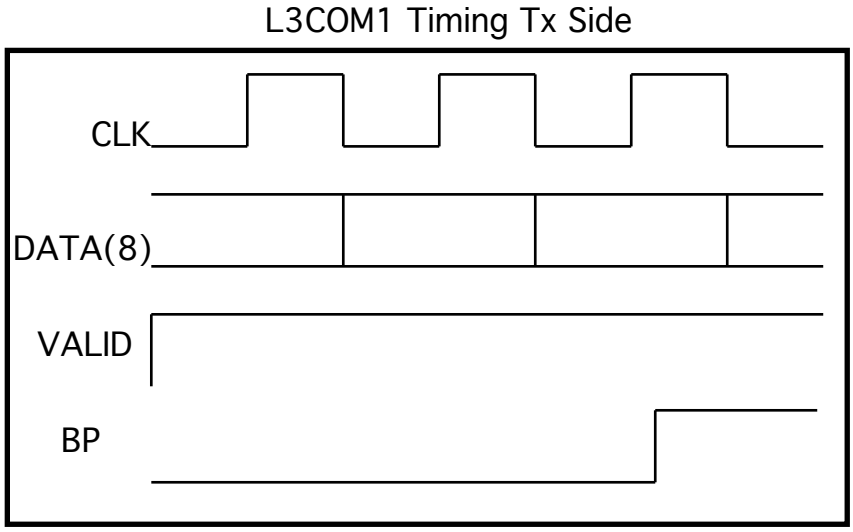


Figure 2 PCIEBISERIALDB37L3COM1 Timing Diagram

The clock is free running with data valid on the rising edge. The transmitter provides data with close to 50% duty cycle – changing on the falling edge and stable on the rising edge. The Valid signal is used as a data enable by the receiver to capture the data on a clock by clock basis. Typically enable is set for the length of a frame, but the receiver does not require this.

Address Map

Base Address Map

Function	Offset
// **PCIeBiSerialDb37L3COM1 BASE definitions**	
#define L3COM1_BASE_BASE	0x0000 // 0 L3COM1Base Base control register
#define L3COM1_BASE_PLL_WRITE	0x0000 // 0 L3COM1Base Base control register
#define L3COM1_BASE_PLL_READ	0x0000 // 0 L3COM1Base base control register
#define L3COM1_BASE_USER_SWITCH	0x0004 // 1 L3COM1Base User DIP switch read
#define L3COM1_BASE_XILINX_REV	0x0004 // 1 L3COM1Base Xilinx revision read port
#define L3COM1_BASE_XILINX_DES	0x0004 // 1 L3COM1Base Xilinx design read port
#define L3COM1_BASE_STATUS	0x0008 // 2 L3COM1Base status Register offset

Figure 3 PCIeBiSerialDb37L3COM1 Internal Address Map Base Functions

The address map provided is for the local decoding performed within PcieBiserialDb37L3COM1. The addresses are all offsets from a base address. Dynamic Engineering prefers a long-word oriented approach because it is more consistent across platforms.

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

The host system will search the PCI bus to find the assets installed during power-on initialization. The VendorId = 0xDCBA and the CardId = 0x0059 for the PcieBiSerialDb37L3COM1.

The L3COM1 design has 1 channel implemented at this time. 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)	COM name
Base	0-19 (0x0000)	PLL, Switch, Status
Channel 0	20-39 (0x0050)	L3COM1 Transmitter & Receiver

Channel Address Map

Function		Register	Register definitions
#define	CH_CNTL	0	// Main control register for channel
#define	CH_STATUS	1	// Main Status register for channel
#define	CH_DMA_IN	2	// Burst In function Write physical address of linked list here [read into board to support transmit]
#define	CH_TX_CNT	2	// Read Tx FIFO count
#define	CH_DMA_OUT	3	// Burst Out function Write physical address of linked list here [write out of board to support Rx]
#define	CH_RX_CNT	3	// Read Rx FIFO count with complete chain represented
#define	CH_FIFO	4	// Single Word Read from Rx or Write to Tx FIFO
#define	CH_TX_AE	5	// Transmit Almost Empty definition - When CH_TX_AE is >= to FIFO Count, the Tx FIFO is Almost Empty
#define	CH_RX_AF	6	// Receive Almost Full definition - When Rx Total count > CH_RX_AF, the Rx FIFO is almost Full
#define	CH_TX_CNTL	7	// Transmit Control Register
#define	CH_PCK_LEN_TX_FIFO	8	// 1Kx32 FIFO - write to store Tx Packet Length descriptor
#define	CH_PCK_LEN_TX_REG	9	// Register for Fixed packet size operation
#define	CH_STATUS_II	10	// Extended Status register
#define	CH_TX_START_CNT	12	// 16 bit start count - Tx FIFO will need this level before transmission begins
#define	CH_RX_CNTL	13	// Receive Control Register
#define	CH_PCK_LEN_RX	14	// 1Kx32 FIFO - Read (only) stored Rx Packet Length descriptor

Figure 4 PcieBiSerialDb37L3COM1 Channel Address Map

Register number is a Long Word. Multiply by 4 and add to the Channel Offset for a byte address pointer.

