DYNAMIC ENGINEERING
150 DuBois St., Suite B&C, Santa Cruz, CA. 95060
831-457-8891 sales@dyneng.com
www.dyneng.com
Est. 1988

User Manual

PCIe-ASCB
PCI-ASCB
Bus Versions D/E
Avionics Bus Interface

2-channel Bi-directional
Transformer coupled ASCB interface
PCIe/PCI Board and PMC Module Carrier

Revision 1.3
Current Fab: 10-2018-1104
Revised 12/15/23

PCIe-ASCB & PCI-ASCB
2-Channel Bi-Directional
PCIe / PCI Board and PMC Module
Carrier

Dynamic Engineering
150 DuBois St., Suite B&C
Santa Cruz, CA 95060
831-457-8891

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

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

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

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

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

Table of Contents

PRODUCT DESCRIPTION 6

THEORY OF OPERATION 8

ADDRESS MAP 11

PROGRAMMING 12

Revisions 12

REGISTER DEFINITIONS 13

BASE_CONTROL 13
BASE_STATUS 15
BASE_FRAME_END 16
BASE_FRAME_INT 16
BASE_FRAME_CNT 17
BASE_PKT_DELAY 17
BASE_LM75 18
CHAN_CONTROL 19
CHAN_STATUS 21
CHAN_TX_DMA 22
CHAN_RX_DMA 23
CHAN_TX_INDX 23
CHAN_RX_INDX 24
CHAN_X_PKT_Y 24
TX_SM_MEM 26
RX_SM_MEM 27

PCIE-ASCB IO PIN ASSIGNMENT 28

APPLICATIONS GUIDE 29

Interfacing 29

Construction and Reliability 30

Thermal Considerations 30

Warranty and Repair 31

Service Policy 31

Out of Warranty Repairs 31

SPECIFICATIONS 32

ORDER INFORMATION 33

List of Figures

FIGURE 1 PCIE-ASCB BLOCK DIAGRAM 7
FIGURE 2 PCIE-ASCB MANCHESTER TIMING DIAGRAM 8
FIGURE 3 PCIE-ASCB 8B10B DIAGRAM 9
FIGURE 4 PCIE-ASCB INTERNAL ADDRESS MAP 11
FIGURE 5 PCIE-ASCB BASE CONTROL REGISTER BIT MAP 13
FIGURE 6 PCIE-ASCB STATUS REGISTER 15
FIGURE 7 PCIE-ASCB FRAME END COUNT REGISTER 16
FIGURE 8 PCIE-ASCB FRAME INTERRUPT COUNT REGISTER 16
FIGURE 9 PCIE-ASCB CURRENT FRAME COUNT REGISTER 17
FIGURE 10 PCIE-ASCB PACKET DELAY REGISTER 17
FIGURE 11 PCIE-ASCB LM75 18
FIGURE 12 PCIE-ASCB CHANNEL CONTROL REGISTER 19
FIGURE 13 PCIE-ASCB CHANNEL STATUS REGISTER 21
FIGURE 14 PCIE-ASCB CHANNEL WRITE DMA POINTER PORT 22
FIGURE 15 PCIE-ASCB CHANNEL READ DMA POINTER PORT 23
FIGURE 16 PCIE-ASCB CHANNEL TRANSMIT INDEX REGISTER 23
FIGURE 17 PCIE-ASCB CHANNEL RECEIVE INDEX REGISTER 24
FIGURE 18 PCIE-ASCB CHANNEL RECEIVE INDEX REGISTER 24
FIGURE 19 ASCB CONNECTOR PINOUT 28

Product Description

PCIe-ASCB and PCI-ASCB are special-purpose PCIe/PCI board designed to interface with the Avionics Standard Communication Bus versions D the newer “enhanced” version “eASCB”. The interface can be used for a variety of applications including the testing avionics components. This manual covers both the PCI and PCIe versions of the design.

The ASCB function is the same on both devices. The architecture is also the same other than the bridge. PCIe has a PCIe-PCI bridge while the PCI version has a PCI to PCI bridge. For simplicity where PCIe-ASCB is written PCI-ASCB can be substituted.

PCIe-ASCB has an on-board PCIe/PCI bridge that creates a local two-position PCI bus. Position 0 is an FPGA implementing two ASCB-D/E interfaces complete with a frame-count timing standard. Two nine-pin D-connectors on the front panel provide the ASCB ports, each with primary and back-up transformer-coupled bus I/O with fail-safe signal disconnect shunts and individual 16 KByte frame data memory for the transmit and receive functions.

PCI Position 1 provides a PMC slot for an optional PrPMC (Processor PCI Mezzanine Card) module. The ASCB interrupt can be configured to be serviced by either the local processor or the PCI host computer.