Status => Reg 1 => x54 [x50 + x4]

Programming

Programming the PcieBiSerialDb37L3COM1 requires only the ability to read and write data in the host's PCIe space.

Once the initialization process has occurred, and the system has assigned addresses to PcieBiSerialDb37L3COM1 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 PcieBiSerialDb37L3Com1. The PLL will need to be programmed to use the transmit function. The Cypress CyberClocks software can be used to create new .JED files if desired. PLLA should be set to the transmit reference frequency x2.

The driver comes with a .JED file prepared. 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 for the L3COM1 direction and termination are hardwired in this design. The ports are unidirectional and initialization is simplified with this approach.

The control bits for the L3COM1 will select how the data is transmitted – Byte ordering, size of transfer etc.

For Windows™ and Linux systems the Dynamic Drivers 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 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 L3COM1 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 or at least the Packet loaded if shorter than the FIFO size. For reception it means that the FIFO is under HW control and the delay from starting reception to starting DMA won't cause an

overflow condition.

DMA can be programmed with a specific length. The length can be as long as you want within standard memory limitations. At the end of the DMA transfer the Host will receive an interrupt. 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 referred to as “ping-pong”.

The Receive Byte Count register can be used to control the input Packet size to work with your DMA scheme. Alternately the Byte Count register can be programmed to the size of the Packet if known. For situations where the size is unknown the timeout and Almost Full interrupt options can be used.

Packets can be used to control the Transmission of data. Packet interrupts can be enabled to support this mode of operation. For short lengths the packet length interrupts may overlap since the HW is transmitting in parallel with the SW processing the interrupts. The Tx Almost Empty interrupt may be useful in this situation. The Tx Packet Interrupt is set at end of each packet sent. The status is always available and the interrupt can be enabled should it be desired for use. This allows both polled and interrupt modes to be used. The Packet FIFO for transmission can hold up to 1K packet descriptors. The packet descriptor is the byte count for that packet.

The transmitter state machine when in Packet Mode stays in the Idle state until enabled, a packet descriptor is available, and the FIFO Start amount of data is stored. When a packet is completed the state-machine rechecks the FIFO Start signal and packet descriptor status before starting the next packet.

FIFO Start should be programmed to be large enough to ensure sufficient data to the transmitter to prevent underflow conditions. For small packet sizes the size of the packet is recommended. For larger packets think in terms of the time the start up data represents compared to your system capabilities.

The transmitter can also be used in non-packet mode when data is on LW boundaries. The data sent will be continuous until the Data FIFO is empty. Make sure enough data is preloaded and that the Tx Almost Empty level is programmed with enough remaining buffer to allow for continuous operation should that be desired.

For received data packet descriptors are generated and stored in the Rx Packet Descriptor FIFO. The descriptor is the number of Bytes of data per packet received. Packets are padded to be on LW boundaries. If extra Padding is selected the LW bounded packet has an additional 3 LW [12 bytes] added. Please note: the Padding is not in the descriptor. Your software will need to be aware of the mode you are in and round the packet length to the next LW for DMA purposes and possibly add 12 bytes on top of that to retrieve the complete packet.

Please see the channel control register bit maps for more information.

Base Register Definitions

BASE_BASE

[$00 Base 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
15-1	spare
0	BigEndianDma

Figure 5 PcieBiSerialDb37L3COM1 Control Base Register Bit Map

This is the base control register for the L3COM1. 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.

BigEndianDma : '0' disables this option. '1' enables this option. When operating with a Big Endian platform and using PCI accesses DMA can have challenges. The register accesses directly over the PCI bus are usually taken care of automatically with byte swapping within the CPU or PCI interface on the CPU. DMA data is written to or read from the local memory and is not swapped. The direct read/write from memory ends up with scrambled data [relative to little endian definitions]. Setting this bit will byte reverse the data for the DMA path into the Tx and out of the Rx FIFO's only. Register accesses are not affected.

31-24, 23-16, 15-8, 7-0 ⇔ 7-0, 15-8, 23-16, 31-24 byte swapping pattern implemented.

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 software, and to use the translation and programming files ported to your environment. This procedure will save you a lot of time. For those who want to do it themselves the Cypress PLL in use is the 22393. The output file from the Cypress tool can be passed directly to the Dynamic Driver [Linux or Windows] and used to program the PLL without user intervention.