Along with the primary and back-up differential data busses, each nine-pin connector has two active-low bus disables; one for each of the primary and back-up busses and a frame sync pulse output that indicates the start of a data-frame for bus test analysis purposes.

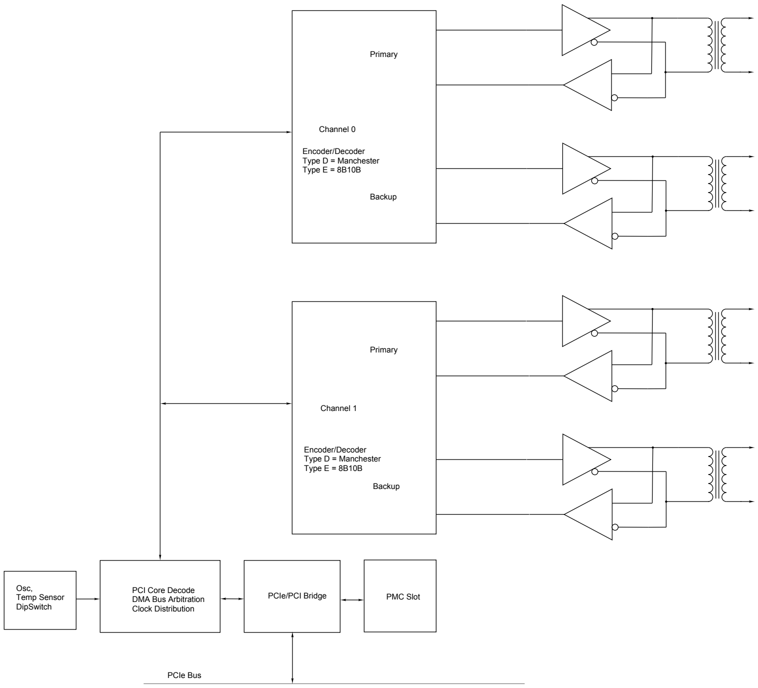


FIGURE 1 PCIE-ASCB BLOCK DIAGRAM

Theory of Operation

PCIeASCB is designed for transferring data from one point to another with a serial protocol. Currently two of the ASCB protocols are implemented. Type “D” and type “E”.

PCIeASCB features a Xilinx Spartan FPGA. There are two models with the same functionality – Spartan VI and Spartan III. With the two options enhanced availability is achieved. The FPGA contains all of the registers and protocol controlling elements of the design. Only the transformers, transceivers, and switches are external to the Xilinx device.

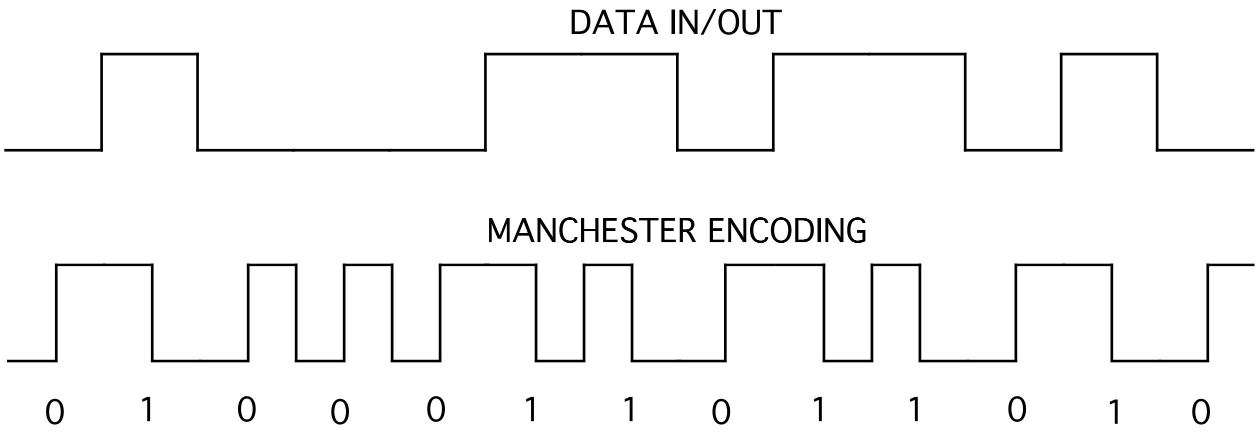


FIGURE 2 PCIE-ASCB MANCHESTER TIMING DIAGRAM

PCIe-ASCB can support many protocols. The D Type uses Manchester serial encoded data and clock. Data is sent in 16-bit words concatenated for multiple word transfers. The Manchester timing is shown in Figure 2. The E Type uses 8B10B encoding and operates at 20 MHz. See Figure 3 for the message format.