The reference frequency for the PLL is 50 MHz.

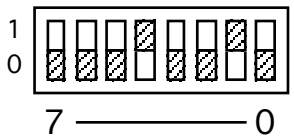
BASE_ID

[$04 Switch and Design number port read only]

DATA BIT	DESCRIPTION
31-24	Design Number
23-16	Revision Minor
15-8	Revision Major
7-0	DIP switch

Figure 6 PcieBiSerialDb37L3COM1 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 PcieBiserialDb37L3COM1 physical card matches each PCIe 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 Number and Revision Fields are 8 bits allowing for 256 designs and 256 revisions of each. The L3COM1 design is 0x03 the current Major revision is 0x01. The Minor revision is used for small HW updates within a design cycle. Major client releases cause a Major Revision update.

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

BASE_STATUS

[$08 Board level Status Port read only]

DATA BIT	DESCRIPTION
31-10	set to ‘0
9	undefined
8	undefined
7-1	set to ‘0’, reserved for additional channels
0	Unmasked Ch0 Interrupt

Figure 7 PcieBiSerialDb37L3COM1 Status Port Bit Map

Channel Interrupt – The local 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.

Channel Bit Maps

The L3COM1 design has 1 channel. The basic control signals are the same for the channel base, channel status, FIFO and DMA interfaces across multiple designs.

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

CHAN_CNTL

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

Channel Control Register

Data Bit	Description
31-20	spare
19	BackPressSwBit
18	BackPressCntl
17	Force Busy
16	Transmit
15-10	spare
9	External FIFO Reset
8	OutUrgent
7	InUrgent
6	Read DMA Interrupt Enable
5	Write DMA Interrupt Enable
4	Force Interrupt
3	Channel Interrupt Enable
2	Bypass
1	Receive FIFO Reset
0	Transmit FIFO Reset

Figure 8 PcieBiSerialDb37L3COM1 channel Control Register

FIFO Transmitter/Receiver Reset: When set to a one, the transmit and/or receive FIFO's will be reset. This includes the Packet FIFO's and Transmit / Receive State Machines. When these bits are zero, normal FIFO operation is enabled.

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 enable is 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.

The Tx side has 2 FIFO's in series and the first 4K x 32 is part of the Bypass loop. On the receive side all but the specialized IO processing memories are part of the loop including both 128Kx32 "external" FIFO's, Rx DMA FIFO, and a 1Kx32 internal FIFO placed between the two external FIFO's.

FIFO External Reset: When set to '1', the External FIFOs will be reset. When cleared the External FIFO is enabled.

Transmit when '1' sets the channel to Transmit operation. The separate TX Enable bit on the transmit control register enables the Transmit state-machine. When in Transmit mode the free running clock is enabled to be transmitted, the LVDS lines for Data, Data En are enabled to be driven, and the BackPressure signal is set to input. The output signals are not terminated and the input signals are.

Transmit when cleared '0' sets the channel to receive mode. Data, DataEn, and clock become inputs, BackPressure is the only output. Terminations are enabled for the inputs and disabled on the output.

Transmit should be enabled prior to the receiver being enabled [on another card] to provide the clock in case it is required for proper reset operation. When configured to be a receiver the external transmitter is required to drive the clock signal prior to initializing the receiver.

Force Busy when '1' will cause the BackPressure signal to be asserted whether the conditions call for it or not. This bit should only be asserted when in Receive mode

[Transmit = 0]. This bit only applies to the Automatic BackPressure mode.

BackPressCntl when '1' enables the Automatic control of BackPressure – FIFO level controlled. When this control is cleared, use the SW control bit to Assert or disable BackPressure. Power on default is SW control.

BackPressSwBit directly controls the BackPressure signal when in receive mode and BackPressCntl is cleared to SW control. Power on default is BackPressure low.

Please note: with the Rx SM not enabled any data received due to BP not being asserted will be ignored. See Channel Rx Control register.

CHAN_STATUS

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

Channel Status Register

Data Bit	Description
31	Interrupt Status
30	LocalInt
29	RxOvFlLat
28	TxUnFlLat
27	TxFfIIFL
26	TxFfIIAfl
25	TxFfAmt
24	TxFfIIMt
23	BurstInIdle
22	BurstOutIdle
21	ExtFifo1Fl
20	ExtFifo0Fl
19	ExtFifo1Hf
18	ExtFifo0Hf
17	ExtFifo1Mt
16	ExtFifo0Mt
15	Read DMA Interrupt Occurred
14	Write DMA Interrupt Occurred
13	Read DMA Error Occurred
12	Write DMA Error Occurred
11	RxAFLvlIntLat
10	TxAELvlIntLat
9	RxPckLat
8	TxPckLat
7	Rx Idle
6	Rx FIFO Full
5	Rx FIFO Almost Full – complete chain
4	Rx FIFO Empty
3	Tx Idle
2	Tx FIFO Full
1	Tx FIFO Almost Empty – complete chain
0	Tx FIFO Empty

Figure 9 PcieBiSerialDb37L3COM1 Channel STATUS PORT

FIFO: 4K + 8K x 32 FIFO's are used to create the internal Tx memory. The Rx side uses a combination of internal block RAM FIFO and two 128Kx32 FIFO's. The status for the Tx FIFO and Rx FIFO refer to these FIFO's. The status is active high. The Full and Empty status come from the "DMA" FIFO's while the Almost Full and Almost Empty

status reflects the state of the total FIFO. 0x13 would correspond to empty Rx and empty Tx DMA FIFO's. The DMA FIFO's are the pair of internal FIFO's which interact with the DMA engine. First in pipeline for TX and last in pipeline for RX.

Please note with the Rx 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 DMA 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. Based on the total count within the FIFO chain.

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. Rx DMA 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 DMA 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. Total Tx FIFO chain.

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

Tx/Rx IDLE when set indicates the corresponding state-machine is in the Idle state. Useful to know when frequency changes etc. can be made.

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.

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.

RxAFLvlIntLat: When set the Rx Data FIFO has become almost Full based on the programmed count. The software can do a looped read or use DMA to retrieve the programmed count amount of data from the storage FIFO. The signal is latched and can be cleared via write back with this bit set. The signal can be used to generate an interrupt if desired. Interrupt enable located in RX Control register.

TxAELvlIntLat: When set the Tx Data FIFO has become almost Empty based on the programmed count. The software can do a looped write or use DMA to load the programmed count amount of data to the storage FIFO. The signal is latched and can be cleared via write back with this bit set. The signal can be used to generate an interrupt if desired. Interrupt enable located in TX Control register.

TxPckLat: When set the Tx State-Machine has completed a packet. The signal is latched and can be cleared via write back with this bit set. The signal can be used to generate an interrupt if desired. Interrupt enable located in TX Control register.

RxPckLat: When set the Rx State-Machine has completed receiving a packet. The signal is latched and can be cleared via write back with this bit set. The signal can be used to generate an interrupt if desired. Interrupt enable located in RX Control register.

Tx FIFO II is the 8K x 32 FIFO tied to the IO port. A transfer engine moves data from the 4K x 32 Tx DMA FIFO to FIFO II. The Afl and Amt levels are fixed for this FIFO. The Afl and Amt levels are used to control the transfer engine and are provided mainly for data flow tracking purposes and usually are not needed for standard operation.

TxFfIIMt 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.

TxFfIIAmt When a one is read, the FIFO contains less than x10 LW of data; when a zero is read, there is at least x10 data words are in the FIFO.

TxFfIIAfl When a one is read, the FIFO contains the Full FIFO size - x10 LW of data; when a zero is read, there are more than x10 open data words in the FIFO.

TxFfIIFl When a one is read, the FIFO is full; when a zero is read, there is at least 1 open data location in the FIFO.

TxUnFlLat When set the Data FIFO did not have enough data to support the programmed packet – set when it is time to read the next word and the FIFO is empty. Latched bit cleared by write back with the same bit set. Can be used as an interrupt – enable bit in Tx control register.

RxOvFlLat: When set the Data FIFO was full when time to write more data into the Rx FIFO chain. Latched bit cleared by write back with the same bit set. Can be used as an interrupt – enable bit in Rx control register.

The External FIFO status are provided for data tracking. Usually the total count and packet size is used to determine the amount of data available for

External FIFO 0 is the 128Kx32 FIFO closest to the receiver logic. External FIFO 1 is a second 128Kx32 FIFO in series located closest to the PCI bus logic.

ExtFifo0Mt, ExtFifo1Mt when set the corresponding FIFO is empty. When '0' at least one data word is stored in the FIFO.

ExtFifo0Hf, ExtFifo1Hf when set indicate at least ½ the size of the FIFO is populated with data. When not set more than ½ of the FIFO size is available.

ExtFifo0Fl, ExtFifo1Fl when set indicate the FIFO's are full. When not set at least 1 location is available.

CHAN_WR_DMA_PNTR
[0x8] Write DMA Pointer (write only)

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

Figure 10 PcieBiSerialDb37L3COM1 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.

CHAN_TX_FIFO_COUNT

[0x8] 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 11 PcieBiSerialDb37L3COM1 TX FIFO data count Port

This read-only register port reports the number of 32-bit data words in the Transmit FIFO. The FIFO count is padded with '0' to a word boundary.