State machines within the FPGA control all transfers between the internal DPR and FPGA logic, and the FPGA and the data buffers. The TX state machine reads from the transmit memory and loads the shift-register before sending the data. The RX state machine receives data from the data buffers and takes care of moving data from the shift-register into the RX memory.

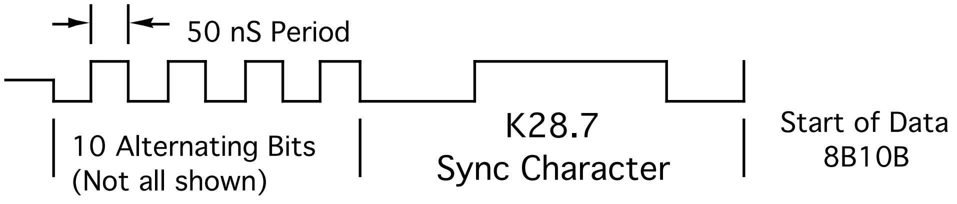


FIGURE 3 PCIE-ASCB 8B10B DIAGRAM

The E type interface starts with the transmitter coming out of tri-state and driving the bus with an alternating pattern of 1's and 0's for 20 bit periods. The preamble is followed by the K character (K28.7) which is encoded as shown. Data follows the sync character and continues until the next to last position. The data is assumed to be on 32 bit boundaries with the last 32 bits being the CRC for the message. Data is encoded and decoded based on the 8b10b standard. No control characters are contained in the message body. The CRC is calculated using the same method as the D type and applied to all of the data in the body of the message.

The hardware [when receiving] determines the lack of new data arriving and terminates the current message reception.

The K character has two encodings, and since the new message always starts with the same running disparity, the encoding for the initial position is always the same. The Hardware resets the disparity at the start of each message to enforce this behavior.

The control interface and memory operating procedure is the same for both versions. A single bit is switched to change from D to E and vice-versa. In addition this version allows for the primary and redundant data to be captured in parallel. The previous model required the choice of primary or redundant. The single channel larger memory allocation is also under software control.

Custom interfaces are available. We will redesign the state machines and create a version for your specific requirements.

ASCB implements 4 – Dual-Port RAMs (DPR) using the internal block RAM of the Xilinx. Each port has two associated DPRs. Each DPR is configured to have a 32-bit port on the PCI side, and a 32-bit port on the IO side.

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

Interrupts are supported by PCIe-ASCB. An interrupt can be configured to occur at the end of a transmitted packet or message. An interrupt can be set at the end of a received packet. All interrupts are individually maskable, and a master interrupt enable is also provided to disable all interrupts simultaneously.

PCIe-ASCB is interconnected to other equipment via cable. The cable can have multiple taps and is terminated at the end points. The terminations are 124 ohms each providing a 62 ohm parallel impedance. Ideally the cable will match this as well. PCIe-ASCB uses transformer coupling with a 1:1 transformer. The cable side has 30.1 ohm stub resistors to conform to other fielded standard equipment and approximate the impedance on the cable.

The FPGA is isolated from the transformer with specialized cable transmitter devices providing ~10V Pk-Pk. The receive function is done with RS485 transceivers strapped for receive only. The 485 devices are special extra high bandwidth to allow decoding of long cable modified signals.

PCIe-ASCB has been tested using flight cables of approximately 300 Ft.

Address Map

BAR 0:	Offset	Reg num	Function
BASE_CNTRL	0x0000	0	Base control register offset
BASE_STATUS	0x0004	1	Base status register offset
BASE_FRAME_END	0x0008	2	Base frame clock end count register offset
BASE_FRAME_INT	0x000C	3	Base frame clock interrupt count register offset
BASE_FRAME_CNT	0x0010	4	Base current frame count read port offset
BASE_PKT_DELAY	0x0014	5	Base Set and Read Packet Delay port offset
BASE_LM75	0x0058	22	LM75 interface
CHAN_0_CNTRL	0x0018	6	Channel 0 control register offset
CHAN_0_STATUS	0x001C	7	Channel 0 status register offset
CHAN_0_TX_DMA	0x0020	8	Channel 0 write DMA start register offset
CHAN_0_RX_DMA	0x0024	9	Channel 0 read DMA start register offset
CHAN_0_PKT_1	0x0024	9	Channel 0 Pointer to next Packet Upper RO
CHAN_0_TX_INDX	0x0028	10	Channel 0 Transmitter index read port offset
CHAN_0_RX_INDX	0x002C	11	Channel 0 Receiver index read port offset
CHAN_0_PKT_0	0x0030	12	Channel 0 Pointer to next Packet Lower RO
CHAN_1_CNTRL	0x0034	13	Channel 1 control register offset
CHAN_1_STATUS	0x0038	14	Channel 1 status register offset
CHAN_1_TX_DMA	0x003C	15	Channel 1 write DMA start register offset WO
CHAN_1_RX_DMA	0x0040	16	Channel 1 read DMA start register offset WO
CHAN_1_PKT_1	0x0040	16	Channel 1 Pointer to next Packet Upper RO
CHAN_1_TX_INDX	0x0044	17	Channel 1 Transmitter index read port offset
CHAN_1_RX_INDX	0x0048	18	Channel 1 Receiver index read port offset
CHAN_1_PKT_0	0x004C	19	Channel 1 Pointer to next Packet Lower RO