CHAN_RD_DMA_PNTR

[0xC] Read DMA Pointer (write only)

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

Figure 12 PcieBiSerialDb37L3COM1 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.

CHAN_RX_FIFO_COUNT

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

RX FIFO Data Count Port	
Data Bit	Description
31-0	RX Data Words Stored

Figure 13 PcieBiSerialDb37L3COM1 RX 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 FIFO count is padded with '0' .

CHAN_FIFO

[0x10] Write TX/Read RX FIFO Port

RX and TX FIFO Port	
Data Bit	Description
31-0	FIFO data word

Figure 14 PcieBiSerialDb37L3COM1 RX/TX FIFO Port

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

CHAN_TX_AMT_LVL

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

Tx Almost-Full Level Register	
Data Bit	Description
31-16	Spare
15-0	Tx FIFO Almost-Empty Level

Figure 15 PcieBiSerialDb37L3COM1 TX ALMOST EMPTY LEVEL register

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

CHAN_RX_AFL_LVL

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

Rx Almost-Full Level Register	
Data Bit	Description
31-0	Rx FIFO Almost-Full Level

Figure 16 PcieBiSerialDb37L3COM1 RX ALMOST FULL LEVEL register

This read/write port accesses the almost-full level register. When the number of data words in the receive data FIFO is equal or greater than this value, the almost-full status bit will be set. The mask is valid for a size matching the depth of the Rx FIFO chain. The level includes the pipeline for an additional 4 locations.

CHAN_TX

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

Channel TX Control Register

Data Bit	Description
11	BPFC
10	BPHO
9	PKT
8	RegPkt
7-6	spare
5	spare
4	TxUnFlIntEn
3	TxAEIntEn
2	TxPktIntEn
1	spare
0	TxEn

Figure 17 PcieBiSerialDb37L3COM1 Channel Transmit Control Register

TxEn when set causes the Transmit State Machine to begin operation. When the Transmitter has determined the start-up criteria are met, data will be transmitted. Clearing TxEn will return the State Machine to the idle state. While in the Transmit Mode [see channel control] and not enabled the transmitter SM generates the output clock. This is accomplished by having two IDLE states. The IDLE status is when in either state.

Disabling TxEn while transmitting is considered a “panic stop” situation. The FIFO’s should be cleared and the process restarted from scratch when this is done. Restarting without clearing will allow the data to be completed but with the incorrect remaining packet count likely leading to error(s) being detected.

TxPktIntEn when set enables the interrupt based on the Packet Complete Interrupt. Packet Complete indicates the programmed amount of data has been transferred. The interrupt is cleared by writing back to the status bit for Tx Packet Int in the status register. The interrupt can be masked by disabling the master interrupt for the channel or clearing this bit.

TxAEIntEn when set enables the interrupt based on the TX FIFO Almost Empty flag. When the interrupt occurs a programmable amount of data can be stored into the FIFO making for an efficient DMA or burst of writes to load the FIFO.

Interrupt status available in the Status Register. Clear by writing a '1' to the corresponding status position. When transmitting data lengths larger than 48K bytes, allow enough FIFO data points to insure the FIFO does not go empty due to system time to reload the DMA.

At 40 MHz the byte rate is one per 25 nS or 4 per 100 nS. If the level is set to ½ the FIFO size, the buffer time is 6K*100nS => 614 uS which should be sufficient for most situations to restart the DMA. Since the DMA has flow control you can program longer DMA transfers and use the DMA interrupt to restart as well in which case the FIFO will be nearly full when the interrupt is received.

TxUnFlIntEn when set '1' allows an underflow condition to generate an interrupt request. Status can be polled either way. When in Packet Mode [does not apply to non packet mode] the Under Flow bit is set if the Data FIFO is empty when it is time to read the next LW and a packet is not complete. Clear by write back to the status register. Mask with this bit or the master interrupt enable.

RegPkt when '0' selects the Packet FIFO for use when in Packet Mode. When '1' the Packet Register is used. In either case the Packet Descriptor is read and used to reload the local counter to track the number of bytes to send. In the case of the register all packets are the same length, and the transmission is gated with the FIFO Start level since the register is assumed to always be loaded.

PKT when '1' selects Packet Mode. In Packet Mode the Packet Descriptor controls the number of bytes to send. Larger DMA transfers can be set-up covering multiple packets and the descriptors used to send each packet.

Transmitted data is LW aligned for the first location. The last LW is transmitted as controlled by the Packet Descriptor remainder and then discarded to start on the next LW for the next Packet. For example with a Packet Descriptor of 0x09, three LW are required. The first 8 bytes from the first two LW plus 1 byte from the 3rd LW leaving a remainder of 3 discarded bytes in this case.