BAR 1: I/O channel RAM blocks are mapped to this space for single-word access

CHAN_0_TX_RAM	0x0000 – 0x3FFC
CHAN_0_RX_RAM	0x4000 – 0x7FFC
CHAN_1_TX_RAM	0x8000 – 0xBFFC
CHAN_1_RX_RAM	0xC000 – 0xFFFC

FIGURE 4 PCIE-ASCB INTERNAL ADDRESS MAP

The address map provided is for the local decoding performed within the PCIe-ASCB. The addresses are all offsets from a base address, which is assigned by the system when the PCI bus is configured.

Programming

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

Depending on the software environment, it may be necessary to set-up the system software with the PCI "registration" data.

In order to receive data, the software is only required to enable the Rx channel and set the frequency parameters. To transmit the software will need to load the message into the appropriate channel Dual Port RAM, set the frequency and mode and enable the transmitter.

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

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

Refer to the Theory of Operation section above and the Interrupts section below for more information regarding the exact sequencing and interrupt definitions.

The VendorId = 0xDCBA. The CardId = 0x0067.

Revisions

26.4 7/23 JTAG reprogramming removed, PCI and PCIe models will have the same programming model, temperature sensor updated to LM75, Bridge replaced due to EOL part, external PLL removed, DCM locked status added.

Register Definitions

BASE_CONTROL

[0x00] ASCB Base Control Register (read/write)

Control Register

DATA BIT	DESCRIPTION
31	Test Clock En
30-11	Spare
10	FilterDisable
9	FrameDisable
8	Spare
7	Channel 1 Rx Interrupt Enable
6	Channel 1 Tx Interrupt Enable
5	Channel 0 Rx Interrupt Enable
4	Channel 0 Tx Interrupt Enable
3	Frame Count Loop Enable
2	Frame Interrupt Enable
1	Frame Count Clear
0	Frame Count Enable

FIGURE 5 PCIE-ASCB BASE CONTROL REGISTER BIT MAP

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

Channel 1 Rx Interrupt Enable: Set to enable the receive interrupt from port 1. Default is disabled.

Channel 1 Tx Interrupt Enable: Set to enable the transmit interrupt from port 1. Default is disabled.

Channel 0 Rx Interrupt Enable: Set to enable the receive interrupt from port 0. Default is disabled.

Channel 0 Tx Interrupt Enable: Set to enable the transmit interrupt from port 0. Default is disabled.

Frame Count Loop Enable: If set the Frame Counter will count 0->End Count and repeat. If cleared the counter will stop at the end count.

Frame Interrupt Enable: Set to enable the Frame Count based interrupt. Clear to disable. Matching count in Frame Interrupt Count register.

Frame Count Clear: Set to clear the Frame Count. Must be returned to ‘0’ to return to standard operation.

Frame Count Enable: Set to enable Frame Counting. See Frame End Count register.

FrameDisable: When ‘1’ the Frame output signal is forced to 0. The counter continues to operate but the pulse is not transmitted.

FilterDisable: When ‘1’ the input filter is disabled. ‘0’ for normal operation.

Test Clock En: When set to ‘1’ A 10 MHz test clock is output on the 4 transmit ports for debugging. ‘0’ for normal operation. Useful for debugging, testing cables etc.

BASE_STATUS

[0x04] ASCB Base Status Register (read status/write clear)

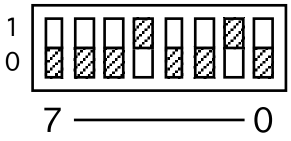
Status Register

DATA BIT	DESCRIPTION
31	Interrupt Active (read only)
30	Lock 200
29	Lock 80
28	LM 75 NAK
27-24	Xilinx Flash Revision Minor
23	Lock 40
22	Int Port 1
21	Int Port 0
20	Frame Interrupt Status (read/write clear)
19	Channel 1 Rx Interrupt Status (read/write clear)
18	Channel 1 Tx Interrupt Status (read/write clear)
17	Channel 0 Rx Interrupt Status (read/write clear)
16	Channel 0 Tx Interrupt Status (read/write clear)
15-8	Xilinx Flash Revision (read only)
7-0	User Switch 7-0 (read only)

FIGURE 6 PCIE-ASCB STATUS REGISTER

The Flash Major revision is 0x1A. the Minor Revision is currently x4. The FLASH ID will be updated as features are added or revisions made. The revision is also readable from the configuration space revision register.

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



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

Channel interrupts are duplicated here for user convenience. The interrupts can be operated from either location.

Lock 40, 80, 200 should all be set ‘1’ for normal operation. If not set the DCM is not locked and erratic behavior can result.

LM 75 NAK is set when an access to the temperature sensor fails due not being acknowledged. The external sensor asserts ACK when it receives a command as part of the I2C protocol. If not asserted when expect this bit is set. Data read is meaningless in this case. Should be ‘0’ for normal operation. This is a sticky bit and should be cleared by writing back to the port with this bit set.

BASE_FRAME_END

[0x08] ASCB Frame End Count Register (read/write)

Frame End Count Register

DATA BIT	DESCRIPTION
31-18	Spare
17-0	Frame End Count

FIGURE 7 PCIE-ASCB FRAME END COUNT REGISTER

The end count for the frame counter is loaded into this register. The count is based on the 12.5 MHz Frame Clock. The count runs from 0 -> End Count. With the Loop Control bit the user can select continuous operation or to stop at the End Count.

BASE_FRAME_INT

[0x0C] ASCB Frame Interrupt Count Register (read/write)

Frame Interrupt Count Register

DATA BIT	DESCRIPTION
31-18	Spare
17-0	Frame Interrupt Count

FIGURE 8 PCIE-ASCB FRAME INTERRUPT COUNT REGISTER

Separate from the End Count is the Interrupt Count for the frame. The Interrupt count represents the Frame Count where a programmable interrupt can occur. When in loop-mode this will happen once per loop with a period based on the End Count and the 80 nS period. Please note: it is possible to program the Interrupt Count larger than the end count which will result in no interrupts being generated. Recommended to program to a value within the range of the End Count programmed.

BASE_FRAME_CNT

[0x10] ASCB Frame Count Register (read only)

Current Frame Count Register

DATA BIT	DESCRIPTION
31-18	Spare
17-0	Current Frame Count

FIGURE 9 PCIE-ASCB CURRENT FRAME COUNT REGISTER

The Current Frame Count is available in this register. The Frame Count is generated on the 12.5 MHz clock and is synchronized to the PCI clock for read-back purposes. The count may jump a bit when reading due to the synchronization.

BASE_PKT_DELAY

[0x14] ASCB Inter Packet Delay Count Register

Current Frame Count Register

DATA BIT	DESCRIPTION
31-8	Spare
7-0	Packet Delay

FIGURE 10 PCIE-ASCB PACKET DELAY REGISTER

The transmitter for D or E type operation will provide a minimum inter-packet gap time. The time is programmable as an 8 bit value with 80 nS granularity per bit. 256 * 80 = 20.48 uS maximum gap. The gap time is in addition to the transmitter delay time used to insure the last bit makes it through the transformers [~250 nS].

The scheduler can also be used to transmit at specific times. Please see the transmit memory control section for more information.

The inter-packet minimum gap defaults to 9.6 uS to meet the D type processing requirements. Normally the first packet sent will be scheduled and additional packets will transmit based on the minimum gap for any one time slot allocated by the system. The user must make sure the size of the transmissions will fit within the allocated time when the stop time is not being used.

BASE_LM75

[0x58] Temperature Port

Temperature Port

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

FIGURE 11 PCIE-ASCB LM75

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

0.125 C is the bit value.

CHAN_CONTROL

[0x18, 0x34] ASCB Channel 0, 1 Control Register (read/write)

Channel Control Register

DATA BIT	DESCRIPTION
17	Encoding
16	RxIntEn1
15	RxSources
14	RxStart1
13	InternalCrc
12	Init
11-10	Spare
9	Channel Force Interrupt
8	Channel Master Interrupt Enable
7	Receive DMA Enable
6	Transmit DMA Enable
5	RxIntEn0
4	TxIntEn
3	Manchester Data Invert
2	Spare
1	RxStart0
0	TxEn

FIGURE 12 PCIE-ASCB CHANNEL CONTROL REGISTER

TxEn when set ‘1’ enables the Transmitter to operate in accordance with the programmed options contained in the DPR and this register.