BPHO – When set '1' BackPressure Hold Off programs the Tx State Machine to use the BackPressure signal as a Hold Off. Operationally this means the current packet will complete, and when in the IDLE state checking for the next packet descriptor, FIFO Start and BackPressure will wait for BackPressure to be removed if asserted. The receiver will need to take the packet length for the system into account when programming the BackPressure control. This control normally would be used independently of BPFC.

BPFC - when set '1' BackPressure Flow Control programs the State Machine to use BackPressure as a Flow Control signal. At the next LW boundary the transmit state machine will pause for BackPressure being asserted and then resume when

BackPressure is removed. The Data Enable will be disabled when paused. Gernerally this control is used with the receiver programmed to go busy [BackPressure asserted] with a smaller x10 from full condition and allows potentially larger transfers. The side affect is that a paused transfer may become multiple packets at the receiver. This control normally would be used independently of BPHO. When '0' the SM ignores this control.

CHAN_PKT_LEN_TX_FIFO

[0x20] Channel TX Packet FIFO (read/write)

Channel Tx Packet FIFO	
Data Bit	Description
31-0	Tx Packet Descriptor

Figure 18 PcieBiSerialDb37L3COM1 Channel Tx Packet FIFO

Write and read-back of Tx Packet Descriptors is allowed from this FIFO. Please note that read-back is for test and removes the descriptors from the queue.

The descriptor is a byte count. The data stored into the Tx FIFO is parsed and transmitted based on the stored descriptors if the Packet Mode is selected and Packet FIFO is selected. Non LW lengths are allowed. Data is assumed to be LW oriented and read from the LW 3 2 1 0 with 0 first and 3 last. When bytes are stranded by non-LW aligned counts, those bytes are ignored and the next packet starts on the next LW boundary.

CHAN_PKT_LEN_TX_REG

[0x24] Channel TX Packet Reg (read/write)

Channel Tx Packet Register	
Data Bit	Description
31-0	Tx Packet Descriptor

Figure 19 PcieBiSerialDb37L3COM1 Channel Tx Packet Reg

The Packet Register can be used when the system uses a consistent size packet to eliminate the writing of packet descriptors for each transmission.

The descriptor definition is the same from the register and FIFO please see above.

CHAN_STATUS_II

[0x28] Channel Status Read/Clear Latch Write Port

Channel Status Register

Data Bit	Description
31-16	RxPacketCount
6	BackPressureLat
5	BackPressure
4	FifoSt
3	RxPktFl
2	RxPktMt
1	TxPktFl
0	TxPktMt

Figure 20 PcieBiSerialDb37L3COM1 Channel STATUS PORT II

TxPktMt when set indicates the Tx Packet FIFO is empty. This does not not mean the Tx Packet is completed, only read. When not set at least 1 packet descriptor is available to the Tx State machine.

TxPktFl when set indicates the Tx Packet FIFO is full and additional Packet Descriptors should not be written at this time.

RxPktMt when set indicates no Rx Packet Descriptors are available. When clear at least 1 Packet Descriptor is available to be read.

RxPktFl when set indicates the Rx Packet FIFO is full and unless descriptors are read additional descriptors will not be stored resulting in a loss of descriptors should additional data be received. The Rx FIFO can hold 1K descriptors.

FifoSt when set indicates the Tx FIFO chain data stored >= programmed level. When not set the level is less than the programmed level. FifoSt is used to trigger the Transmitter to begin transmission.

BackPressure is a copy of the current state of the BackPressure signal. When acting as a receiver the signal can be used to verify that BackPressure is asserted and as a transmitter can be read to know that the transmitter is held off.

BackPressureLat is a latched version – if BackPressure is asserted this signal is set high and held until cleared via write back with the corresponding bit set.

RxPacketCount represents the number of Packet Descriptors available in the Rx Packet FIFO. When operating with small size packets at a high rate the count should be checked with the Packet Interrupt to see if more than one is available to process.

CHAN_TX_START_CNT

[0x30] Channel TX Start Count (read/write)

Tx Start Count	
Data Bit	Description
31-16	spare
15-0	Tx Start Count

Figure 21 PcieBiSerialDb37L3COM1 Channel Tx Start Count Register

Tx Start Count is the number of LW required to be in the Tx Data FIFO before transmission begins. Set to 0x01 to remove this control. Set to the length of the Packet in LW [0x09 becomes x3] for small packets. For larger packets set to a value larger enough to prevent underflow situations while starting up. Underflowing means the Tx FIFO is empty when it is time to read the next value and results in the Data Enable being disabled while more data is written. The packet may be received as multiple packets if this happens.