RxStart0, RxStart1 when set ‘1’ enable the two receive ports per channel separately. Both the Primary and the Secondary data streams can be directed to the 0 port using the RxSources and RxSourceSelect bits. Alternatively both the Primary(0) and Secondary(1) can be received in parallel.

RxSourceSelect is used to select which channel [Primary ‘0’ Secondary ‘1’] are received by the lower memory section. This bit only has meaning when the channel is in single port mode. This bit has been removed with the update to the “DE” level design.

RxSources is used to select between 1 receive channel and 2 receive channels. The full receive DPR for the channel is used for the single port when cleared = LargeMap model. When set ‘1’ the small map version is used where the DPR is split in half with RxStart0 – primary used to load the lower half of the DPR and the secondary input plus RxStart1 used to load the upper half of the DPR allocated to the channel.

When in LargeMap mode use only one RxStart bit at a time. Which ever is enabled will determine which data is captured. RxStart0 will capture the primary and RxStart1 will capture the secondary channel. Both data sets will be loaded to the full DPR range. Enabling both channels will cause data overwrite with “interesting” results.

Manchester Data Invert is used to optionally invert the transmit and receive data to be compatible with non-standard equipment. The default is ‘0’ and usually is not needed. If the other equipment is not receiving what this board is sending and vice-versa, try this bit. No effect on 8B10B data.

TxIntEn, RxIntEn0, RxIntEn1 are enables to allow interrupts from the transmitter, and each of the receivers.   Set to enable interrupts.

Transmit DMA Enable, Receive DMA Enable are set to allow DMA operation into and out of the DPR associated with the transmit and receive functions in the channel. The driver will normally manage these bits.

Channel Master Interrupt Enable must be set to allow the user interrupts to be asserted. This bit allows for a quick on/off control for the interrupts without individually disabling them.

Channel Force Interrupt when set causes an interrupt from the channel. Clear to remove the interrupt. Used for test purposes and SW development. You can set to simulate an interrupt from the channel to test your ISR etc.

Init acts as a local reset for the channel. Active when set. Normal operation when cleared. Returns State Machines etc. to their Idle or default states.

InternalCrc is used to control the insertion of the internally calculated CRC into the receive data stream. We used this option to help debug the calculated versus received CRC when doing the initial integration. When the CRC error bit is not set in the message status the received CRC matches the internally generated CRC.

Encoding selects between the Manchester “D” type operation and the 8B10B “E” type. The default is D type (0). When set ‘1’ the “E” type is enabled.

CHAN_STATUS

[0x1C, 0x38] ASCB Channel 0, 1 Status Register (read status/write clear)

Channel Status Register

DATA BIT	DESCRIPTION
31	Interrupt Active
30-14	Spare
13	RxPacket1
12	RxIntLat1
11	Receive DMA Ready
10	Transmit DMA Ready
9	Receive DMA Error
8	Transmit DMA Error
7	Receive DMA Interrupt
6	Transmit DMA Interrupt
5	RxIntLat0 (read/write clear)
4	TxIntLat (read/write clear)
3	Spare
2	RxPacket0
1	Receiver Active
0	Spare

FIGURE 13 PCIE-ASCB CHANNEL STATUS REGISTER

Spare bits are tied to ‘0’.

Receiver Active is an additional read-back location for the RxStart0 bit. When ‘1’ in the control register it will be ‘1’ in this status as well and vice-versa.

RxPacket0, RxPacket1 indicate that a packet receive is in progress on that channel.

TxIntLat, RxIntLat0, RxIntLat1 when set indicate that an interrupt is active for the respective port. Clear by writing back to the bit with a ‘1’. Enabled in the control register. The enable is anded after the status latch allowing for polled operation using the status instead of being interrupt based.

Transmit DMA Interrupt, Receive DMA Interrrupt are set when the respective DMA operation completes. Usually handled by the driver.

Transmit DMA Error, Receive DMA Error when set indicate that an error occurred during interrupt processing. Usually only seen during initial integration when pointers or commands are not quite right.

Transmit DMA Ready, Receive DMA Ready are status signals from the DMA state machines indicating that the DMA engines are back to the IDLE state and ready for a new command. Used when a DMA action is terminated and the next command should not be written until the DMA system “unwinds”.

Interrupt Active is a quick read status bit to indicate that no active user interrupts are present or conversely that something in the channel is interrupting. Software can check this bit to make a quick decision when processing interrupts and determining the source.

CHAN_TX_DMA

[0x20, 0x3C] ASCB Channel 0, 1 Tx DMA Register (write only)

Input DMA Pointer Address Port

Data Bit	Description
31-0	First Chaining Descriptor Physical Address

FIGURE 14 PCIE-ASCB CHANNEL WRITE DMA POINTER PORT

This write-only port is used to initiate a scatter-gather input DMA. When the address of the first chaining descriptor is written to this port, the DMA engine reads four successive long words beginning at that address. The first is the address of the first memory block of the DMA buffer, the second is the local RAM address to start writing that memory block into, the third is the length in bytes of that block [only the lower 22 bits are valid], the second is the local RAM address to start writing that memory block into, and the fourth 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 is set in one of the next pointer values.

CHAN_RX_DMA

[0x24, 0x40] ASCB Channel 0, 1 Rx DMA Register (write only)

Output DMA Pointer Address Port

Data Bit	Description
31-0	First Chaining Descriptor Physical Address

FIGURE 15 PCIE-ASCB CHANNEL READ DMA POINTER PORT

This write-only port is used to initiate a scatter-gather output DMA. When the address of the first chaining descriptor is written to this port, the DMA engine reads four successive long words beginning at that address. The first is the address of the first memory block of the DMA buffer, the second is the local RAM address to start reading that memory block from, the third is the length in bytes of that block [only the lower 22 bits are valid], and the fourth 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 is set in one of the next pointer values.

CHAN_TX_INDX

[0x28, 0x44] ASCB Channel 0, 1 Tx Index Register

Channel Transmit Index Register

DATA BIT	DESCRIPTION
31-28	Spare
27-16	Starting RAM Address of Next Transmit DMA
15-12	Spare
11-0	Current Transmit Access RAM Address

FIGURE 16 PCIE-ASCB CHANNEL TRANSMIT INDEX REGISTER

This register can be accessed to see where to write to and where data is currently being transmitted from. Since the memory is Dual Ported, the Next Data can be written before the current data is completed. Since the memory is circular in nature, the new data should not be written if it will overwrite the current block of data. Some pointer arithmetic is required for proper management of the TX side.

The transmit side uses the read DMA – the names are relative to the ASCB and for transmit the data is read from the system memory and written to local memory.

CHAN_RX_INDX

[0x2C, 0x48] ASCB Channel 0, 1 Rx DMA Register (read only)

Channel Receive Index Register

DATA BIT	DESCRIPTION
31-28	Spare
27-16	Starting RAM Address of Next Output DMA
15-12	Spare
11-0	Current Receive Access RAM Address

FIGURE 17 PCIE-ASCB CHANNEL RECEIVE INDEX REGISTER

This register can be accessed to see where to read from and where data is currently being received into. Since the memory is Dual Ported, the Previous Data can be read before the current data is completed. Since the memory is circular in nature, the new data should be read before new data will overwrite it. Some pointer arithmetic is required for proper management of the RX side.

CHAN_X_PKT_Y

[0x24, 0x30, x40, x4C] ASCB Channel 0, 1 Rx Next Packet Address (read only)

Channel Receive Index Register

DATA BIT	DESCRIPTION
31-12	Spare
11-0	Address of Next Packet to be stored

FIGURE 18 PCIE-ASCB CHANNEL RECEIVE INDEX REGISTER

X = channel, Y = 0,1 for lower, upper memory within the channel. The upper memory pointer information only has meaning when the upper memory is enabled.

During the reception of a packet the address is incremented. At the end of storing the current packet, the location of the start of the next packet becomes known. The next packet location information is stored within the RAM as part of the status. It is also available to be read from these 4 registers. The memory version will be valid until overwritten or cleared.

When both memory's per channel are enabled the memory is split and becomes smaller. The addresses are '0' extended for the lower side and have the upper bit set for the upper portion. The addresses can be polled and when updated the previous packet can be read.

TX_SM_MEM

In transmit the first two 32-bit words are the control word. The packet follows: Data, CRC. The CRC position should be set to zero as this is part of the CRC calculation. The hardware will calculate and insert the CRC.

```
  nextstrt      <= control(11 downto 0);
// pointer to the next transmitted block start location
  tpri_dis      <= control(12);  // primary disable
  tbkp_dis      <= control(13); // secondary disable
  crc_dis       <= control(14); // CRC generation disable – use SW defined
  start_en      <= control(15); //  Set to require use of start time
  stop_en       <= control(16); // set to require use of stop time
  strtcnt       <= control(34 downto 17); // time to start transmission
  stopcnt       <= control(52 downto 35); // stop on or before
```

The start time is required for the first block of data to be sent. After that block is sent as long as the transmitter is enabled and the next block is ready to be sent before the prior one is completed, the start_en can be set to '0'. In this case the second block is sent immediately after the first block based on hardware delays.

Memory configuration per block of data.