CHAN_RX

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

Channel Control Register

Data Bit	Description
23-16	PadPattern
15-5	spare
4	RxOvFlIntEn
3	RxFifoAFIntEn
2	RxPktIntEn
1	EPAD
0	RxEn

Figure 22 PcieBiSerialDb37L3COM1 Channel Rx Control Register

RxEn when set causes the Rx State Machine to begin operation. Data is captured on the rising edge of the received clock.

EPAD when set '1' causes an extra 12 bytes of PadPattern to be appended to the stored message. Standard padding is to the next LW boundary. If a packet ends with Byte 0,1,2 then Byte 3,2,1 will be padded as needed for a complete LW. With extra padding the LW padding is still done and an additional 3 LW of padding added.

RxPktIntEn when set enables the interrupt based on the Packet Complete Interrupt. Packet Complete indicates the DataEnable signal has transitioned to indicate the end of a packet. The interrupt is cleared by writing back to the status bit for Rx Packet Int in the status register. The interrupt can be masked by disabling the master interrupt for the channel or clearing this bit.

RxFifoAFIntEn when set enables the interrupt based on the Rx FIFO Almost Full flag. When the interrupt occurs a programmable amount of data can be read from the FIFO making for an efficient DMA read or burst of reads to unload the FIFO.

PadPattern is programmed with the pattern to use as a fill for packets not on a LW boundary. For EPAD situations the pattern is repeated 4 times per LW and written 3 times for 12 extra bytes of PadPattern.

CHAN_PKT_LEN_RX_FIFO

[0x38] Channel RX Packet FIFO (read)

Channel Rx Packet FIFO	
Data Bit	Description
31-0	Rx Packet Descriptor

Figure 23 PcieBiSerialDb37L3COM1 Channel Rx Packet FIFO

Received data is stored into the data FIFO. The Packet Descriptor is stored into this read only FIFO. The descriptor is a byte count. The amount of data in the Data FIFO is the packet size converted to LW and potentially increased by 3. For example if 11 bytes are received in a packet, 3 LW will be reported with the last LW being padded by 1 byte. In addition if the EPAD bit is set the total will be 6 since 3 additional LW of the PadPattern are stored.

SW can read the packet size to know the byte count of live data and then adjust as needed for the DMA byte count to include the padding. Please note that most DMA sizes are in terms of bytes but require LW compatible counts due to the size of the bus.

Loop-back

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

The test set-up included PcieBiSerialDb37L3COM1 and loop-back cable. The IO names can be used to accommodate a different set-up. The loop-back cable is a DB37 female connector [L77SDC37S] with the interconnections protected with a connector shell [977-037-010R031]. Part numbers shown for reference. Other similar parts are fine.

The ports are half duplex, the test software configures one side to transmit and one to receive. The signals are shown for one direction. The names are reversed for the reverse direction. The differential pairs are on twisted pair wiring. The shields are tied together and to pin 19. The Connector reference header can be used to cause AC, DC or open connections on each end.

Signal	From	To	Signal
TxData_0+	1	1	RxData_0+
TxData_0-	2	2	RxData_0-
TxData_1+	20	20	RxData_1+
TxData_1-	21	21	RxData_1-
TxData_2+	3	3	RxData_2+
TxData_2-	4	4	RxData_2-
TxData_3+	22	22	RxData_3+
TxData_3-	23	23	RxData_3-
TxData_4+	5	5	RxData_4+
TxData_4-	6	6	RxData_4-
TxData_5+	24	24	RxData_5+
TxData_5-	25	25	RxData_5-
TxData_6+	7	7	RxData_6+
TxData_6-	8	8	RxData_6-
TxData_7+	26	26	RxData_7+
TxData_7-	27	27	RxData_7-
TxTiming+	9	9	RxTiming+
TxTiming-	10	10	RxTiming-
TxValid+	28	28	RxValid+
TxValid-	29	29	RxValid-
TxBackPressure+	11	11	RxBackPressure+
TxBackPressure-	12	12	RxBackPressure-
GndRef	19	19	GndRef

Front Panel IO Interface Pin Assignment

The figure below gives the pin assignments for the IO Interface on the PcieBiSerialDb37L3Com1.