Location IO	Value (32-bits)
0	lower command
1	upper command
4	First data
…	
N	last data
N+1	CRC or zero data

RX_SM_MEM

Each received data block has the first two 32-bit words used for status/control. The packet follows: Data, CRC. The CRC is the last 32 bit word received. The hardware will calculate the CRC on the received data and compare with the received value setting [or not] the CRC Error bit in the status word with the results.
Receiver Status (first two 32-bit words in packet)

```
status(11 downto 0)      <= rxend; // pointer to the start of the next block
status(16 downto 12)     <= num_bits;  //
status(17)               <= bus_sel;  // which port was received
status(18)               <= man_err; // Manchester error was detected
status(19)               <= crc_err; // CRC error was detected
status(37 downto 20)     <= strt_cnt; // start time of message received
status(55 downto 38)     <= stop_cnt; // end time of message received
status(63 downto 56)     <= x"AA"; // fixed termination pattern
```

PCIe-ASCB IO Pin Assignment

The figure below gives the pin assignments for the IO Interface the ASCB connectors. J2 is the Channel 0 or Left port, and J3 is the Channel 1 or Right port.

PrimaryP	1
PrimaryN	6
PTXDIS	2
STXDIS	3
SecondaryP	4
SecondaryN	8
Frame	5
Grounds	7,9

FIGURE 19 ASCB CONNECTOR PINOUT

Applications Guide

Interfacing

The pin-out tables are displayed with the pins in the same relative order as the actual connectors. The pin definitions are defined with noise immunity in mind. The pairs are chosen to match the ASCB standard.

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

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

Power all system power supplies from one switch. Connecting external voltage to the when it is not powered can damage it, as well as the rest of the host system. This problem may be avoided by turning all power supplies on and off at the same time.

Cables. The ASCB specification calls out a specific cable for use in ASCB systems. The cable is for a half duplex system with nodes interconnected linearly and terminations at the ends of the cable. The D type interface operates with Manchester encoding at 10 MHz bit rate. The E type operates with 8B10B encoding at 20 MHz. The interface is transformer isolated differential 10V signalling. The terminations are set to 124 ohms each or 62 ohm parallel impedance.

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 devices rated voltage.

Construction and Reliability

PCI/PCIe Modules were conceived and engineered for rugged industrial environments. PCIe-ASCB is constructed out of 0.062 inch thick high temperature ROHS compliant material.

Surface mounted components are used. The components are available with commercial and Industrial temperature ranges. Conformal coating is an option for condensing environments or for another measure of board protection. Please order the "-CC" version.

The PCI/PCIe module 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. PCIe-ASCB has a wider keep out than required by PCI specification for most of the top edge 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 PCIe-ASCB 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. See the FAN option.

Warranty and Repair

Please refer to the warranty page on our website for the current warranty offered and options. https://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 B&C
Santa Cruz, CA 95060
831-457-8891
support@dyneng.com

Specifications

Logic Interface:	PCIe 4 lanes Gen1. PCI version has 33/66 32 bit PCI primary interface. 32/33 interface to PMC and ASCB function.
Digital Serial IO:	eASCB & ASCB-D protocol versions.
PMC Position	1 PMC position suitable for PrPMC or standard PMC operation.
DIP Switch:	DipSwitch supplied for board identification and other user purposes.
CLK rates supported:	40 Mhz local oscillator used to generate internal rates as reference for RX and TX of Manchester and 8b10b data.
Software Interface:	ASCB 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
Interrupt:	Each port has independently programmable interrupt sources, DMA interrupts included.
Onboard Options:	All Options are Software Programmable
Interface Options:	2 DB9 Pin connectors at front bezel for ASCB channels.
Dimensions:	Standard 1/2 length PCIe module.
Construction:	Multi-Layer Hi-temp Printed Circuit, Through Hole and Surface Mount Components.
Power:	12V and 3.3V from PCIe used to generate local voltages [5, -5, 3.3, 2.5, 1.8, 1.2]. PCI version has fewer supplies needed.
Weight:	TBD oz
Temperature Range	-40⇔ +85C or better rated parts standard.

Order Information

PCIe-ASCB PCIe Module with 2 redundant ASCB channels. ASCB-D and eASCB supported. PMC position. Software control over mode of operation and operational parameters.
https://www.dyneng.com/PCIe-ASCB.html

PCI-ASCB PCI Module with 2 redundant ASCB channels. ASCB-D and eASCB supported. PMC position. Software control over mode of operation and operational parameters.
https://www.dyneng.com/PCI-ASCB.html

Order Options:

Pick any combination to go with IO
-CC to add conformal coating
-ROHS to add ROHS processing
-FAN to add a Zero Slot 5.2 CFM fan to the assembly
-FANR to add a rear mount 8CFM fan to the assembly – will take two slots to install

Win10/11 driver software available with board purchase. Additional OS porting available upon request.