IO_0p (Data0+)	IO_0m (Data0-)	1	2
IO_1p (Data1+)	IO_1m (Data1-)	20	21
IO_2p (Data2+)	IO_2m (Data2-)	3	4
IO_3p (Data3+)	IO_3m (Data3-)	22	23
IO_4p (Data4+)	IO_4m (Data4-)	5	6
IO_5p (Data5+)	IO_5m (Data5-)	24	25
IO_6p (Data6+)	IO_6m (Data6-)	7	8
IO_7p (Data7+)	IO_7m (Data7-)	26	27
IO_8p (Timing+)	IO_8m (Timing -)	9	10
IO_9p (Valid+)	IO_9m (Valid-)	28	29
IO_10p (BackPressure+)	IO_10m (BackPressure-)	11	12
IO_11p (Spare+)	IO_11m (Spare -)	30	31
IO_12p (Spare +)	IO_12m (Spare -)	13	14
IO_13p (Spare +)	IO_13m (Spare -)	32	33
IO_14p (Spare +)	IO_14m (Spare -)	15	16
IO_15p (Spare +)	IO_15m (Spare -)	34	35
IO_16p (Spare +)	IO_16m (Spare -)	17	18
IO_17p (Spare +)	IO_17m (Spare -)	36	37
GND*		19	

Figure 24 PcieBiSerialDb37L3COM1 FRONT PANEL Interface

GND is shunt selectable for DC, AC and open configurations. Jumper is located near lower edge of the male DB37 connector. Options labeled in silk-screen.

IO is half duplex. When transmitting: Data, Valid, Timing are outputs and BackPressure is an input. When receiving Data, Valid, Timing are inputs and BackPressure an output.

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 DB37 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.

Coming Soon. Terminal Block. We offer a high quality 37 screw terminal block that directly connects to the DB37. The terminal block can mount on standard DIN rails. DBterm37 has an associated twisted pair cable compatible with the PCIeBiSerialDB37. [http://www.dyneng.com/DBterm37.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

PCIe Modules were conceived and engineered for rugged industrial environments. The PcieBiSerialDb37L3COM1 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 analog switches and termination resistors are located directly under the transceivers and connected with “zero stub” routing to eliminate unwanted effects from unused options.

Surface mounted components are used. Most components are Industrial temperature ranges. Conformal coating is an option for condensing environments or for another measure of board protection. Please order the “CC” version.

The PCIe is secured against the chassis with the connectors and front panel. If more security against vibration is required a chassis with top side support can be used. The PCIeBiSerialDb37 has a wider keep out than required by PCIe specification to allow use in industrial chassis and horizontal mount situations.

The power and ground planes are implemented with relatively heavy copper to help with heat spreading in chassis with limited air flow. The components are spaced to allow for efficient cooling and power dispersion.

Thermal Considerations

The PcieBiSerialDb37L3COM1 design consists of CMOS and similar 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; cooling with forced air 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. 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:	PCIe 1-4 lanes. 4 lanes recommended
Digital Parallel IO:	LVDS IO
Digital Serial IO:	Byte Wide data plus clock for Tx and Rx function. Data valid on rising edge of clock. 40 MHz for initial target design.
DIP Switch:	DipSwitch supplied for board identification and other user purposes.
CLK rates supported:	PLLA is programmed to select Transmit Clock rate. For loop-back and alternate HW implementations. PLLB, C, D reserved for new applications.
Software Interface:	Control Registers, IO registers, IO Read-Back registers, FIFO. R/W, 32 bit boundaries.
Initialization:	Programming procedure documented in this manual
Access Modes:	LW to registers, read-write to most registers
Access Time:	Frame to TRDY 121 nS [4 PCI clocks] or burst mode DMA – 1 word per PCI clock transferred.
Interrupt:	Each port has independently programmable interrupt sources, DMA interrupts included.
Onboard Options:	All Options are Software Programmable
Interface Options:	37 Pin male DB connector at front bezel.
Dimensions:	Standard 1/2 length PCIe module.
Construction:	Multi-Layer Printed Circuit, Through Hole and Surface Mount Components.
Power:	+12 and +3.3 used from PCIe interface. No secondary power supply connections required. 1.2, 2.5 and 5V developed locally.
Weight:	TBD oz

Order Information

standard temperature range Industrial
PcieBiSerialDb37L3COM1 PCIe Module with18 IO channels. Half Duplex. [Byte wide data, plus clock, strobe, flow control]
http://www.dyneng.com/pciebiserialdb37.html

Order Options:

Pick any combination to go with IO
-CC to add conformal coating
-ROHS add ROHS processing

Dbterm37: 37 position terminal block with DB37 connector.
http://www.dyneng.com/DBterm37.html

Dbcabl37: DB37 cable compatible with PCIeBiSerialDB37. Twisted pairs on correct pin pairs. **http://www.dyneng.com/DBcabl37.html**

PCIe BiSerial DB37 L3COM1 Eng Kit : Windows or Linux Driver software, Loop-Back cable, reference schematics. Recommended for first time purchases.
http://www.dyneng.com/pciebiserialdb37.html